The LLM Data Auditor: A Metric-oriented Survey on Quality and Trustworthiness in Evaluating Synthetic Data

Kaituo Zhang *kzhang42@cougarnet.uh.edu*
University of Houston

Mingzhi Hu
Worcester Polytechnic Institute

Hoang Anh Duy Le
Rice University

Fariha Kabir Torsha
University of Houson

Zhimeng Jiang
Texas A&M University

Minh Khai Bui
University of Wisconsin - Madison

Chia-Yuan Chang
Texas A&M University

Yu-Neng Chuang
Rice University

Zhen Xiong
University of Southern California

Ying Lin
University of Houston

Guanchu Wang
University of North Carolina at Charlotte

Na Zou
University of Houston

Abstract

Large Language Models (LLMs) have emerged as powerful tools for generating data across various modalities. By transforming data from a scarce resource into a controllable asset, LLMs mitigate the bottlenecks imposed by the acquisition costs of real-world data for model training, evaluation, and system iteration. However, ensuring the high quality of LLM-generated synthetic data remains a critical challenge. Existing research primarily focuses on generation methodologies, with limited direct attention to the quality of the resulting data. Furthermore, most studies are restricted to single modalities, lacking a unified perspective across different data types. To bridge this gap, we propose the **LLM Data Auditor framework**. In this framework, we first describe how LLMs are utilized to generate data across six distinct modalities. More importantly, we systematically categorize intrinsic metrics for evaluating synthetic data from two dimensions: quality and trustworthiness. This approach shifts the focus from extrinsic evaluation, which relies on downstream task performance, to the inherent properties of the data itself. Using this evaluation system, we analyze the experimental evaluations of representative generation methods for each modality and identify substantial deficiencies in current evaluation practices. Based on these findings, we offer concrete recommendations for the community to improve the evaluation of data generation. Finally, the framework outlines methodologies for the practical application of synthetic data across different modalities. Our repository has been released: https://anonymous.4open.science/r/Awesome-LLM-Data-Generation-6457.

1 Introduction

Data serves as the cornerstone of modern AI development. As real-world data sources are gradually being depleted, synthetic data have become increasingly favored by the research community, with model-based data generation emerging as a new paradigm. Consequently, Large Language Models (LLMs), with their powerful generative capabilities, play a pivotal role in this process. Numerous studies have already utilized LLMs for data generation in various dimensions. For scenarios with existing data but no annotations, Martorana et al. (2024) proposed a method to support metadata enrichment using LLM-generated topic annotations. LLMs can also be used to control or compose existing corpora (Penedo et al., 2025; Soldaini et al., 2024). Furthermore, LLMs are capable of generating diverse data types, including text and code (Wang et al., 2024e; Nadăş et al., 2025), tabular data (Fang et al., 2025), and graph data (Ji et al., 2025). They are also applied in specific practical domains, such as generating clinical records (Barr et al., 2025) and designing safety-critical scenarios for autonomous driving (Adekanye, 2024). Clearly, leveraging LLMs for data generation is becoming a crucial strategy to address the challenge of data scarcity.

However, the quality and rigorous evaluation of LLM-generated data is of paramount importance. It is well-known that high-quality data significantly enhance LLM performance; for example, enforcing simple correctness criteria on synthetic examples can be as vital as increasing dataset size for downstream performance (Iskander et al., 2024). In contrast, low-quality data can severely degrade model performance and impact real-world applications. At the model level, theory and large-scale experiments on repeated training over generated corpora reveal that uncontrolled reliance on synthetic data can distort scaling laws, leading to "model collapse," where models gradually lose skills and degenerate when exposed to their own outputs across generations (Dohmatob et al., 2024). In parallel, studies on undesirable memorization and privacy leakage highlight the risk that synthetic data pipelines may endanger personally identifiable or proprietary content (Satvaty et al., 2025; Aditya et al., 2024; Shanmugarasa et al., 2025). While these issues are receiving increasing attention, many current evaluation methods rely heavily on LLMs for scoring or filtering, which introduces significant model-specific biases (Gu et al., 2025). Consequently, data quality profoundly influences both model development and practical applications, highlighting the urgent need for a more systematic organization of data evaluation methodologies.

Current research on LLM-based data generation primarily focuses on the models themselves or the generation process, often overlooking the evaluation of the generated data. For instance, Long et al. (2024) structures the literature around the generic workflow of generation and curation in natural language processing, but only briefly mentions evaluation concepts. Similarly, Wang et al. (2024a) focuses on the lifecycle of synthetic data usage, dividing it into stages such as pre-training, supervised fine-tuning, and alignment. While some surveys do introduce evaluation methods, they are often restricted to a single modality, such as healthcare or tabular data (Pezoulas et al., 2024; Fang et al., 2024). Furthermore, most evaluations of generated data are extrinsic, which means they only measure improvements in model downstream tasks' performance. We summarize some representative works and comparisons in Table 9. In contrast, intrinsic evaluation, which assesses data quality directly, remains underdeveloped. Although works like Dataflow (Liang et al., 2025) have formalized a "generate-evaluate-filter-refine" paradigm, the literature still lacks a unified framework to audit synthetic data before it enters the training loop.

To ensure that LLM-generated data fulfills its true potential, the research focus must shift from generation techniques to evaluation methodologies. In this survey, we adopt a data-based perspective and establish metrics as our primary organizing principle. Unlike existing surveys oriented toward workflows, lifecycles, or generation reviews (Long et al., 2024; Wang et al., 2024a; Goyal & Mahmoud, 2024), we introduce the **LLM Data Auditor framework**, as shown in Figure 1 and 2. This framework organizes various data types through a unified structure encompassing 5 core components: LLM-based data generation methods, quality metrics, trustworthy metrics, evaluation gaps, and data usage. Specifically, quality metrics focus on validity, fidelity, and utility to measure fundamental usability, while trustworthy metrics assess safety and faithfulness to identify potential risks. Applying this framework, we analyze the representative literature to identify existing evaluation gaps and discuss practical usages. Ultimately, our framework provides the community with a comprehensive guide for generating and evaluating high-quality, multi-modal data using LLMs.

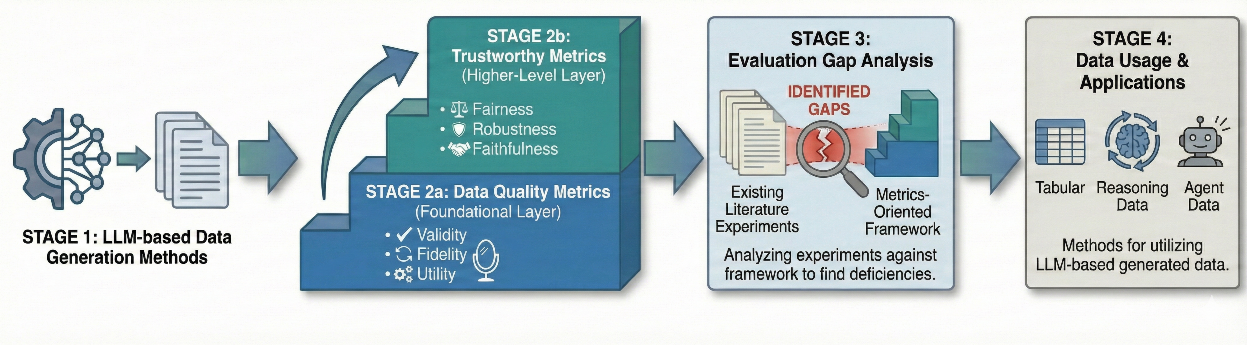
STAGE 1: LLM-based Data Generation Methods
STAGE 2b: Trustworthy Metrics
(Higher-Level Layer)
• Fairness
• Robustness
• Faithfulness
STAGE 2a: Data Quality Metrics
(Foundational Layer)
• Validity
• Fidelity
• Utility
STAGE 3: Evaluation Gap Analysis
IDENTIFIED GAPS
Existing Literature Experiments
Metrics-Oriented Framework
Analyzing experiments against framework to find deficiencies.
STAGE 4: Data Usage & Applications
Tabular
Reasoning Data
Agent Data
Methods for utilizing LLM-based generated data.

Figure 1: Overview of the LLM Data Auditor Framework. Stage 1: LLM-Driven data generation methods (Section 2.1, 3.1, 4.1, 5.1, 6.1, 7.1). Stage 2a: Data Quality Metrics (Section 2.2, 3.2, 4.2, 5.2, 6.2, 7.2). Stage 2b: Data Trustworthy Metrics (Section 2.3, 3.3, 4.3, 5.3, 6.3, 7.3). Stage 3: Evaluation Gap Analysis (Section 2.4, 3.4, 4.4, 5.4, 6.4, 7.4). Stage 4: Data Usage (Section 2.5, 3.5, 4.5, 5.5, 6.5, 7.5).

Finally, we summarize the main contributions of this survey as follows:

- **Shifting to a Data Perspective for Comprehensive Evaluation** Unlike existing literature that primarily focuses on the model perspective, our work distinguishes itself by directly adopting a data perspective, which we term **LLM Data Auditor**. This survey is structured around data, starting with data generation using LLMs, followed by an introduction to the evaluation system. We further analyze the deficiencies of current research within our evaluation system and conclude by discussing how these data are utilized in various methods.
- **A Systematic Metrics Taxonomy.** Our framework provides a systematic approach to data evaluation by directly classifying metrics according to their utility. We first categorize metrics into two major pillars: Quality and Trustworthiness. We then offer a more detailed breakdown into sub-categories, such as validity and fidelity for quality, and fairness, robustness, and privacy for trustworthiness. This contribution offers a clear roadmap for the community to evaluate the quality of synthetic data directly.
- **A Unified Cross-Modal Coverage.** LLM Data Auditor goes beyond a single modality by organizing six mainstream data modalities and evaluating them under a unified framework. By observing different modalities through this consistent perspective, our work provides the community with systematic guidance on how to generate, evaluate, and utilize data across various modalities.
- **Evaluation Gap Analysis.** Guided by our framework, we conduct a systematic analysis of representative works in LLM-generated data. By auditing the experimental evaluations of these studies, we identify critical evaluation dimensions that are currently overlooked. Our analysis highlights specific evaluation gaps for the community to address in future research, providing actionable guidance for more rigorous data assessment.

2 Text Data

Text data is the primary modality for LLMs. Within the field of generative artificial intelligence, researchers increasingly use synthetic text to augment or replace real-world datasets in settings characterized by limited

Modality	Strategy	Representative Methods	Evaluation / Metrics	
Text Data	**Corpus-centric**	RedPajama-V2 (Together Computer, 2023), FineWeb (Penedo et al., 2024), FineWeb2 (Penedo et al., 2025)	**Validity**	GAR, $s_{\text{USL-H}}$
	Prompt-based	Self-Instruct (Wang et al., 2023b), Evol-Instruct (Xu et al., 2024a), UltraFeedback (Cui et al., 2024)	**Fidelity**	EDS, $\text{Acc}^{\text{strict}}_{\text{prompt}}$, s_{RUBER}, PMI-FAITH
	Training	Self-Play Fine-Tuning (Chen et al., 2024b), SimPO (Meng et al., 2024), ORPO (Hong et al., 2024), GRPO (DeepSeek-AI et al., 2024)	**Diversity**	Self-CosSim, TTR, Ent-n
			Faithfulness	Attr_{AIS}, $\text{Attr}_{\text{auto}}$, $\text{Pres}_{\text{intent}}$, Pres_{Lev}, $\text{Pres}_{\text{comb}}$, F1_{AP}
	Inference-time	Nucleus Sampling (Holtzman et al., 2020), Diverse Beam Search (Vijayakumar et al., 2018), JAM (Huang et al., 2025b), CoVe (Dhuliawala et al., 2024), RARR (Gao et al., 2023a), CheckGPT (Manakul et al., 2023)	**Safety**	TP, EMT, TSG, NT2T,
Symbolic, Logical Reasoning Data	**Heuristic Evolution**	WizardMath (Luo et al., 2025a), MetaMathQA (Yu et al., 2024a), WizardCoder (Luo et al., 2025b)	**Validity**	$\text{Acc}_{\text{verify}}$, Acc_{prov}, PassRate
	Tool-Verified Generation	OpenMathInstruct-1 (Toshniwal et al., 2024), OpenCodeInstruct (Ahmad et al., 2025), ProofWriter (Tafjord et al., 2021), SynLogic (Liu et al., 2025b), ALT-FLD×2 (Morishita et al., 2024)	**Fidelity**	Acc_{SC}, Agree $\rho_{\text{LLM-human}}$, Acc_{RM}
			Faithfulness	Val_{step}, $\text{Align}_{\text{entail}}$
	Trajectory-Harvesting	OpenAI o1 rollouts (OpenAI et al., 2024), SYNTHETIC-2 (Prime Intellect Team, 2025)	**Robustness**	Δ_{OOD}
	Preference-Curated LLM-Judge Data	UltraFeedback (Cui et al., 2024), Code-UltraFeedback (Weyssow et al., 2024), STaR (Zelikman et al., 2022)		
Tabular Data	**Prompt-Based**	CLLM Seedat et al. (2024), EPIC Kim et al. (2025), LITO Yang et al. (2024b), TabGen-ICL Fang et al. (2025), OCTree Nam et al. (2024)	**Validity**	χ^2, VR
			Fidelity	KST, TVD, Pearson Score, DetScore, P_α, $\widehat{\alpha\text{-Prec}}$
	Fine-Tuning	GReaT Borisov et al. (2023), Nguyen et al. Nguyen et al. (2024), HARMONIC Wang et al. (2024d), Table-LLM-Specialist Xing et al. (2024), TableDreamer Zheng et al. (2025)	**Diversity**	R_β, $\widehat{\beta\text{-Rec}}$
			Utility	AUC, MSE, RMSE
			Privacy	DCR, AUC_{MIA}, Adv_{MIA}, Gain_{AIA}
	Hybrid Architectures (Table↔Text)	AIGT Zhang et al. (2024c), gTBLS Sundar et al. (2024), hybrid LLM+tabular / GAN-style models	**Fairness**	Δ_{SPD}, Δ_{EO}, Δ_{EOp}, CovGap, Cond-Shift
Semi-Structured	**Graph Data**	LLM4GraphGen Yao et al. (2024), GoG Xu et al. (2024b), ontology-grounded KG generators Feng et al. (2024), GraphJudge Huang et al. (2025a), GAG Ji et al. (2025), GraphMaster Du et al. (2025)	**Validity**	$\text{Valid}_{\text{rule}}$
			Fidelity	MMD, FCD
			Diversity	Novelty, Uniq
	JSON Data	Outlines dottxt-ai (2025), LM Format Enforcer Lu et al. (2025), SchemaBench-style RL methods Lu et al. (2025)	**Utility**	Acc, F1
			Privacy	node-DP, edge-DP, edge-LDP
	Log Data	LogBench Li et al. (2024d), AUCAD Zhang et al. (2025a),	**Robustness**	σ_{SCR}, GAD, GAD^{cap}, D_{L_2}
Vision–Language (Image / Video)	**Image–Text Native Autoregressive**	Emu (Sun et al., 2024), Emu3 (Wang et al., 2024c), Chameleon (Team, 2025)	**Validity**	WFR
			Fidelity	ITA, $\text{Fidelity}_{\text{subj}}$, $\text{Fidelity}_{\text{prompt}}$,
	Image–Text External Diffusion Control	Kosmos-G (Pan et al., 2024), GILL (Koh et al., 2023)	**Diversity**	IS, Div_{seq}, $\text{Score}_{\text{div}}$
	Video–Text Native Spatiotemporal	VideoPoet (Kondratyuk et al., 2024), Emu3 (Wang et al., 2024c)	**Utility**	Acc_{QA}, Win_{Rate}
			Safety	ASR , RR
	Video–Text Planner-based Diffusion	FlowZero (Lu et al., 2023), LVD (Lian et al., 2024), VideoDirectorGPT (Lin et al., 2024)	**Provenance**	$\text{Rate}_{\text{validate}}$
Agent Data (DTs & Embodied)	**Environment & Task Data**	ChatSUMO Li et al. (2024c), AutoScenario Lu et al. (2024), TTSG Ruan et al. (2025), ODD2CARLA Danso & Büker (2025), SDT-LLM Naeem et al. (2025), L3M+P Agarwal et al. (2025b), SELP Wu et al. (2025), T^3 Planner Li & Zhao (2025), PARTNR Chang et al. (2024)	**Validity**	ExecRate, SR_{valid}, PC
			Fidelity	FID/FVD, SSIM, PSNR, LPIPS, SemFid, MatchRate
			Diversity	AD, RD
	Control & Decision Data	Grid-Agent Zhang et al. (2025c), Twin-2K-500 Toubia et al. (2025), BEHAVIORCHAIN Li et al. (2025a), LLM Trainer George & Farimani (2025), ELLMER Mon-Williams et al. (2025), Instruct2Act Huang et al. (2023), ProgPrompt Singh et al. (2022)	**Utility**	SR_{eval}, SimSteps, Offloading, PC, CR, G_t
			Safety	RVR, RI, MSCR, Jerk_{RMS}, HardBrakeRate, SafetySat, Rejection/Risk, minTTC, MDC
	Perception & Telemetry Data	DefectTwin Ferdousi et al. (2024), SceneCraft Hu et al. (2024), Blender-LLM Du et al. (2024)		

Note. For modalities with multiple data types (e.g., semi-structured, vision–language), The "Evaluation / Metrics" column instantiates representative metrics for the first data type (graphs, image–text pairs); metrics for the remaining data types (JSON/logs, video–text) follow the same structure and are detailed in Sections 5, 6.

Table 1: LLM-driven data generation across modalities, strategies, methods, and evaluation metrics.

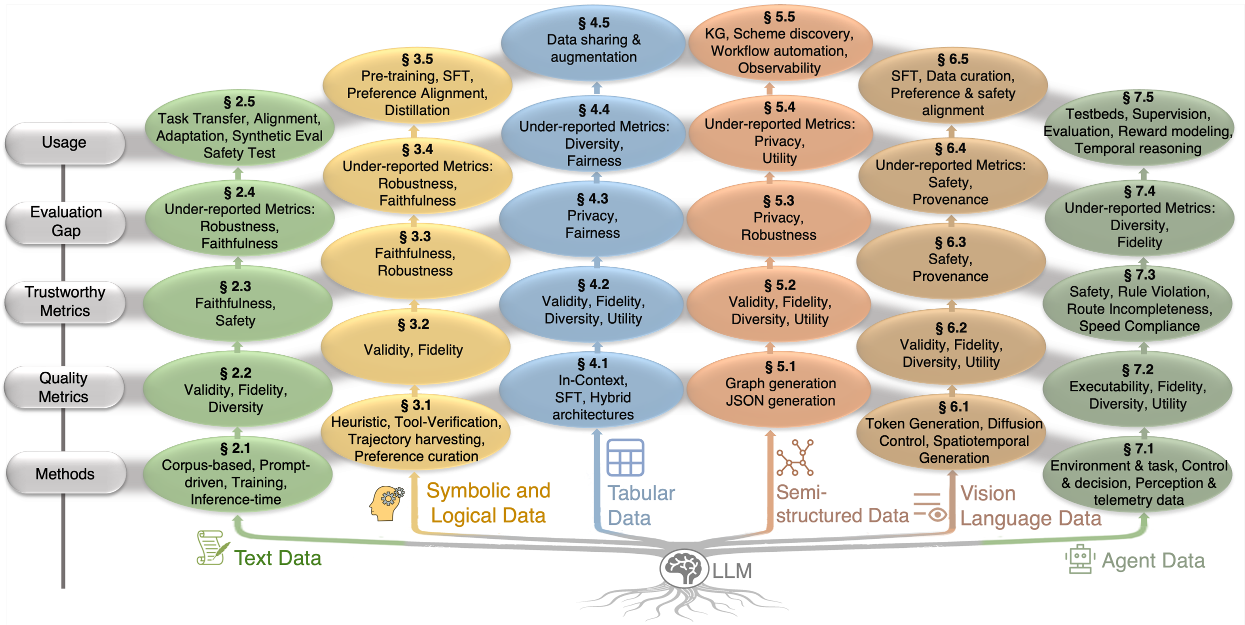
Usage
Evaluation Gap
Trustworthy Metrics
Quality Metrics
Methods
§ 2.5 Task Transfer, Alignment, Adaptation, Synthetic Eval Safety Test
§ 2.4 Under-reported Metrics: Robustness, Faithfulness
§ 2.3 Faithfulness, Safety
§ 2.2 Validity, Fidelity, Diversity
§ 2.1 Corpus-based, Prompt-driven, Training, Inference-time
Text Data
§ 3.5 Pre-training, SFT, Preference Alignment, Distillation
§ 3.4 Under-reported Metrics: Robustness, Faithfulness
§ 3.3 Faithfulness, Robustness
§ 3.2 Validity, Fidelity
§ 3.1 Heuristic, Tool-Verification, Trajectory harvesting, Preference curation
Symbolic and Logical Data
§ 4.5 Data sharing & augmentation
§ 4.4 Under-reported Metrics: Diversity, Fairness
§ 4.3 Privacy, Fairness
§ 4.2 Validity, Fidelity, Diversity, Utility
§ 4.1 In-Context, SFT, Hybrid architectures
Tabular Data
§ 5.5 KG, Scheme discovery, Workflow automation, Observability
§ 5.4 Under-reported Metrics: Privacy, Utility
§ 5.3 Privacy, Robustness
§ 5.2 Validity, Fidelity, Diversity, Utility
§ 5.1 Graph generation JSON generation
Semi-structured Data
§ 6.5 SFT, Data curation, Preference & safety alignment
§ 6.4 Under-reported Metrics: Safety, Provenance
§ 6.3 Safety, Provenance
§ 6.2 Validity, Fidelity, Diversity, Utility
§ 6.1 Token Generation, Diffusion Control, Spatiotemporal Generation
Vision Language Data
§ 7.5 Testbeds, Supervision, Evaluation, Reward modeling, Temporal reasoning
§ 7.4 Under-reported Metrics: Diversity, Fidelity
§ 7.3 Safety, Rule Violation, Route Incompleteness, Speed Compliance
§ 7.2 Executability, Fidelity, Diversity, Utility
§ 7.1 Environment & task, Control & decision, Perception & telemetry data
Agent Data
LLM

Figure 2: Overview of LLM-driven synthetic data generation across six modalities. We discuss text data (Section 2), symbolic and logical reasoning data (Section 3), tabular data (Section 4), semi-structured graph, JSON and log data (Section 5), vision–language data (Section 6), and agent data (Section 7).

resources or privacy concerns. This is achieved by creating artificial yet relevant training examples through pipelines that rely on specific prompts. As discussed by Nadăş et al. (2025), modern techniques for synthetic text generation are largely prompt-conditioned and prioritize controllability. These methods focus on specific attributes, styles, and edge cases while ensuring that the generated text remains realistic and relevant to the task. In terms of structure, common formats for synthetic text include instruction and output pairs. In this format, a directive is matched with a target response to support the training of models to follow instructions. Another common format involves multi-turn instruction sequences that represent interactions across multiple exchanges where the content depends on the previous context.

2.1 Generation Methods

Methods for controlling text data generation can be taxonomized by the locus of intervention: Source Control and Composition, Prompt-Driven Generation and Refinement, Parameter-Efficient and Alignment-Based Control, and Inference-Time Steering and Verification.

Source Corpus Control and Composition. For text data, control is often achieved by processing existing corpora into a trainable mixture with explicit and auditable signals rather than by generating new samples. Modern curation pipelines follow a standard sequence that begins with normalization and language identification. This is followed by the removal of near-duplicate content through approximate matching techniques such as Jaccard or MinHash deduplication across different crawl snapshots. Finally, document-level filtering is performed based on descriptors structured by metadata that are attached to each document (Penedo et al., 2024; 2025). LLMs are increasingly employed as scalable annotators to transform nuanced, abstract attributes—such as educational quality—into operational labels or calibrated scores. By distilling these high-level judgments into lightweight classifiers, researchers can convert subjective criteria into systematic signals. This pipeline enables the efficient filtering and reweighting of massive data mixtures (HuggingFaceFW, 2024).

Composition is treated as a primary control method. Composition refers to the combination of different sources and domains as well as their relative amounts. In practice, mixture policies are put into use through three main strategies. The first strategy involves mixing and weighting across various data streams. The second strategy involves selecting data at the level of individual documents by using quality signals. The third strategy involves the removal of specific parts of the collection that carry high risk. At the mixture level, transparent mixture design and reproducible tools such as Dolma allow practitioners to adjust domain coverage and data budgets without confusing a larger quantity of data with higher quality data (Soldaini et al., 2024). At the selection level, to enable the customization of downstream policies, some releases separate raw documents from quality signals. For example, RedPajama-V2 provides web text together with quality-related metadata for a subset of the corpus. This allows users to apply their own selection thresholds without the need to crawl the data again (Weber et al., 2024; Together Computer, 2023). At the exclusion level, safety-oriented processing can be performed earlier in the pipeline. Rather than relying only on refusal mechanisms after training, pretraining data filtering removes documents identified as enabling harmful capabilities. This includes content that provides significant support for misuse related to chemical, biological, radiological, and nuclear materials. This strategy targets measurable reductions in harmful capabilities while minimizing negative impacts on performance for standard tasks (Anthropic, 2025).

Prompt-Driven Generation and Refinement. When model parameters are frozen, control is managed through in-context bootstrapping and instruction design. Self-Instruct (Wang et al., 2023b) established the approach of extracting the internal knowledge of a model into labeled datasets by prompting it to generate diverse instances from seed tasks. To address the challenge of increasing the complexity of generated data, evolutionary strategies such as Evol-Instruct from the WizardLM project (Xu et al., 2024a) use mutation operators that systematically rewrite instructions to increase constraints and reasoning depth. These generation methods are often integrated into pipelines that follow a sequence of generation, ranking, and selection. In these workflows, an auxiliary LLM serves as a judge to evaluate candidate outputs against specific rubrics. For instance, the UltraFeedback framework (Cui et al., 2024) evaluates outputs based on criteria such as helpfulness and honesty. This feedback loop effectively shifts the control mechanism from manual prompt engineering to scalable and automated preference filtering (Gu et al., 2025).

Parameter-Efficient and Alignment-Based Control. To fundamentally alter the output distribution, parameter updates are required. While Supervised Fine-Tuning sets the behavioral baseline, recent advances have shifted toward iterative self-play mechanisms. Methods such as Self-Play Fine-Tuning, also known as SPIN (Chen et al., 2024b), allow the generator to improve by contrasting its own generated responses with human demonstrations in an iterative self-play loop. This approach effectively breaks the ceiling of static supervision without the need for extra annotations.

For preference alignment, the field is moving beyond the complex two-stage Reinforcement Learning from Human Feedback pipeline. New reference-free objectives eliminate the memory-heavy reference model and integrate instruction following directly into the alignment loss. Notable examples include Simple Preference Optimization (Meng et al., 2024) and Odds Ratio Preference Optimization (Hong et al., 2024). These methods help mitigate length bias, such as verbosity, without requiring separate reward modeling.

Furthermore, specialized objectives now target efficient group-level dynamics. Group Relative Policy Optimization (DeepSeek-AI et al., 2024), for instance, normalizes rewards across a group of generated outputs rather than using a critic model. This technique enables scalable preference optimization that has been used to improve reasoning and mathematical performance. Once equipped with these parameterized controls, the model effectively transitions from a generic predictor into a specialized data synthesizer. In this role, it is capable of autonomously producing vast quantities of high-fidelity samples that strictly adhere to target formats and reasoning protocols.

Inference-Time Steering and Verification. The final layer of control modulates the decoding process at runtime. Stochastic decoding strategies such as Nucleus Sampling (Holtzman et al., 2020) balance the trade-off between diversity and plausibility. Similarly, diversity-promoting penalties in beam search, such as Diverse Beam Search (Vijayakumar et al., 2018), are used to prevent redundancy. More recently, latent space interventions have emerged as a precise steering mechanism. Methods such as JAM (Huang et al., 2025b) edit activation vectors during the forward pass to adjust attributes such as sentiment or safety without retraining.

To ensure reliability, post-hoc verification mechanisms act as a quality filter. Systems such as Chain-of-Verification, which is also known as CoVe (Dhuliawala et al., 2024), prompt the model to cross-check its own outputs. At the same time, the RARR framework (Gao et al., 2023a) revises drafts by comparing them against retrieved evidence. Furthermore, SelfCheckGPT (Manakul et al., 2023) utilizes consistency sampling to detect and flag or filter likely hallucinated content. This process ensures that only verified samples are retained in the final dataset.

2.2 Quality Metrics for Text

We unify the evaluation of plain text and multi-turn dialogue under three complementary dimensions: **Validity**, **Fidelity**, and **Diversity**.

Validity. Validity requires generations to be structurally well-formed under explicit constraints as well as linguistically acceptable.Beyond structural considerations, we evaluate linguistic well-formedness using a CoLA-style grammatical acceptability scorer. Following prior work, we adopt a sentence-level acceptability classifier to assign acceptability scores to generated sentences (Raman & Shah, 2023). For this purpose, we use a RoBERTa-large model fine-tuned on the CoLA benchmark.Building on this practice, we report a Grammatical Acceptability Rate, also referred to as GAR, as the proportion of generated sentences whose acceptability score exceeds a fixed threshold:

$$\mathrm{GAR} = \frac{1}{\sum_{i=1}^{|\mathcal{T}|} m_i} \sum_{i=1}^{|\mathcal{T}|} \sum_{j=1}^{m_i} \mathbb{I}\big[C_{\mathrm{acc}}(u_{ij}) \geq \tau\big],$$

where the i-th generation t_i is segmented into m_i sentences $\{u_{ij}\}_{j=1}^{m_i}$. The term $C_{\mathrm{acc}}(u) \in [0, 1]$ denotes the acceptability score or the probability output by the CoLA-trained classifier. The variable τ is a pre-specified decision threshold such as 0.5, and $\mathbb{I}[\cdot]$ is the indicator function.

In conversational domains, validity extends to the area of social pragmatics. We adopt the hierarchical USL-H metric (Phy et al., 2020). This metric suggests that a response must satisfy three criteria in a specific order. We denote s_U, s_S, and s_L for scores of understandability, sensibleness, and likability , respectively, and s_L can be comprised of one or more qualities q_j, i.e., specificity or empathy. The concept is in the following equation:

$$s_{\text{USL-H}}(t) = \alpha_1 s_U(t) + \alpha_2 s_U(t) s_S(t) + \alpha_3 s_U(t) s_S(t) s_L(t).$$

$$s_L = \sum_j \beta_j q_j.$$

In these formulations, the variables s_U, s_S, s_L, and q_j are continuous values that range from 0 to 1. The coefficients α_i and β_j represent the weights assigned to each quality and are defined such that their respective sums equal 1. These methods are suitable for calculating scores in both automatic and human evaluation settings.

Fidelity. Fidelity measures how faithfully generations preserve semantic content and adhere to source materials. To quantify the semantic proximity between synthetic and real data at the corpus level, we adopt an embedding-based similarity method. This approach compares mean normalized embedding vectors using cosine similarity, which follows previous analyses that calculate the similarity between the average normalized embeddings of different groups or corpora (Hämäläinen et al., 2023).Specifically, we use a sentence encoder such as a SentenceTransformer to map a text instance to an embedding vector. We report the Embedding Distribution Similarity, also referred to as EDS, as follows:

$$\mathrm{EDS}(\mathcal{X}_{\mathrm{syn}}, \mathcal{X}_{\mathrm{real}}) = \cos\Big(\overline{\mathbf{e}}_{\mathrm{syn}}, \overline{\mathbf{e}}_{\mathrm{real}}\Big), \overline{\mathbf{e}}_{\star} = \frac{1}{|\mathcal{X}_{\star}|} \sum_{x \in \mathcal{X}_{\star}} \frac{f(x)}{\|f(x)\|_2}.$$

In this formulation, $\mathcal{X}_{\mathrm{syn}}$ and $\mathcal{X}_{\mathrm{real}}$ represent the synthetic and real corpora respectively. And the mean embedding $\overline{\mathbf{e}}_{\star}$ for a dataset $\mathcal{X}_{\star}$ (where $\star \in \{\mathrm{syn}, \mathrm{real}\}$) is normalized using the ℓ_2 norm prior to averaging. $f(\cdot)$ denotes the embedding function used to map raw data points into a high-dimensional feature space. Finally, the cosine similarity between two vectors $\mathbf{a}$ and $\mathbf{b}$ is defined as the dot product of the vectors divided by the product of their magnitudes.

For instruction following tasks, we measure instructional fidelity using the strict prompt level accuracy in IFEval. This metric evaluates verifiable constraints specified in the prompt through automatic checkers (Zhou et al., 2023). Specifically, the prompt level strict accuracy requires that all verifiable instructions are satisfied and is defined as follows:

$$\mathrm{Acc}_{\mathrm{prompt}}^{\mathrm{strict}} = \frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} \mathbb{I}\Big[\bigwedge_{k \in \mathcal{I}(p_i)} V_k(t_i, p_i) = 1 \Big],$$

where $\mathcal{P}$ represents the set of prompts and p_i is the i-th prompt. The term $\mathcal{I}(p_i)$ denotes the subset of verifiable instructions in p_i. Each function $V_k(\cdot) \in \{0, 1\}$ serves as an automatic checker for the k-th instruction, such as length, keyword count, or constrained formatting. We define $\mathrm{Fid}_{\mathrm{instr}}$ as being equivalent to $\mathrm{Acc}_{\mathrm{prompt}}^{\mathrm{strict}}$.

In dialogue settings, fidelity requires grounding in the dialogue history or external evidence. We adopt RUBER (Tao et al., 2018), which blends a referenced score and an unreferenced score. Following Tao et al. (2018), we first normalize each component score to the range between 0 and 1 via min max normalization:

$$\tilde{s} = \frac{s - \min(s)}{\max(s) - \min(s)}.$$

We then compute the RUBER score by heuristic aggregation. In this work, we use arithmetic averaging:

$$s_{\mathrm{RUBER}}(c, r, r_{\mathrm{ref}}) = \frac{1}{2}\Big(\tilde{s}_{\mathrm{ref}}(r, r_{\mathrm{ref}}) + \tilde{s}_{\mathrm{unref}}(c, r)\Big),$$

where $\tilde{s}_{\mathrm{ref}}$ measures candidate reference similarity and $\tilde{s}_{\mathrm{unref}}$ evaluates context response appropriateness.

When references are unavailable, we employ USR (Mehri & Eskenazi, 2020), which is a reference free dialogue evaluation framework. This framework provides interpretable sub metrics aligned with five dialogue qualities such as understandability, naturalness, context maintenance, interestingness, and knowledge usage. Following the original formulation, these sub scores are aggregated into an overall quality estimate via a regression model trained to reproduce human overall ratings (Mehri & Eskenazi, 2020). Finally, to detect hallucinations in document grounded generation, we use PMI-FAITH (Nandwani et al., 2023):

$$\text{PMI-FAITH}(d, h, r) = \log P(r \mid d, h) - \log P(r \mid h).$$

This differential captures the information gain from response r contributed specifically by the document d beyond the dialogue history h.

Diversity. To detect mode collapse and repetition, we evaluate diversity across semantic and lexical dimensions. For semantic diversity, we adopt Self Cosine Similarity. This metric represents the average pairwise cosine similarity between embeddings of all generated texts. A lower Self Cosine Similarity indicates a broader semantic spread (Zhang et al., 2024d):

$$\text{Self-CosSim} = \frac{2}{M(M-1)} \sum_{1 \le i < j \le M} \cos(\mathbf{h}_i, \mathbf{h}_j),$$

where $\mathbf{h}_i$ is the embedding of the i-th generated text and M is the number of generations.

For lexical diversity, we report the Type Token Ratio (Treffers-Daller et al., 2018) and Distinct N (Li et al., 2016). These metrics are defined as follows:

$$\text{TTR} = \frac{|\text{Types}|}{|\text{Tokens}|}, \quad \text{Distinct-}N = \frac{|\text{Unique } n\text{-grams}|}{|\text{Total } n\text{-grams}|}.$$

Higher values indicate richer vocabulary usage and reduced n gram repetition. Finally, we compute n gram Response Entropy to quantify how evenly the model utilizes the vocabulary space (Zhang et al., 2024d):

$$\text{Ent-}n = -\sum_{g \in \mathcal{G}_n} p_{\mathcal{T}}(g) \log p_{\mathcal{T}}(g),$$

where $\mathcal{G}_n$ is the set of all n grams in the generated corpus $\mathcal{T}$ and $p_{\mathcal{T}}(g)$ is the empirical corpus level n gram distribution.

2.3 Trustworthy Metrics for Text Data

We assess the trustworthiness of generated text through two primary lenses. These include faithfulness to evidence and persona as well as safety against adversarial or organic toxicity.

Faithfulness and Consistency. We evaluate faithfulness from two coupled aspects. The first aspect involves evidence selection when gold supporting evidence is available. The second aspect focuses on the attribution of the generated content to the provided sources.

When gold supporting evidence is available, we first evaluate evidence selection by comparing the predicted evidence set $\mathcal{E}_{\text{pred}}$ against the gold set $\mathcal{E}_{\text{gold}}$ at the evidence unit level such as supporting fact sentences. Following the standard supporting fact protocol in multi hop QA tasks like HotpotQA (Yang et al., 2018), we compute instance wise precision, recall, and F1 score:

$$\text{Prec}_{\text{evi}} = \frac{|\mathcal{E}_{\text{pred}} \cap \mathcal{E}_{\text{gold}}|}{|\mathcal{E}_{\text{pred}}|}, \quad \text{Rec}_{\text{evi}} = \frac{|\mathcal{E}_{\text{pred}} \cap \mathcal{E}_{\text{gold}}|}{|\mathcal{E}_{\text{gold}}|}, \quad \text{F1}_{\text{evi}} = \frac{2\text{Prec}_{\text{evi}}\text{Rec}_{\text{evi}}}{\text{Prec}_{\text{evi}} + \text{Rec}_{\text{evi}}}.$$

In this context, the vertical bars denote set cardinality and the intersection term counts correctly selected evidence units.

Beyond selecting the correct evidence, we evaluate whether the generated content is supported by the provided sources using the Attributable to Identified Sources framework (Rashkin et al., 2023). This approach defines attribution through the According to A, y test. A passage or sentence is considered attributable if a reader can verify it from the source set A instead of relying on external knowledge (Rashkin et al., 2023). Following the sentence level formulation in Retrofit Attribution using Research and Revision (RARR) (Gao et al., 2023a), we compute the average attributable rate across sentences:

$$\text{Attr}_{\text{AIS}}(y, A) = \frac{1}{|S(y)|} \sum_{s \in S(y)} \text{AIS}(s, A),$$

where y is the generated passage and $S(y)$ denotes the set of sentences in y. The term A represents the set of provided evidence snippets and the function $\text{AIS}(s, A)$ indicates whether sentence s is fully supported by one or more snippets in A under the AIS guideline (Gao et al., 2023a).

To enable scalable evaluation, we also use auto AIS proposed in RARR (Gao et al., 2023a). This approach approximates human attribution judgments using a factual consistency model based on natural language inference as surveyed in TRUE (Honovich et al., 2022). We let $\mathrm{NLI}(e, s)$ represent the entailment probability of evidence snippet e entailing sentence s. The auto AIS metric identifies the best supporting snippet for each sentence and computes the average score as follows:

$$\mathrm{Attr}_{\mathrm{auto}}(y, A) = \frac{1}{|S(y)|} \sum_{s \in S(y)} \max_{e \in A} \mathrm{NLI}(e, s).$$

To quantify how much a revised passage $\tilde{A}$ remains faithful to the original passage A beyond factual fixes, we adopt the preservation metrics from RARR (Gao et al., 2023a). These metrics evaluate preservation from two complementary perspectives. The first perspective is intent preservation, which determines whether the revision maintains the original intent. This is typically judged by human annotators according to a specific rubric. The second perspective is edit minimality, which measures the extent of changes to the surface form. This metric is used to penalize excessive rewrites such as stylistic paraphrasing, reordering, or unnecessary additions (Gao et al., 2023a). Formally, given N paired instances, we define Precision Intent as a binary score. This score assigns a value of 1 if the revision completely preserves the intent of the original and 0 otherwise. This is represented by an intent violation indicator $\nu(\tilde{A}_i, A_i)$ as follows:

$$\mathrm{Pres}_{\mathrm{intent}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[\nu(\tilde{A}_i, A_i) = 0].$$

To discourage unnecessary edits when intent is preserved, we use Precision Levenshtein. This metric is based on the character level Levenshtein distance normalized by the original length and clipped at zero:

$$\mathrm{Pres}_{\mathrm{Lev}} = \frac{1}{N} \sum_{i=1}^{N} \max\left(1 - \frac{d_{\mathrm{lev}}(\tilde{A}_i, A_i)}{\mathrm{len}(A_i)}, 0\right).$$

To prioritize semantics, we define Precision Combined by zeroing out surface similarity when intent is violated (Gao et al., 2023a):

$$\mathrm{Pres}_{\mathrm{comb}} = \mathrm{Pres}_{\mathrm{intent}} \cdot \mathrm{Pres}_{\mathrm{Lev}}.$$

Finally, we summarize the trade off between attribution and preservation using their harmonic mean (Gao et al., 2023a):

$$\mathrm{F1}_{\mathrm{AP}} = \frac{2 \cdot \mathrm{Attr} \cdot \mathrm{Pres}_{\mathrm{comb}}}{\mathrm{Attr} + \mathrm{Pres}_{\mathrm{comb}}},$$

where the term Attr can be instantiated by the human based $Attr_AIS$ or the automatic version $Attr_auto$.

Safety. We evaluate safety by measuring the propensity of the model to generate toxic content under both single turn and multi turn settings.

In single turn prompted generation, we follow RealToxicityPrompts (Gehman et al., 2020) and quantify the propensity of a model to degenerate into toxic continuations along two complementary axes. The first axis measures how toxic the worst case sampled continuation can be. The second axis measures how frequently toxicity appears across repeated sampling. For each prompt j, we sample K continuations and score each generation with a toxicity function $t(\cdot)$ such as Perspective API. We treat a span as toxic when $t(\cdot)$ is greater than or equal to a threshold τ. Following RealToxicityPrompts, we set the threshold τ to 0.5 for the toxicity score (Gehman et al., 2020).

Generation method	**Validity**	**Fidelity**	**Diversity**	**Faithfulness**	**Safety**
FineWeb2 (Penedo et al., 2025)	×	△	△	×	×
Dolma (Soldaini et al., 2024)	×	△	△	×	✓
RedPajama (Weber et al., 2024)	△	△	△	×	△
Self-Instruct (Wang et al., 2023b)	△	△	△	×	×
RARR (Gao et al., 2023a)	×	×	×	✓	×
PMI-FAITH (Nandwani et al., 2023)	×	✓	×	✓	×

Table 2: Whether representative text control/data-construction/alignment methods explicitly evaluate each dimension. ✓: explicitly evaluated; △: partially/indirectly covered; ×: not reported or not applicable.

We then report the Toxicity Probability and the Expected Maximum Toxicity. The Toxicity Probability is defined as the empirical probability that at least one of the K samples crosses the threshold. The Expected Maximum Toxicity is defined as the average worst case toxicity across prompts (Gehman et al., 2020):

$$\mathrm{TP} = \frac{1}{N}\sum_{j=1}^{N}\mathbb{I}\left[\max_{1\le\ell\le K} t(g_{j,\ell}) \ge \tau\right], \qquad \mathrm{EMT} = \frac{1}{N}\sum_{j=1}^{N}\max_{1\le\ell\le K} t(g_{j,\ell})\,.$$

In multi turn interactions, we adopt dialogue level safety measurements from ToxicChat (Chen et al., 2023a). These measurements define risk over an entire conversation rather than an isolated response. Let $\mathcal{D}_{\mathrm{dlg}}$ be a set of dialogues. For each dialogue d in the set $\mathcal{D}_{\mathrm{dlg}}$, we let R_d denote the set of all chatbot responses and let $q_{d,i}$ and $r_{d,i}$ represent the query response pair at turn i. We compute the Toxic Sentence Generation rate, which we refer to as TSG, as the percentage of conversations in which the chatbot generates at least one toxic response during the interaction (Chen et al., 2023a):

$$\mathrm{TSG} = \frac{1}{|\mathcal{D}_{\mathrm{dlg}}|}\sum_{d\in\mathcal{D}_{\mathrm{dlg}}}\mathbb{I}\left[\max_{r\in R_d} t(r) \ge \tau\right].$$

To isolate cases where toxicity is elicited even with non toxic user inputs, we report the Non Toxic to Toxic rate, which we refer to as NT2T. This metric represents the percentage of conversations in which the chatbot produces at least one toxic response at any point during the interaction while the corresponding user query is non toxic (Chen et al., 2023a):

$$\mathrm{NT2T} = \frac{1}{|\mathcal{D}_{\mathrm{dlg}}|}\sum_{d\in\mathcal{D}_{\mathrm{dlg}}}\mathbb{I}[\exists i:\ t(q_{d,i}) < \tau \ \wedge\ t(r_{d,i}) \ge \tau]\,.$$

This dialogue level NT2T definition is consistent with prior turn level toxicity categorization of query response pairs into NT2T, NT2NT, T2T, or T2NT. Within this classification, NT2T specifically denotes a non toxic query that elicits a toxic response (Si et al., 2022).

Finally, to ensure safety interventions do not degrade linguistic quality, we report Perplexity and Distinct N (Liang et al., 2024; Jozefowicz et al., 2016; Li et al., 2016).

2.4 Evaluation Practice Gap

We analyze representative methods from Section 2.1 and categorize their reported evaluation protocols in Table 2.

Validity. As shown in Table 2, validity is rarely evaluated using explicit or reproducible metrics across text generation methods. Most studies overlook specific scores for grammatical acceptability or dialogue quality. Instead, they rely on downstream benchmarks or human judgments as general proxies for quality. However, these external signals often fail to identify specific validity errors. Even with improved models, large-scale automated pipelines can still suffer from basic structural breakdowns, including encoding noise, formatting violations, and truncation. Therefore, explicit monitoring of validity remains a practical necessity.

Faithfulness. Faithfulness describes how the output of LLMs matches the original source material. This property is one of the most important factors in solving the problem of hallucinations. When the ability of LLMs to find exact evidence is improved, the models are much less likely to produce fabricated information. However, Table 2 shows that some studies do not evaluate this dimension. We provide several metrics that can be used to measure faithfulness in these models.

Safety. Safety reporting is largely absent in the surveyed methods. Few studies adopt standardized protocols, such as Perspective API or Toxic List, to measure toxicity score or banned words. Consequently, future work on synthetic text should incorporate safety evaluation as a standard reporting component. By analyzing safety alongside quality indicators, researchers can better manage the trade-off between safety and utility. This approach is essential to prevent safety regressions even as performance improves.

2.5 Usages

In this section, we shift our focus from the methods of text and dialogue generation to the practical applications of the resulting synthetic corpora. We focus on offline uses, which refer to settings where model outputs are treated as static artifacts for training and evaluation. This approach is distinct from online prompting or control during the decoding process. Synthetic corpora are most effective when they are carefully curated and aligned with target deployment distributions. These corpora should also be explicitly stress tested according to the quality and trustworthiness dimensions of our taxonomy. We first discuss training time uses such as data augmentation, alignment, and adaptation, and then we describe evaluation uses.

Supervised Data Augmentation and Task Transfer. A primary use case is to augment supervised training data in low or no resource regimes. For machine reading comprehension, synthetic question answer pairs generated from unlabeled passages can substantially improve downstream accuracy when filtered rather than used wholesale. Early work showed that LLMs can produce synthetic QA corpora that, by themselves, support competitive models on benchmarks such as SQuAD (Puri et al., 2020). Additionally, end to end synthetic QA generation can further enable domain adaptation when paired with filtering and validation (Shakeri et al., 2020).

More recent approaches integrate LLM based rewards or selectors to identify high value examples. For instance, Jin & Wang (2024) treat a generative LLM as a reward model to score synthetic QA pairs and retain only those predicted to be most useful for training. Survey evidence on data selection for instruction tuning suggests diminishing returns from scaling instruction corpora alone. This motivates data-efficient selection based on utility signals such as quality, diversity, difficulty, and complexity. Empirically, highly curated small sets such as LIMA with 1000 instances and large-scale human AI collaboration corpora such as OpenAssistant conversations can be competitive. Furthermore, automated selectors improve efficiency through model-based filtering and strong LLM scoring using models such as GPT 4 to remove low quality or incorrect examples and retain diverse and high utility instances (Albalak et al., 2024; Köpf et al., 2023).

Instruction Tuning and Alignment Corpora. Beyond task labels, synthetic text and dialogue are widely used to construct instruction following and preference datasets for alignment. Large-scale instruction tuning corpora routinely mix human-written and LLM-written instructions, demonstrations, and task variants. The synthetic components expand coverage over skills, formats, and domains, while human labor focuses on seeding and spot checking. Preference and critique datasets such as UltraFeedback use a strong LLM like GPT 4 to provide multi-dimensional scores and natural language feedback on candidate responses. This yields large synthetic preference corpora for reward modeling and preference-based finetuning (Cui et al., 2024). In practice, synthetic instructions, responses, and AI-generated feedback are often combined into pipelines where models both propose and evaluate data. This process blurs the line between augmentation and alignment.

Domain Adaptation and Retrieval Grounded Systems. In domain-specific applications, synthetic corpora support both task adaptation and retrieval augmented generation. For question answering in specialized domains such as biomedical, practitioners often synthesize in-domain questions and answers from domain passages or corpora. They then filter low-quality generations to improve robustness under domain shift (Shakeri et al., 2020). These synthetic examples are then used to finetune general purpose language models. This process simplifies earlier multi-stage synthetic question answering pipelines for domain adaptation (Shakeri et al., 2020). In retrieval grounded setups, model generated query document or query identifier pairs can be used to train domain specific retrievers. This approach improves ranking through hard negative mining and preference learning on distributions that better match deployment scenarios (Wen et al., 2025). Because synthetic queries can be controlled through granularity and domain specific constraints, they provide controllable testbeds for measuring retrieval faithfulness and downstream task gains.

Synthetic Evaluation Sets and Judge Labeled Benchmarks. Once models are trained or adapted, synthetic text and dialogue are increasingly treated as evaluation artifacts. Instead of relying solely on static human written benchmarks, practitioners construct synthetic test suites that probe specific failure modes under controlled distributions. These modes include compositional reasoning, long context consistency, and safety constraints.

The LLM as a judge pipeline can score large pools of model outputs using fine-grained criteria such as helpfulness, safety, and faithfulness. This approach supports the rapid construction of synthetic evaluation sets as well as public leaderboards and meta evaluation benchmarks such as MT-Bench and Chatbot Arena (Gu et al., 2025; Zheng et al., 2023). Self-verification methods such as Chain-of-Verification explicitly generate verification questions and independent checks to reduce hallucination(Dhuliawala et al., 2024). At the same time, sampling-based approaches like SelfCheckGPT produce alternative generations whose consistency can be logged as a black box signal for hallucination and robustness (Manakul et al., 2023).

Safety Testing and Privacy Preserving Surrogates. Finally, synthetic corpora play a growing role in risk management. Privacy studies exploit LLM generated variants of sensitive text as surrogates that retain statistical utility while reducing direct exposure of personally identifiable information. Memorization and extraction attacks demonstrate that models can regurgitate rare training examples, including personally identifiable information. This motivates careful deduplication and privacy-aware data handling (Carlini et al., 2021). SRD (Zhang et al., 2026) framework utilizes LLMs to generate toxic and good text and uses this contrast dataset to detoxify the model. For provenance and attribution, organizations increasingly embed watermarks or content credentials into synthetic corpora to support downstream detection and policy enforcement. Keyed statistical watermarks allow reliable statistical detection of model-generated text and are commonly evaluated under robustness and security considerations (Kirchenbauer et al., 2023; Coalition for Content Provenance and Authenticity, 2025). By framing fingerprint injection as a knowledge-editing problem over fingerprint text pairs, RFEdit enables efficient and robust injection with minimal impact on unrelated knowledge (Li et al., 2025c). Synthetic safety and red teaming corpora are likewise used to stress test models and safety filters. These corpora consist of adversarial prompts, toxic continuations, or jailbreak dialogues generated by or with LLMs.

Overall, LLM generated text and dialogue serve as flexible building blocks for training, adapting, evaluating, and governing language technologies. Their benefits are most pronounced when synthetic corpora are aligned with target tasks and distributions. Furthermore, these corpora should be coupled with strong selection and verification mechanisms and instrumented with provenance and privacy safeguards that make their quality and trust properties measurable.

3 Symbolic and Logical Data

Symbolic and logical data generation has gained significant attention with the development of large reasoning models. These models emphasize multi-step inference and structured intermediate reasoning which are often produced through chain-of-thought prompting (Wei et al., 2023; Kojima et al., 2023; Wang et al., 2023a; Liu et al., 2025b; Morishita et al., 2024; Toshniwal et al., 2024). Unlike general data augmentation, this field focuses on creating instances that are intrinsically checkable. Each example typically consists of a well-specified problem, a structured reasoning trace or intermediate artifacts, and an explicit verification signal. These signals include exact keywords (Li et al., 2025d) or answer matching in mathematics word problems such as tasks in the GSM8K dataset (Cobbe et al., 2021) . They also include compilation and unit tests in code generation or logical validity under symbolic rules (Tafjord et al., 2021; Morishita et al., 2024; Liu et al., 2025b).

In practice, modern pipelines based on LLMs do more than generate diverse problem and solution pairs. They also propose intermediate steps, critique and revise candidate solutions, and filter outputs using self-consistency (Wang et al., 2023a) or external tools (Ahmad et al., 2025). These pipelines help scale reasoning supervision beyond datasets written entirely by humans. They produce large quantities of verified or partially verified trajectories for distillation and post-training. Recent examples of this approach include reasoning models trained with reinforcement learning or augmented by tools and verifiers (OpenAI et al., 2024; DeepSeek-AI, 2025).

3.1 Generation Methods

In this section, we introduce how to generate symbolic and logic data by using LLMs.

Heuristic Evolution Methods for Expanding Reasoning Tasks. A primary category of reasoning data generation driven by LLMs relies on the iterative evolution of seed tasks. In this approach, strong models repeatedly rewrite problems to increase structural diversity and difficulty while attempting to preserve the validity of the solutions. This pattern originates from the rewriting techniques used in Evol-Instruct (Xu et al., 2024a) and has been adapted to domains such as mathematics and programming. For example, WizardMath applies a mathematics-specific evolution pipeline combined with reinforcement learning from evolution feedback to strengthen step-by-step reasoning (Luo et al., 2025a). Similarly, MetaMath scales up mathematics corpora by generating and diversifying data in the form of questions, answers, and rationales (Yu et al., 2024a).

In the field of programming, WizardCoder adapts instruction evolution to coding tasks by generating progressively harder instructions and responses from simpler seeds (Luo et al., 2025b). Across these studies, the core idea remains the same. The process starts from existing benchmarks or seed instructions and then applies controlled mutations such as paraphrasing, adding constraints, or making the problems more compositional. Finally, the process uses lightweight heuristics to maintain semantic accuracy. These heuristics often involve checking the agreement of answers, verifying basic formats, or screening for consistency.

Tool-Verified Generation Methods with Solvers, Executors, and Provers. A second approach combines synthesis from LLMs with programmatic verification. This method uses executors, compilers, unit tests, or symbolic checkers as reliable oracles. In mathematical reasoning, OpenMathInstruct-1 synthesizes large-scale problem and solution pairs by generating solutions in the style of a code interpreter. It retains instances where the computations are executable and consistent (Toshniwal et al., 2024).

In the field of programming, OpenCodeInstruct assembles large instruction tuning corpora with execution feedback and unit test signals. This strengthens acceptance criteria beyond surface plausibility (Ahmad et al., 2025). For formal logical reasoning, datasets and generators such as ProofWriter and ALT-FLD×2 construct instances where the validity is grounded in formal rules. This enables the deterministic checking of entailment or proof steps under symbolic constraints (Tafjord et al., 2021; Morishita et al., 2024). SynLogic further advances scalable and verifiable logic synthesis by generating diverse logical tasks whose correctness can be verified by simple rule-based checkers (Liu et al., 2025b). In all these cases, the models propose candidates while external tools act as gatekeepers that filter for correctness and logical structure.

Trajectory Harvesting via Rewarded Rollouts and Distillation. A third category of methods generates reasoning data by collecting trajectories from strong teacher models or reasoners trained through reinforcement learning. These models interact with tasks that allow for verification. In this context, trajectories serve as more than just explanations. They are the results of a policy exploring solution spaces guided by reward and verification signals. These trajectories are then distilled into student models.

Recent reasoning model pipelines follow this approach by collecting large volumes of partially verified rollouts. These rollouts are filtered using automated verifiers and learned judges and are then used as signals for downstream post-training. Similarly, public releases such as SYNTHETIC-2 provide large-scale datasets that are the results of reasoning agents working with verifiers across various task families (Prime Intellect Team, 2025). This approach produces verified reasoning traces that are suitable for distillation. The key difference in this method lies in the nature of the data. The data are policy traces generated during search and optimization processes instead of static and one-shot synthetic question and answer pairs.

Preference-Curated Generation via LLM as a Judge. Finally, in scenarios where rigid programmatic checks are incomplete or unavailable, many pipelines rely on preference curation and supervision using LLMs as judge mechanisms. UltraFeedback constructs large scale preference signals by sampling multiple candidate responses and scoring them using a superior model. These scores are based on dimensions such as helpfulness, truthfulness, and instruction adherence (Cui et al., 2024).

CodeUltraFeedback applies a similar concept to the programming domain. In this context, judges assess candidate solutions based on coding preferences such as readability and adherence to instructions (Weyssow et al., 2024). Related self training approaches, such as STaR, create rationales through generate and filter loops where correctness serves as a general acceptance signal (Zelikman et al., 2022). Collectively, these methods emphasize model driven evaluation. Reasoning data are generated at scale and then organized by judge models into ranked or preference annotated corpora. This process supports alignment, preference optimization, and self improvement.

3.2 Quality Metrics for Symbolic and Reasoning Data

The fundamental criterion for evaluating the quality of reasoning data generated by LLMs is objective correctness. This requires that both intermediate reasoning steps and final conclusions serve as logically valid and empirically verifiable results. Given that reasoning data encompass mathematical, symbolic, programmatic, and open-domain contexts, the specific verification protocols vary by domain. However, these protocols share a unified objective. They validate that generated reasoning traces adhere strictly to ground truth logic or executable truth conditions.

Validity. For mathematical and code based reasoning, correctness can often be validated algorithmically through domain verifiers. Let $\mathcal{R} = \{(q_i, c_i, y_i)\}_{i=1}^{N}$ denote reasoning examples, where q_i is the problem, c_i is the generated rationale such as a chain of thought or a code interpreter trace, and y_i is the final answer. When a reference answer $y_i^\star$ or an executable specification is available, we define a domain specific checker f_{check} as follows:

$$f_{\text{check}}(q_i, c_i, y_i) = \begin{cases} 1, & \text{if the candidate is verified as correct for } q_i; \\ 0, & \text{otherwise.} \end{cases}$$

We then report the overall verification accuracy as

$$\text{Acc}_{\text{verify}} = \frac{1}{N} \sum_{i=1}^{N} f_{\text{check}}(q_i, c_i, y_i).$$

In MetaMath, the answer augmentation stage uses rejection sampling. In this process, diverse reasoning paths are generated and only those yielding correct answers are retained. This method ensures validity through answer level verification (Yu et al., 2024a).

In OpenMathInstruct-1, solutions are represented in a code interpreter style format. Correctness is enforced by keeping only solutions that lead to the ground truth answer. The paper also uses training set coverage measured as pass at k. This metric identifies whether any of k sampled solutions reaches the ground truth (Toshniwal et al., 2024).

For program synthesis and code reasoning, validity is commonly measured by unit test execution. Let $\mathcal{T}_i$ be the test suite for example i. The unit test pass rate for a candidate solution is defined as follows:

$$\text{PassRate}_i = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \mathbb{I}\big(\text{exec}(y_i, t) = \texttt{pass}\big),$$

The dataset level pass rate is calculated as $\text{PassRate} = \frac{1}{N} \sum_{i=1}^{N} \text{PassRate}_i$. This approach aligns with OpenCodeInstruct, which executes solutions on generated unit tests and records pass rates as metadata (Ahmad et al., 2025).

For logical and deductive reasoning with explicit proofs such as proof graphs, validity requires correct entailment prediction and a correct proof. Following ProofWriter, proof correctness is evaluated using a strict metric called Full Accuracy. In this metric, the predicted proof graph must exactly match a gold proof. If the graphs do not match, the proof scores zero (Tafjord et al., 2021).

Accordingly, let $y_i^\star$ be the gold entailment or answer label and let $c_i^\star$ be a gold proof representation. The strict proof accuracy can be summarized as

$$\text{Acc}_{\text{proof}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i = y_i^\star) \cdot \mathbb{I}(c_i = c_i^\star).$$

Finally, for robustness style validity checks, FAIRR (Sanyal et al., 2022) defines consistency over an equivalence set of perturbed inputs. For a theory and statement pair (T, s) and its equivalence set $E(T, s) = \{(T'_k, s'_k)\}_{k=1}^{K}$, consistency is defined by the following equation:

$$C(T, s) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\big(f(T, s) = f(T'_k, s'_k)\big),$$

This value is averaged over the dataset. FAIRR reports entailment accuracy, strict proof accuracy, and consistency (Sanyal et al., 2022).

Fidelity. Fidelity measures the alignment between proxy evaluation signals and actual quality standards. Unlike validity which assesses the objective correctness of a solution against a ground truth, fidelity evaluates the reliability of the tools or models used to judge the data. This metric ensures that automated evaluations reflect true human reasoning or logical consistency when direct verification is not possible.

Following the self consistency decoding paradigm, multiple reasoning paths are sampled and the most consistent answer is selected (Wang et al., 2023a). For each question q_i, we sample K chains $\{c_{i,k}\}_{k=1}^{K}$ and extract their final answers $\{y_{i,k}\}_{k=1}^{K}$. We define the majority vote answer as follows:

$$\hat{y}_i = \arg\max_{y} \sum_{k=1}^{K} \mathbb{I}\big(y_{i,k} = y\big).$$

To measure the alignment between the consensus proxy and the reference label, we compute the self consistency accuracy:

$$\text{Acc}_{\text{SC}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\big(\hat{y}_i = y_i\big).$$

To further quantify the internal consistency of sampled answers for the same question, we use the average pairwise answer agreement:

$$\text{Agree}(q_i) = \frac{1}{K(K-1)} \sum_{k \neq k'} \mathbb{I}\big(y_{i,k} = y_{i,k'}\big).$$

For open ended natural language reasoning where executable checking is often not possible, judges based on LLMs provide automated scoring signals. For example, GPT 4 is frequently used to provide these evaluations. To evaluate how faithfully these judge signals reflect human judgments, let $u_i^{(\mathrm{LLM})}$ and $u_i^{(\mathrm{human})}$ denote scalar scores from a model judge and a human annotator for the i-th example.

We measure the rank association between these scores using the Spearman rho correlation or the Pearson correlation. This follows established evaluation methods for alignment in model judges (Lai et al., 2025):

$$\rho_{\text{LLM-human}} = \mathrm{corr}_{\mathrm{Spearman}}\big(u^{(\mathrm{LLM})}, u^{(\mathrm{human})}\big).$$

When the judge outputs M rubric dimensions such as coherence, faithfulness, and factuality, we can aggregate them into an overall proxy quality score:

$$Q_{\mathrm{reason}} = \frac{1}{M}\sum_{m=1}^{M} \mathrm{score}_{i,m}.$$

The reliability of model judge signals is commonly validated on benchmarks such as JudgeLM and MT Bench or the Chatbot Arena. These benchmarks report the agreement between model judges and human preferences (Zhu et al., 2025; Zheng et al., 2023).

Preference datasets based on AI feedback such as UltraFeedback (Cui et al., 2024) and pipelines using reinforcement learning from AI feedback provide training signals for preference modeling and downstream alignment. These pipelines replace expensive human labels with preferences generated by LLMs (Lee et al., 2024). On pairs labeled with preferences, a reward model R_ϕ can be evaluated by whether it ranks the preferred output higher.

We consider a pair $(c_{i,a}, c_{i,b})$ with a binary preference label z_i that is either zero or one. For example, z_i equals one if $c_{i,a}$ is preferred. We define the label implied by the model as follows:

$$\hat{z}_i = \mathbb{I}\big(R_\phi(c_{i,a}) > R_\phi(c_{i,b})\big).$$

We then compute the pairwise preference accuracy:

$$\mathrm{Acc}_{\mathrm{RM}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(\hat{z}_i = z_i\big).$$

This metric evaluates reward models on pairwise preference data used in training and benchmarking for reinforcement learning from human feedback (Bai et al., 2022; Frick et al., 2024).

3.3 Trustworthy Metrics for Symbolic and Reasoning Data

Even when a generated reasoning trace yields a correct final answer, the logical coherence and faithfulness of intermediate steps are critical. A model that reaches the right answer without valid reasoning introduces spurious patterns and unreliable supervision.

Faithfulness. Let $c_i = [c_{i,1}, \ldots, c_{i,T_i}]$ be the chain of thought for example i. A step verifier f_{step} checks each step. This verifier could be a symbolic executor in formal domains or an inference based entailment checker for natural language. It evaluates each step as follows:

$$f_{\mathrm{exec}}(c_{i,t}) = \begin{cases} 1, & \text{if } c_{i,t} \text{ is a valid transformation,} \\ 0, & \text{otherwise.} \end{cases}$$

The step validity rate is defined by the following equation:

$$\mathrm{Val}_{\mathrm{step}} = \frac{1}{\sum_i T_i} \sum_i \sum_{t=1}^{T_i} f_{\mathrm{exec}}(c_{i,t}).$$

This metric measures how often intermediate steps obey mathematical, logical, or semantic rules. Correctness metrics at the step level have been explored in reasoning evaluation frameworks such as ReCEval. This framework scores the quality of steps and chains through textual entailment and informativeness (Prasad et al., 2023). Furthermore, causal faithfulness analyses such as FRODO test whether the content of a chain of thought has a causal influence on the final answer rather than simply correlating with it (Paul et al., 2024).

Besides executable step verification, entailment models can provide a proxy for step support in natural language. For each pair of adjacent steps $c_{i,t-1}$ and $c_{i,t}$, an entailment scorer g_{entail} provides an output as follows:

$$p_{i,t} = g_{\mathrm{entail}}(c_{i,t-1} \Rightarrow c_{i,t}),$$

We calculate the average entailment alignment to represent the support within the chain:

$$\mathrm{Align}_{\mathrm{entail}} = \frac{1}{\sum_i (T_i - 1)} \sum_i \sum_{t=2}^{T_i} p_{i,t}.$$

Higher values indicate that the reasoning progresses through steps supported by semantics and logic instead of abrupt or contradictory jumps. Evaluation methods using entailment have been used to assess the correctness and informativeness of reasoning chains (Prasad et al., 2023). These methods have also been applied to assign partial credit based on textual entailment (Yao & Barbosa, 2024).

Robustness. To study generalization under a shift in distribution, we compare the pairwise accuracy of reward models on in domain evaluations and out of domain evaluations. This difference is defined as follows:

$$\Delta_{\mathrm{OOD}} = \mathrm{Acc}_{\mathrm{RM}}^{\mathrm{in}} - \mathrm{Acc}_{\mathrm{RM}}^{\mathrm{out}}.$$

Recent evaluations report these comparisons by measuring in domain accuracy on preference data such as UltraFeedback. They also measure out of domain accuracy on benchmarks such as RewardBench (Mahan et al., 2024; Lambert et al., 2024). A smaller difference indicates more stable preference ranking across different domains.

3.4 Evaluation Practice Gap

We analyze representative methods from Section 3.1 and categorize their reported evaluation protocols in Table 3.

Table 3 highlights a significant disparity in the evaluation landscape for symbolic and logical data generation. Specifically, dimensions that assess *process quality* are covered substantially less than those measuring simple answer-level correctness.

Faithfulness. Explicit evaluation of faithfulness is mainly limited to benchmarks that contain structured proof annotations. In these benchmarks, strict proof correctness can be directly measured. For example, ProofWriter reports full proof accuracy based on exact proof graph matching (Tafjord et al., 2021). Similarly, FaiRR includes strict proof accuracy as a core part of its protocol (Sanyal et al., 2022). In contrast, many generation pipelines that rely on answer verification or filtering based on execution prioritize the correctness of the final answer. Notable examples include MetaMath (Yu et al., 2024a), OpenMathInstruct 1 (Toshniwal et al., 2024), and OpenCodeInstruct (Ahmad et al., 2025). As a result, the validity of intermediate reasoning steps and the question of whether they truly support the conclusion often remain under evaluated.

Generation method	**Validity**	**Fidelity**	**Robustness**	**Faithfulness**
MetaMath (Yu et al., 2024a)	✓	×	×	×
OpenMathInstruct-1 (Toshniwal et al., 2024)	✓	✓	×	×
OpenCodeInstruct (Ahmad et al., 2025)	✓	△	×	×
UltraFeedback (Cui et al., 2024)	×	✓	×	×
ProofWriter (Tafjord et al., 2021)	✓	×	△	✓
FaiRR (Sanyal et al., 2022)	✓	×	△	△

Table 3: Whether representative Symbolic and Logical Data generation methods explicitly evaluate each dimension in their experimental sections. ✓: explicitly evaluated; △: partially/indirectly covered; ×: not reported or not applicable.

Robustness. Robustness is reported in fewer studies. When researchers address this property, it appears in different forms that are not directly comparable. For instance, ProofWriter evaluates out of domain transfer across rule sets (Tafjord et al., 2021). In another case, FaiRR reports consistency based on perturbations alongside entailment and accuracy (Sanyal et al., 2022). Beyond these specific examples, most research related to generation focuses mainly on performance within the same domain. These studies rarely treat changes in data distribution or stability under perturbations as their main evaluation goals. This lack of standardization shows that there is a need to define specific robustness metrics that are suitable for symbolic and logical generation. Researchers should avoid relying only on general proxies for robustness.

Collectively, these gaps suggest that evaluations focused on correctness may obscure the true quality of reasoning when the goal is to generate data with reliable intermediate structures. Therefore, adding measurements centered on faithfulness and robustness to standard validity metrics is very important for symbolic and logical reasoning. In this domain, the desired behavior is not just arriving at the correct answer but doing so through stable and interpretable reasoning processes.

3.5 Usage

Pre Training and Continual Pre Training with Synthetic Corpora. A primary application of synthetic reasoning data involves integrating reasoning trajectories directly into the foundation model stage. In this paradigm, synthetic corpora are integrated during pretraining or continual pretraining to provide models with structured and multiple step reasoning capabilities prior to instruction tuning (Yang et al., 2024a; Ying et al., 2024). This approach views reasoning as a fundamental skill. In this context, symbolic, arithmetic, and procedural inference patterns are acquired alongside natural language. This process helps the internal representations of the model move toward compositional reasoning.

Recent advancements in large scale reasoning models such as DeepSeek R1 and OpenAI o1 confirm the effectiveness of optimization after pretraining. These frameworks build upon pretrained base models and use large scale reinforcement learning to produce long horizon and step wise problem solving behaviors. This process creates models specialized in reasoning instead of relying on superficial post hoc fine tuning (DeepSeek-AI, 2025; OpenAI et al., 2024). Specifically, DeepSeek R1 uses rule based and verifiable outcome rewards to provide precise feedback for domains such as mathematics and coding. This enables scalable improvements through automatic correctness signals (DeepSeek-AI, 2025). Similarly, OpenAI o1 emphasizes large scale reinforcement learning to enhance careful reasoning and the effective use of chain of thought methods. This achieves strong performance across benchmarks in mathematics, coding, and science (OpenAI et al., 2024).

In parallel, families specialized in mathematics such as Qwen Math and InternLM Math adopt a strategy of continued pretraining derived from strong base models. These methods incorporate math focused corpora followed by downstream post training. This is typically followed by downstream post-training, e.g., supervised fine-tuning with chain-of-thought data and tool-augmented/verifier-guided supervision, as well as reward-model-based reranking or filtering of multi-step solutions. Together, this continued-pretraining post-training pipeline improves mathematical reasoning through specialization (Yang et al., 2024a; Ying et al., 2024). Furthermore, related data pipelines synthesize new problems and solutions at scale while using screening based on correctness to retain only solutions where the final answer matches a ground truth. This methodology is shown by MetaMathQA and OpenMathInstruct-1. These systems use strong LLMs to generate candidate solutions while using verification based on execution to validate reliability (Yu et al., 2024a; Toshniwal et al., 2024). Early pipelines often used closed source models while later work uses increasingly strong open models. Collectively, these efforts treat synthetic reasoning corpora as primary components of the training distribution. By exposing the underlying language model to step wise mathematical and logical derivations, these methods improve data efficiency during later specialization stages.

Supervised Fine Tuning on Verified Reasoning Traces. A distinct application of reasoning data generated by LLMs involves explicitly providing chain of thought capabilities through supervised fine tuning. This process uses verified reasoning triplets consisting of an input, a reasoning trace, and a final answer. Unlike pretraining which primarily shapes inductive biases, supervised fine tuning aligns model outputs with concrete multiple step examples. This encourages the model to internalize intermediate problem solving procedures.

In the domain of mathematical reasoning, synthetic datasets fine tune open models using explicit verification signals although the specific mechanisms vary. MetaMathQA scales data production through bootstrapped question generation and uses rejection sampling to filter reasoning traces based on the correctness of the final answer (Yu et al., 2024a). OpenMathInstruct 1 emphasizes verification based on execution by allowing solutions to combine natural language reasoning with executable code. This method uses execution by an interpreter as a validation signal (Toshniwal et al., 2024). Similarly, releases such as the synthetic GSM8K dataset by Gretel include structured reflection style instances paired with automated validation. Other variants incorporate rigorous automated checks including model judge evaluations and symbolic verification through tools such as sympy to screen and refine candidates (AI, 2024). WizardMath extends beyond conventional supervised fine tuning by synthesizing complex instructions and applying reinforcement learning from evolved instructions feedback to strengthen reasoning behaviors beyond simple imitation (Luo et al., 2025a).

In the programming domain, corpora such as CodeAlpaca and Magicoder provide instruction and solution pairs for code generation. Magicoder further anchors synthesis in open source code snippets to better reflect realistic development scenarios (Chaudhary, 2023; Wei et al., 2024). OpenCodeInstruct improves this paradigm with explicit test suites and execution outcomes along with quality assessments based on models. This provides verifiable signals that support data curation during training with synthetic programs (Ahmad et al., 2025).

Regarding logical and symbolic reasoning, frameworks such as ALT construct principled synthetic logic corpora. These corpora consist of multiple step deductive instances generated by programs and grounded in formal logic. This data facilitates supplementary training to improve entailment and inference patterns. In parallel, SynLogic synthesizes diverse logic tasks paired with rule based verifiers to enable scalable training with verifiable feedback (Morishita et al., 2024; Liu et al., 2025b).

In summary, supervised fine tuning on verified reasoning traces offers a controllable method for using synthetic data. Generation pipelines propose candidate problems and explanations while external verifiers provide correctness signals. The resulting validated instances are then used to train models that are capable of explaining and justifying their reasoning step by step.

Optimization via Preference Modeling and Reinforcement Learning. A significant application of synthetic reasoning data lies in providing evaluative feedback rather than direct supervision. This process encourages robust reasoning behaviors through preference modeling and reinforcement learning. In this context, reasoning traces generated by LLMs or evaluated by verifiers function as inputs to reward models or policy optimization objectives. These objectives explicitly prioritize logical consistency, correctness, and clarity within multiple step inference.

For instance, UltraFeedback constructs multiple domain preference datasets by sampling candidate responses from diverse models and employing a superior LLM judge to rank them according to specific criteria. These criteria include instruction adherence and truthfulness. This process yields large scale AI feedback signals suitable for reward model training and optimization styles similar to reinforcement learning from human feedback (Cui et al., 2024). Approaches such as reinforcement learning from AI feedback extend this methodology by substituting human preference labels with comparisons judged by LLMs. This substantially enhances the scalability of preference data collection across tasks and domains (Lee et al., 2024).

Within the reasoning domain, contemporary systems increasingly integrate reinforcement learning with programmatically verifiable feedback for mathematical, coding, and logical problems. Specifically, DeepSeek R1 uses rule based rewards and executable feedback to score reasoning trajectories during training. This feedback includes actions such as compiling and executing code to enable policy updates that reinforce verifiable correctness (DeepSeek-AI, 2025). OpenAI o1 incorporates large scale reinforcement learning to refine multiple step reasoning behaviors, although the specific implementation details of the reward and verifier mechanisms remain undisclosed (OpenAI et al., 2024). Furthermore, the SYNTHETIC-2 corpus provides millions of reasoning traces including reinforcement learning rollouts accompanied by reward signals. These signals facilitate downstream distillation and offline reinforcement learning research (Prime Intellect Team, 2025). Consequently, reasoning data evolve from static training examples into dynamic evaluative signals. Synthetic traces combined with judge or verifier outputs define a reward landscape that guides iterative model refinement.

Knowledge Distillation and Implicit Reasoning Transfer. A complementary application of reasoning data generated by LLMs involves the compression of explicit reasoning into implicit internal representations. This process distills detailed chain of thought traces from complex teacher models into compact student models. These student models are capable of performing robust reasoning without generating extensive explanations during inference. In this paradigm, reasoning traces function as latent supervision that guides representation learning rather than serving only as textual targets.

Methods for implicit chain of thought distillation such as those proposed by Deng et al. (2023) train a teacher model using explicit supervision. They subsequently distill its reasoning competence into a student model through hidden state learning signals. This enables the creation of efficient models that reason implicitly without producing lengthy traces. In symbolic and deductive contexts, ProofWriter provides structured supervision over natural language proofs. This approach offers process level signals that support the training and potential distillation of lightweight models capable of following proof like reasoning patterns with minimal explicit rationales (Tafjord et al., 2021).

More broadly, large scale reasoners trained with reinforcement learning and trajectory corpora such as DeepSeek R1 and SYNTHETIC-2 provide rich multiple step rollouts that serve as distillation targets. These resources enable compact models to inherit high level reasoning behaviors from computationally intensive teacher models (DeepSeek-AI, 2025; Prime Intellect Team, 2025). Therefore, synthetic reasoning data facilitate not only explicit chain of thought generation but also implicit reasoning transfer. The goal is to embed process level competence into models that yield correct answers without necessarily explaining every intermediate step.

Evaluation, Benchmarking, and Generalization Testing. Finally, synthetic and programmatically structured reasoning corpora increasingly serve as evaluation instruments. These corpora offer diverse and automatically verifiable tasks to test generalization capabilities beyond static benchmarks curated by humans. The primary objective is the construction of dynamic benchmarks that evolve alongside model capabilities while maintaining rigorous and verifiable scoring protocols.

In the domain of mathematics, GSM-HARD extends the GSM8K dataset by systematically substituting numerical values with larger or more complex alternatives. This process creates a stress test suite generated by programs to evaluate arithmetic robustness. When integrated with programmatic execution solvers, model outputs can be verified automatically at scale (Gao et al., 2023b). Although not generated by LLMs, SciBench aggregates scientific problems at the college level in physics, chemistry, and mathematics. This enables systematic evaluation and detailed error analysis of scientific problem solving skills (Wang et al., 2024b).

In the fields of causal and logical reasoning, CREPE introduces a benchmark for assessing causal reasoning regarding event plausibility and entity states. This benchmark uses human judgments to identify discrepancies between model behavior and human reasoning (Zhang et al., 2023). Additionally, SynLogic provides held out validation splits for synthetic logical tasks. It reports performance using variance reducing metrics such as the average at eight (Liu et al., 2025b). Collectively, these evaluation frameworks demonstrate the dual utility of synthetic and programmatically structured reasoning datasets. They not only provide training and reinforcement learning signals but also support benchmarks for monitoring reasoning robustness, distribution shifts, and cross domain generalization as models advance.

4 Tabular Data

Tabular data consists of structured records organized according to a fixed schema of heterogeneous features that include rows and columns. As described in the survey by Shi et al. (2025b), the generative modeling of this data requires the joint representation of mixed type attributes including both numerical and categorical domains as well as complex dependencies between columns. At the same time, the model must preserve statistical fidelity to the original source distribution. Validity in this context is defined by a data point representing a row vector that follows schema level constraints and intrinsic functional dependencies. Failure to enforce these constraints results in synthetic tables that might appear statistically plausible but actually violate domain logic or structural rules (Shi et al., 2025b; Xu et al., 2025). In applied settings such as healthcare, Barr et al. (2025) demonstrated the effectiveness of zero shot prompting with LLMs to produce clinically plausible perioperative datasets. They validated the fidelity of this data through statistical comparison with real world reference data. More broadly, the field treats synthetic tabular data generation as an optimization problem that balances data utility and fidelity against privacy preservation. Consequently, evaluation protocols are organized around these two dimensions of data quality and privacy protection (Shi et al., 2025b).

4.1 Generation Methods

Prompt-Based and In-Context Tabular Synthesis A prominent category of methods leverages prompt engineering or in context learning to synthesize tabular records without requiring full model fine tuning. These methods typically prioritize schema validity and coverage of long tail distributions. Within this paradigm, CLLM addresses low data regimes by utilizing LLM priors and introducing a curation pipeline based on learning dynamics to filter synthetic rows using confidence and uncertainty metrics (Seedat et al., 2024). A distinct subset of approaches explicitly steers generation toward minority or under represented data segments. For instance, EPIC and LITO employ class conditioned or group conditioned prompting strategies including self authentication mechanisms to bias sampling toward rare classes. This enhances both schema compliance and the representativeness of the long tail (Kim et al., 2025; Yang et al., 2024b). In a complementary approach, TabGen-ICL improves fidelity through the adaptive selection of in context exemplars. It iteratively retrieves real samples that represent the residual between generated and authentic distributions to refine the exemplar pool for a fixed base model (Fang et al., 2025). Beyond direct row synthesis, Nam et al. (2024) incorporate decision tree feedback to guide LLMs in generating new features. This extends prompting methodologies into the domain of feature engineering. Despite these advancements, empirical analyses indicate that unconstrained row wise prompting and arbitrary feature ordering often induce violations of functional dependencies. Furthermore, standard univariate or correlation based metrics frequently fail to detect these constraint failures. This necessitates the development of strictly schema aware and constraint aware prompting and evaluation strategies (Xu et al., 2025).

Fine Tuning and Specialized Tabular Generation A distinct category of methodologies involves fine tuning LLMs specifically for table synthesis. These approaches aim to model structural constraints and distributional dependencies more accurately to enhance both validity and fidelity. For example, GReaT fine tunes an autoregressive language model on serialized tabular rows to enable fully conditional sampling across arbitrary feature subsets. This mechanism improves sample realism and reduces the occurrence of implausible cross feature combinations (Borisov et al., 2023). Extending this architecture, Nguyen et al. (2024) introduce permutation based training combined with feature conditional sampling to preserve feature label correlations and further enhance sample realism. REaLTabFormer targets relational tabular synthesis by generating a parent table autoregressively and conditioning child table generation on the sampled keys (Solatorio & Dupriez, 2023). To address memorization risks and capture inter row structural nuances, HARMONIC incorporates

k-nearest neighbor-based instructional signals that emphasize neighborhood relations across records (Wang et al., 2024d). Other frameworks integrate explicit self correction mechanisms. Table-LLM-Specialist proposes an iterative generator validator self training paradigm where candidate supervision is generated by models and validated via table specific consistency signals such as permutation and execution invariance. This approach facilitates specialist fine tuning without reliance on manual annotations (Xing et al., 2024). Furthermore, adaptive methods such as TableDreamer progressively synthesize instances that expose model weaknesses by targeting observed failure modes to improve data efficiency and downstream utility (Zheng et al., 2025). Collectively, these approaches directly address challenges identified in recent empirical analyses, specifically constraint and functional dependency violations, by incorporating structural awareness during both training and sampling phases (Xu et al., 2025).

Hybrid Architectures and Structured Synthesis Hybrid architectures integrate LLMs with external structure or distribution aware components. Alternatively, they reformulate table synthesis as structured question answering to jointly address gaps in validity, fidelity, and utility. In these designs, LLMs typically function as engines for schema reasoning, constraint handling, and consistency checking, while auxiliary modules ensure structural control and statistical alignment. For instance, AIGT leverages table metadata and long token partitioning to facilitate the generation of wide tables while maintaining quality at scale (Zhang et al., 2024c). Regarding text to table transformation, gTBLS formulates cell filling as a conditional question answering task. This approach explicitly targets syntactic validity to reduce the reliance on extensive corrections after the generation process (Sundar et al., 2024). Motivated by findings regarding constraint violations and distributional mismatches (Xu et al., 2025), these systems explicitly combine reasoning based on LLMs with external components. This combination optimizes end-to-end table realism and downstream applicability (Zhang et al., 2024c; Sundar et al., 2024).

4.2 Quality Metrics for Tabular Data

Validity. We assess validity from both marginal and structural perspectives.

For categorical marginals, we additionally report the column wise Chi-squared test as a sanity check criterion. This metric is used in TabSyn as a quality indicator and typically requires passing at a high threshold such as $p \geq 0.95$ (Zhang et al., 2024a). The calculation is defined as follows:

$$\chi^2 = \sum_{c \in \Omega} \frac{(O_c - E_c)^2}{E_c}, \qquad p = \Pr\left(\chi^2_{\mathrm{df}} \geq \chi^2\right),$$

where O_c and E_c represent the observed synthetic counts and the expected real counts for category c within the set Ω. A larger p value indicates closer marginal agreement, which signifies that the test fails to reject the hypothesis that the distributions are similar.

To measure structural validity, we report the violation rate, which we denote as VR. This rate is defined as the fraction of integrity checks that fail:

$$\mathrm{VR} = \frac{\#\text{violations}}{\#\text{checks}}.$$

The checks include functional dependencies, range limits, uniqueness, geographic consistency, and other schema constraints. A lower violation rate indicates higher validity.

Fidelity. Fidelity measures how closely the synthetic data distribution matches the real data distribution. Following the low order statistics protocol used in TabSyn, we evaluate fidelity at the marginal, pairwise, and global or sample levels (Zhang et al., 2024a).

Marginal fidelity is evaluated with the Kolmogorov Smirnov Test statistic for numerical columns and the Total Variation Distance for categorical columns (Zhang et al., 2024a). The formulas are defined as follows:

$$\mathrm{KST} = \sup_x \left|F_{\mathrm{real}}(x) - F_{\mathrm{syn}}(x)\right|,$$
$$\mathrm{TVD} = \tfrac{1}{2} \sum_{\omega \in \Omega} \left|R(\omega) - S(\omega)\right|,$$

where R and S denote the real and synthetic empirical frequencies of a category ω within the set Ω.

Pairwise fidelity is captured by the Pearson Score for numerical pairs and the Contingency Score, also known as ContSim, for categorical pairs. These metrics follow the methodology described in the TabSyn Appendix E.3 (Zhang et al., 2024a). For a numerical pair consisting of variables x and y, the Pearson correlation is calculated as follows:

$$\rho(x,y) = \frac{\mathrm{Cov}(x,y)}{\sigma_x \sigma_y},$$

The Pearson Score aggregates correlation gaps using a normalization factor of one half because the correlation coefficient ρ ranges from negative one to one (Zhang et al., 2024a):

$$\mathrm{PearsonScore} = \tfrac{1}{2} \cdot \frac{1}{K} \sum_{(i,j)} \left| \rho_{\mathrm{real}}^{(i,j)} - \rho_{\mathrm{syn}}^{(i,j)} \right|.$$

For categorical pairs consisting of variables A and B, the contingency table discrepancy is defined by the following equation:

$$\mathrm{ContingencyScore} = \tfrac{1}{2} \sum_{\alpha \in A} \sum_{\beta \in B} \left| R_{\alpha,\beta} - S_{\alpha,\beta} \right|.$$

For mixed type pairs involving both numerical and categorical data, TabSyn buckets numerical values into categorical bins before computing the corresponding contingency score (Zhang et al., 2024a).

Global realism and detectability are assessed by applying a classifier two sample test in the same spirit as TabSyn (Lopez-Paz & Oquab, 2018). This approach utilizes the SDMetrics detection score based on logistic regression (Zhang et al., 2024a; DataCebo, 2024). In this context, we define the area under the curve as the mean receiver operating characteristic area under the curve of the discriminator when distinguishing between real and synthetic data over cross validation splits. SDMetrics defines the detection score as follows DataCebo (2024):

$$\mathrm{DetScore} = 1 - \big(\max(\mathrm{AUC}, 0.5) \times 2 - 1\big).$$

This formula maps an area under the curve of zero point five, which indicates that the data is indistinguishable, to a detection score of one. Similarly, an area under the curve of one, which indicates that the data is perfectly distinguishable, is mapped to a detection score of zero. Therefore, a higher detection score suggests higher fidelity from the perspective of global detectability (DataCebo, 2024; Zhang et al., 2024a).

Sample level fidelity and α-precision are reported using the support based definition of Alaa et al. (2022), which is also adopted by the TabSyn framework (Zhang et al., 2024a). In this evaluation, let P_r and P_g denote the real and synthetic distributions where S_r is the support of the real distribution and S_g is the support of the synthetic distribution. Following the work of Alaa et al. (2022), the α-support of the real distribution is defined as the minimum volume subset of S_r that contains a probability mass equal to α:

$$S_r^{\alpha} \triangleq \arg\min_{S \subseteq S_r} \ \mathrm{Vol}(S) \quad \text{s.t.} \quad P_r(S) = \alpha,$$

In this context, the volume function represents the Lebesgue volume measure. The corresponding α-precision is defined as the probability that a synthetic sample lies within the real α-support (Alaa et al., 2022):

$$P_{\alpha} \triangleq \Pr_{x \sim P_g} \left(x \in S_r^{\alpha} \right).$$

In practice, we estimate the alpha support from finite samples and compute the empirical α-precision by averaging binary membership indicators over the set of synthetic samples (Alaa et al., 2022):

$$\widehat{\alpha\text{-Prec}} = \frac{1}{|\hat{X}|} \sum_{\hat{x} \in \hat{X}} \mathbf{1}\left[\hat{x} \in \widehat{S}_r^{\alpha} \right],$$

where $\hat{X}$ represents the set of synthetic samples and $\widehat{S}_r^{\alpha}$ is an estimated α-support of the real distribution.

Diversity. The diversity of the generated data is evaluated to measure the coverage of the real distribution. We use β-Recall as proposed by Alaa et al. (2022), which is also used in the TabSyn framework (Zhang et al., 2024a). The β-support for the synthetic distribution is defined in a similar way as follows:

$$S_{\text{syn}}^{\beta} = \arg\min_{S} \; |S| \quad \text{subject to} \quad \Pr(X_{\text{syn}} \in S) = \beta.$$

Based on this definition, β-Recall is expressed by the following equation Alaa et al. (2022):

$$R_{\beta} = \Pr_{X_{\text{real}} \sim P_{\text{real}}} \left(X_{\text{real}} \in S_{\text{syn}}^{\beta} \right).$$

The sample-level estimator for this metric follows the same approach as the one used for α-Precision:

$$\widehat{\beta\text{-Rec}} = \frac{1}{|X|} \sum_{x \in X} \mathbf{1}\left[x \in \widehat{S}_{\text{syn}}^{\beta} \right].$$

A high β-Recall value indicates that the synthetic data provides broad coverage of the original distribution and serves as a complement to the Alpha-Precision metric Alaa et al. (2022); Zhang et al. (2024a).

Utility. We evaluate downstream utility using the Train-on-Synthetic Test-on-Real protocol, which is also referred to as Machine Learning Efficiency in the TabSyn framework Zhang et al. (2024a). Following the evaluation methodology of TabSyn, we report the Area Under the Curve for classification tasks and the Root Mean Square Error for regression tasks (Zhang et al., 2024a):

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR}, \quad \text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \qquad \text{RMSE} = \sqrt{\text{MSE}}.$$

4.3 Trustworthy Metrics for Tabular Data

Privacy. We quantify privacy risks from three complementary perspectives including geometric memorization through nearest neighbor proximity, empirical inference leakage through membership or attribute inference, and formal privacy guarantees such as differential privacy.

To assess geometric memorization and potential record replication, we analyze the distance to the closest record. Following prior work on tabular synthesis based on LLMs, for each synthetic record $s \in \hat{D}$ we compute its distance to the nearest training record (Borisov et al., 2023; Fang et al., 2024):

$$\text{DCR}_{\text{train}\to\text{gen}}(s) = \min_{x \in D_{\text{train}}} d(s, x),$$

where systematically low values indicate a warning signal of potential copying. Because absolute values for the distance to the closest record depend on the choice of distance d and feature scaling, we recommend comparing the distribution of these values against a real baseline. This baseline is defined as $\text{DCR}_{\text{train}\to\text{test}}(x) = \min_{x' \in D_{\text{train}}} d(x, x')$ for hold out real records, which is a common practice in tabular synthesis evaluations (Borisov et al., 2023). Conversely, to assess whether the generator covers the support of the real distribution instead of collapsing to a few modes, we compute the real to synthetic proximity on hold out data:

$$\text{DCR}_{\text{real}\to\text{syn}}(x) = \min_{s \in \hat{D}} d\big(x,\, s\big),$$

where a distribution comparable to real baselines suggests realistic coverage rather than a degenerate generator (Borisov et al., 2023).

Beyond geometric metrics, we evaluate empirical inference leakage through adversarial attacks. We quantify membership inference risk in the standard setting where an attacker is given a candidate record x and some access to the released model or synthesizer. The attacker aims to decide whether x was part of the training data (Shokri et al., 2017). We define $g(x)$ as an attack score where larger values indicate higher confidence that x is a member of the training set. We summarize this risk using the Area Under the Receiver Operating Characteristic Curve and the membership advantage:

$$\mathrm{AUC}_{\mathrm{MIA}} = \Pr[g(x_{\mathrm{train}}) > g(x_{\mathrm{holdout}})], \quad \mathrm{Adv}_{\mathrm{MIA}} = \max_{\tau} \left(\mathrm{TPR}(\tau) - \mathrm{FPR}(\tau)\right).$$

Values for the Area Under the Curve significantly above 0.5 or advantage values significantly above 0 indicate non-trivial leakage. The advantage form matches the standard notion based on the difference between the True Positive Rate and the False Positive Rate used to quantify inference leakage (Yeom et al., 2018).

We similarly evaluate attribute inference for a sensitive attribute A. In this setting, an attacker predicts the value of A from the remaining attributes and any access granted by the released model (Yeom et al., 2018). In addition to reporting standard predictive metrics such as accuracy or the Area Under the Curve, we optionally report a simple gain over majority summary as an implementation level diagnostic:

$$\mathrm{Gain}_{\mathrm{AIA}} = \Pr[\hat{a} = A] - \max_{a} \Pr[A = a],$$

where larger values indicate stronger recoverability of the attribute beyond a trivial majority baseline.

We report the differential privacy parameters ε and δ, which represent the privacy budget, together with downstream utility. This approach allows us to contextualize the trade off between privacy and utility under the standard differential privacy framework (Dwork & Roth, 2014).

Fairness. We evaluate fairness in a train on synthetic and test on real setting. This is necessary because downstream models trained on synthetic data may inherit or even amplify disparities when they are deployed on real populations. This setting is a commonly used evaluation protocol for tabular generation. Fairness considerations are concerns that exist across the entire pipeline starting from data through the model and ending at deployment (Barocas et al., 2023).

For outcome fairness including demographic parity and disparate impact, we consider a protected attribute $A \in \{a, b\}$ and a binary decision $\hat{Y}$. We report the Statistical Parity Difference and the Disparate Impact Ratio:

$$\Delta_{\mathrm{SPD}} = \left|\Pr(\hat{Y} = 1 \mid A = a) - \Pr(\hat{Y} = 1 \mid A = b)\right|, \quad \mathrm{DIR} = \frac{\Pr(\hat{Y} = 1 \mid A = a)}{\Pr(\hat{Y} = 1 \mid A = b)}.$$

Ideally, the Statistical Parity Difference should approach 0 and the Disparate Impact Ratio should approach 1. The Disparate Impact Ratio is widely used in the literature concerning disparate impact as a rate ratio criterion such as the eighty percent rule (Feldman et al., 2015; Barocas et al., 2023).

To account for the trade offs regarding accuracy, we evaluate error rate fairness through equalized odds and equal opportunity disparities. These metrics measure the gaps between groups in the True Positive Rate and the False Positive Rate by conditioning on the true label (Hardt et al., 2016):

$$\Delta_{\mathrm{EO}} = \frac{1}{2}\left(|\Delta\mathrm{TPR}| + |\Delta\mathrm{FPR}|\right), \quad \Delta_{\mathrm{EOp}} = |\mathrm{TPR}_a - \mathrm{TPR}_b|.$$

Because parity degradations in the train on synthetic and test on real setting can come from representational mismatches between synthetic and real data, we also report two simple diagnostics. The subgroup coverage gap measures shifts in group proportions, and the label conditional shift captures label distortion within specific groups:

$$\mathrm{CovGap} = \frac{1}{2}\sum_{g \in \mathcal{A}} |p_{\mathrm{syn}}(A{=}g) - p_{\mathrm{real}}(A{=}g)|, \quad \mathrm{CondShift} = \frac{1}{|\mathcal{A}|}\sum_{g \in \mathcal{A}} \mathrm{TV}(p_{\mathrm{syn}}(Y \mid A{=}g),\, p_{\mathrm{real}}(Y \mid A{=}g)).$$

Large values in these representational diagnostics often help explain why parity metrics degrade after the evaluation. This observation is consistent with broader discussions of fairness as a property of the entire pipeline starting from data through the model and ending at deployment (Barocas et al., 2023).

Generation method	Validity	Fidelity	Diversity	Utility	Privacy	Fairness
GReaT (Borisov et al., 2023)	✓	✓	×	✓	✓	×
REaLTabFormer (Solatorio & Dupriez, 2023)	△	✓	×	✓	✓	×
EPIC (Kim et al., 2025)	△	✓	△	✓	×	×
HARMONIC (Wang et al., 2024d)	×	△	×	✓	△	×
TabSyn (Zhang et al., 2024a)	✓	✓	✓	✓	✓	×

Table 4: Whether representative Tabular Data generation methods explicitly evaluate each dimension in their experimental sections. ✓: explicitly evaluated; △: partially/indirectly covered; ×: not reported or not applicable.

4.4 Evaluation Practice Gap

We analyze representative methods from Section 4.1 and categorize their reported evaluation protocols in Table 4.

Diversity. Our analysis shows that diversity is not evaluated enough in tabular synthesis. This happens because specific measures focused on coverage such as β-Recall are rarely reported in experimental settings. This gap is very important for generators driven by LLMs. In these models, standard training goals based on likelihood and decoding procedures often show behavior that focuses on the most common patterns. Thus, synthetic distributions tend to over represent frequent patterns while failing to capture long tail values and rare combinations of features. Although synthetic datasets may seem realistic within dominant patterns, they often lack enough support for areas with low frequency. We therefore recommend reporting coverage metrics such as β-Recall and diagnostics at the subgroup level alongside standard measures of fidelity and utility.

Fairness. Our review suggests that fairness evaluation is mostly missing from current studies on tabular generation. This is a significant gap because fairness is very important for the Train on Synthetic and Test on Real (TSTR) protocol. When synthetic data are used to train downstream models for deployment on real populations, representational mismatches can lead directly to disparities in outcome fairness and error rate fairness. These mismatches include shifts in subgroup proportions or label conditional distributions. For example, outcome fairness can be measured by statistical parity, and error rate fairness can be measured by equalized odds. Moreover, improved diversity does not always mean improved fairness. Expanding the coverage of the data may accidentally increase disparities if minority subgroups are created with lower fidelity or if the conditional distributions are distorted. Accordingly, we recommend adding group fairness metrics such as the statistical parity difference and equalized odds to the results of the train on synthetic and test on real protocol. We also suggest using representational diagnostics to clearly describe the possible trade offs between diversity, fairness, and downstream performance.

4.5 Usage

Data Sharing. Synthetic tabular data based on LLMs is increasingly used to facilitate safe data sharing and privacy aware anonymization. By producing records that maintain statistical fidelity to the source distribution while differing from original data points, organizations can release or exchange datasets without compromising sensitive or personally identifiable information. This capability is especially important in regulated domains such as healthcare and finance. In these fields, synthetic data serves as a compliant mechanism for collaboration between different institutions (Miletic & Sariyar, 2024; Barr et al., 2025; Long et al., 2025).

In this context, Miletic & Sariyar (2024) benchmark LLMs based on the Transformer architecture against baseline models such as CTGAN (Xu et al., 2019). They show that larger language models yield superior downstream classification utility and maintain competitive performance even at smaller scales. Furthermore, Barr et al. (2025) show that GPT-4o is capable of generating clinical tabular data in a zero shot manner and achieving high fidelity in terms of the statistical properties within each column. This performance is especially high when the model is guided by descriptive statistics. However, the authors did not directly evaluate downstream utility or the risks of duplication and memorization. They highlighted these areas as important directions for further study.

Beyond the imitation of individual rows, frameworks such as LLM-TabFlow use the reasoning capabilities of LLMs to capture logical relationships between columns and synthesize data within a latent space. This approach helps to optimize the trade off between fidelity, utility, and privacy (Long et al., 2025). Together, these findings indicate that synthesis driven by LLMs is becoming a practical way to share sensitive tables under strict regulatory constraints.

Data Augmentation. In situations characterized by data scarcity or a lack of rare feature combinations, generation driven by LLMs provides a way to increase training distributions. This approach helps to improve class balance and the ability of models to generalize to new data (Seedat et al., 2024; Tran & Xiong, 2024; Kim et al., 2025; Yang et al., 2024b).

Recent research is moving beyond simple oversampling methods toward more controllable ways of creating tabular data. In some cases, these methods include formal privacy guarantees. For instance, DP-LLMTGen ensures differential privacy by using a two stage fine tuning pipeline. This process first involves learning the data format and then performing fine tuning with differential privacy using a loss function designed for tabular data, and finally samples the private model to create synthetic tables (Tran & Xiong, 2024).

In a related direction, P-TA uses an optimization scheme based on Proximal Policy Optimization to include feedback from a discriminator. This improves the alignment between the generated data and the real tabular distributions (Yang et al., 2025). As a result, approaches in this field are transitioning from basic oversampling techniques to generation frameworks that are both controllable and aware of privacy. These developments make generators based on LLMs flexible tools for improving the performance of downstream models in difficult data environments.

5 Semi-structured

Semi-structured data can be thought as an intermediate modality between rigid relational schemas and unstructured text. This modality is characterized by flexible schemas and hierarchical organization. In this survey, we unify **Graph**, **JSON**, and **Log** data under this category.

Graph data consists of nodes and edges that define topological relationships. Recent literature demonstrates that LLMs can synthesize such structures by serializing them into text formats such as edge lists (Yao et al., 2024).

JSON data represents nested structures of objects consisting of name-value pairs and arrays (Bray, 2017). However, reliable JSON generation requires strict schema adherence and syntax verification to ensure the output is executable by downstream tools (Agarwal et al., 2025a).

Log data consists of sequential event messages where each entry typically combines a timestamp with a log template and variable parameters. This format allows for automatic parsing into structured templates (He et al., 2017).

5.1 Generation Methods

5.1.1 Graph Data

Recent advancements in LLM-driven graph synthesis can be classified based on the necessity of parameter updates into training-free generation versus learning-based generation.

Training-free Graph Generation. A growing body of work explores the capability of LLMs to produce syntactically valid and structurally plausible graphs without gradient-based tuning. Yao et al. (2024) introduce LLM4GraphGen, demonstrating that models such as GPT-4 can generate graph structures directly from natural language prompts. This work covers both rule-based and distribution-based tasks. While direct prompting offers a domain-agnostic and deployment-friendly approach, existing evaluations indicate limitations in structural fidelity. This is particularly evident when targeting complex distributional parameters such as specific motif counts (Yao et al., 2024). To move beyond naive prompting, Generate-on-Graph tackles Incomplete Knowledge Graph Question Answering through a training-free exploration procedure. This method retrieves Knowledge Graph evidence and generates additional factual triples dynamically when crucial triples are missing (Xu et al., 2024b). In parallel, ontology-grounded Knowledge Graph construction leverages a Wikidata-schema-aligned ontology to ground relation extraction and improve interoperability with existing ontologies (Feng et al., 2024).

Learning-Based and Multi-Agent Generation. Alternative approaches enhance the quality of graph construction through supervised fine-tuning or multi-agent collaboration frameworks (Huang et al., 2025a; Le et al., 2024). GraphJudge (Huang et al., 2025a) employs supervised fine-tuning to train LLMs as discriminators for graph quality. This approach significantly reduces noise during Knowledge Graph enrichment by filtering generated triples. Beyond the use of individual models, multi-agent systems such as GAG and GraphMaster leverage collaborative refinement to improve global consistency. GAG formulates graph synthesis as a scalable and simulation-based multi-agent process. This method enables the generation of large text-attributed graphs that adhere to macroscopic network properties (Ji et al., 2025). Similarly, GraphMaster introduces an evaluation-driven iterative loop among agents. This system is specifically designed to enhance the structural integrity and semantic coherence of the synthesized graphs (Du et al., 2025).

5.1.2 JSON Data

LLMs have been adapted to generate schema-compliant JSON through two representative methodological classes. These include inference time constraint control, often referred to as guided decoding, and learning based alignment such as reinforcement learning to improve schema adherence (vLLM Project, 2024; Lu et al., 2025; Agarwal et al., 2025a).

Constrained Decoding for JSON Generation. In this regime, model parameters remain unchanged and structural compliance is enforced at inference time using decoding time constraints. These constraints include token filtering guided by schemas or regular expressions (vLLM Project, 2024; Gat, 2025; dottxt-ai, 2025). As a complementary step, LLMs can also be used to infer and enrich JSON Schemas from existing corpora. This can be achieved by generating natural language descriptions for schema elements and identifying potentially noisy properties (Mior, 2024). Once a target schema is available, practical frameworks such as vLLM structured outputs and toolkits like Outlines and LM Format Enforcer implement constrained decoding by masking invalid tokens. These systems ensure that the final output conforms to the target schema and prevent malformed JSON that may arise under unconstrained prompting. Furthermore, JSONSchemaBench provides large scale evidence and analysis of compliance, coverage, and efficiency for these constrained decoding systems across 10,000 real world JSON schemas (vLLM Project, 2024; dottxt-ai, 2025; Gat, 2025; Geng et al., 2025). However, while these approaches excel at enforcing surface form validity, satisfying stricter requirements may still require additional validation, post processing, or learning based alignment. Such requirements include fine grained field typing and cross field semantic consistency (Lu et al., 2025).

Learning-Based Generation. When stricter schema adherence is required, reinforcement learning based methods optimize the model policy with reward signals tied to schema correctness. These rewards are typically provided via schema validators or verifier style rewards (Lu et al., 2025; Agarwal et al., 2025a). For instance, Lu et al. (2025) introduce SchemaBench which contains approximately 40,000 JSON schemas. They improve structured generation by incorporating reinforcement learning with a fine grained schema validator. This approach outperforms standard supervised fine-tuning baselines (Lu et al., 2025). Meanwhile, Agarwal et al. (2025a) apply Group Relative Policy Optimization to train smaller models with custom rewards for strict schema adherence. This work demonstrates effective improvements in enforcing schema consistency.

5.1.3 Log Data

Research in log generation highlights a discrepancy between the semantic prediction of log components and the syntactic realization of the final log message. These components include log levels and variables. Li et al. (2024d) observe that while LLMs can accurately determine necessary logging attributes, they often fail to produce full log statements that mimic human-written code. On their benchmark, LogBench, the best-performing models achieved a BLEU score of only 0.249. This indicates limited surface-form similarity (Li et al., 2024d). Furthermore, evaluations on semantically equivalent but transformed code contexts, referred to as LogBench-T, show consistent performance degradation. This is especially true for variable prediction and log-text generation, suggesting that general-purpose LLMs remain brittle under semantics-preserving code transformations (Li et al., 2024d).

To bridge this performance gap, recent studies advocate for model specialization via post-training on domain-specific corpora. Zhang et al. (2025a) propose AUCAD, which is a framework that automatically constructs an alignment dataset called AucadLog derived from log-related software issues. By leveraging this dataset to post-train open-source LLMs, the authors demonstrate that the resulting specialized models significantly outperform existing model-based solutions in log statement generation. These results are confirmed by both human evaluation and quantitative metrics (Zhang et al., 2025a).

5.2 Quality Metrics for Semi-structured Data

5.2.1 Graph Data

Validity. Validity represents the proportion of generated graphs that comply with specific task rules or logical constraints. This metric reflects whether the output of the model is effective in a topological or physical sense. We measure the fraction of generated graphs that satisfy task specific rule-based constraints:

$$\mathrm{Valid}_{\mathrm{rule}} = \frac{1}{|\mathcal{G}_{\mathrm{gen}}|} \sum_{G \in \mathcal{G}_{\mathrm{gen}}} \mathbb{I}\{\mathrm{passes\_rules}(G)\}.$$

where $\mathcal{G}_{\mathrm{gen}}$ denotes the set of generated graphs where $\mathcal{G}_{\mathrm{gen}} \subseteq \mathcal{G}$. The function passes_rules is a task defined rule checker that returns 1 if G satisfies the rules and 0 otherwise. These rules may include constraints specified by the generation task (Yao et al., 2024).

Fidelity. Fidelity measures the structural similarity between generated graphs and reference data. We assess how closely generated graphs match reference structures using kernel based Maximum Mean Discrepancy computed on graph descriptor features:

$$\mathrm{MMD}^2(\mathcal{X}, \mathcal{Y}) = \frac{1}{m^2} \sum_{i,i'} k(x_i, x_{i'}) + \frac{1}{n^2} \sum_{j,j'} k(y_j, y_{j'}) - \frac{2}{mn} \sum_{i,j} k(x_i, y_j),$$

where $\mathcal{X} = \{x_i\}_{i=1}^{m}$ and $\mathcal{Y} = \{y_j\}_{j=1}^{n}$ are descriptor sets extracted from generated and real graphs respectively, and k is a chosen kernel (You et al., 2018; Liao et al., 2020). Typical descriptors include degree distributions, clustering coefficient distributions, orbit or motif counts, and spectral statistics such as Laplacian eigenvalue histograms (You et al., 2018; Liao et al., 2020).

For molecular graphs, we include the **Frechet ChemNet Distance** (Preuer et al., 2018):

$$\mathrm{FCD} = \|\mu_1 - \mu_2\|_2^2 + \mathrm{Tr}\Big(\Sigma_1 + \Sigma_2 - 2\big(\Sigma_1 \Sigma_2\big)^{1/2}\Big),$$

This metric measures the distance between embedding distributions of generated and reference molecules. μ and Σ represent the mean and covariance of these distributions where embeddings are obtained from a pretrained ChemNet model (Preuer et al., 2018).

Diversity. Diversity evaluates the variety and originality among the generated graphs. This includes the novelty of the graphs relative to the prompt and the non-repetition within the generated set. To evaluate variety, we compute Novelty. A generated graph is considered novel if it is different from every example graph provided in the prompt for the same task:

$$\mathrm{Novelty} = \frac{1}{|\mathcal{G}_{\mathrm{gen}}|} \sum_{G \in \mathcal{G}_{\mathrm{gen}}} \mathbb{I}\{G \not\equiv \mathcal{G}_{\mathrm{ex}}\},$$

where $\mathcal{G}_{\text{ex}} \subseteq \mathcal{G}$ denotes the set of example graphs in the prompt. The condition $G \not\equiv \mathcal{G}_{\text{ex}}$ means G is not identical to any example graph under the same equality criterion used by the evaluation pipeline (Yao et al., 2024).

We also compute Uniqueness as the fraction of valid generated graphs that are not duplicates:

$$\text{Uniq} = \frac{|\text{unique}(\mathcal{G}_{\text{valid}})|}{|\mathcal{G}_{\text{valid}}|}, \qquad \mathcal{G}_{\text{valid}} = \{G \in \mathcal{G}_{\text{gen}} : \text{passes\_rules}(G)\},$$

where the function unique removes duplicates under the same equality criterion used in evaluation (Yao et al., 2024).

Utility. Utility assesses the practical value of the generated data for downstream applications. This is typically measured by how much the synthetic data contributes to the performance of a model on a real world task. When a Graph Neural Network is trained on enhanced graphs and evaluated on a fixed split, we report Accuracy and F1 Score. Higher values for Accuracy and F1 indicate greater utility of the synthetic graphs for downstream tasks (Du et al., 2025).

5.2.2 JSON Data

For semi-structured outputs such as JSON responses, quality metrics commonly focus on three aspects. These dimensions include validity, fidelity, and utility.

Validity. Validity represents the requirement that outputs are machine-parseable and conform to the expected schema. This ensures that the generated data is effective for automated processing. Let $\mathcal{J} = \{J_1, \ldots, J_N\}$ be the set of generated JSON objects and $\mathcal{S}$ be the target schema.

We define the Correctness Indicator for a generated object J under schema $\mathcal{S}$ as follows:

$$V(J, \mathcal{S}) = \mathbb{I}[\text{parsable}(J) \wedge \text{schema}(J, \mathcal{S})].$$

In this equation, $\mathbb{I}[\cdot]$ is the indicator function that returns 1 if J is syntactically valid and schema-compliant, and 0 otherwise.

We then report the average Correctness Rate:

$$\text{CorrectnessRate} = \frac{1}{|\mathcal{J}|} \sum_{J \in \mathcal{J}} V(J, \mathcal{S}).$$

In addition, we report a purely syntactic metric called the Valid JSON rate or parsability rate. This metric measures whether the output is parseable as JSON while ignoring specific schema constraints:

$$\text{ValidJSONRate} = \frac{\big|\{J \in \mathcal{J} \mid \text{IsParsable}(J) = \text{true}\}\big|}{|\mathcal{J}|}.$$

Therefore, we report two distinct measures. The first is a purely syntactic parsability metric referred to as the ValidJSONRate. The second is a schema level validity metric referred to as the CorrectnessRate. Parsability metrics are explicitly reported in strict structured output evaluations Agarwal et al. (2025a). At the same time, schema validation protocols are commonly used in settings involving schema constrained generation Lu et al. (2025).

Fidelity. Fidelity measures how closely the generated content matches the semantic information and distribution of the target data. This dimension reflects the quality of the content within the structured JSON framework. Fidelity metrics evaluate whether the model produces the correct fields, values, and semantics beyond simple syntactic validity.

For structured JSON where ground truth instances denoted as J_i^{gt} are aligned to generated instances J_i, the **Mean Match Percentage** compares generated fields against the ground truth instantiations:

$$MMP = \frac{1}{N}\sum_{i=1}^{N} \frac{|\text{fields}(J_i) \cap \text{fields}(J_i^{\text{gt}})|}{|\text{fields}(J_i^{\text{gt}})|}.$$

This metric can be extended to checks at the level of values or constraints when ground truth values and constraints are available (Agarwal et al., 2025a).

Schema focused evaluations also examine whether the schema itself is meaningful. For example, researchers may assess the quality of generated definitions and names through similarity based on embeddings. Additionally, property selection is evaluated as a classification problem between useful and noisy properties. This is typically reported using accuracy Mior (2024).

Utility. Utility evaluates the practical effectiveness of the generated JSON data in solving a specific task or its usefulness as an input for downstream processes. To evaluate the utility of generated JavaScript Object Notation responses for later use, we require each response to follow a predefined schema so that the data can be read reliably. We measure the success of the task by calculating the exact match accuracy between the extracted answer and the ground truth. Let the set of generated responses be denoted as $\{J_i\}_{i=1}^{N}$ and the reference answers be denoted as $A_{\text{gt},i}$. We define Task Accuracy as the average rate at which the extracted answer matches the reference:

$$\text{TaskAcc} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(\text{ExtractAnswer}(J_i) = A_{\text{gt},i}\right).$$

This measurement reflects how well the synthetic data supports the successful completion of the target application. This methodology aligns with evaluations based on task accuracy for schema constrained structured outputs (Geng et al., 2025).

5.2.3 Log Data

Logs are typically semi-structured in nature. Each message consists of a fixed template combined with runtime variables or parameters, which may also be accompanied by metadata such as severity levels and components. Accordingly, evaluation should separately address two distinct aspects. The first is structural validity under a log parser, which involves template and variable extraction as well as grouping. The second is the content fidelity of the generated text, variables, and metadata.

Validity. For log data, validity depends not only on surface well-formedness but also on whether the logs can be correctly structured into templates and variables by a parser. A common message-level metric used for this purpose is Parsing Accuracy, which measures the fraction of log messages that are parsed exactly into the correct template and variables:

$$\text{PA} = \frac{|\mathcal{L}_{\text{correct}}|}{|\mathcal{L}|},$$

where $\mathcal{L}$ represents the set of all log messages $\{\ell_i\}$, and $\mathcal{L}_{\text{correct}}$ refers to the subset whose predicted template, consisting of static tokens, and variable spans, consisting of dynamic tokens, match the ground truth exactly (Khan et al., 2022; Jiang et al., 2024a; Ma et al., 2024b).

Beyond per-message correctness, Grouping Accuracy evaluates whether messages are assigned to the correct template group:

$$\text{GA} = \frac{|\mathcal{L}_{\text{grouped}}|}{|\mathcal{L}|},$$

where $\mathcal{L}_{\text{grouped}}$ denotes messages whose predicted grouping is consistent with the ground-truth template partition. This means that messages belonging to the same true template are grouped together and those from different templates are separated (Khan et al., 2022; Ma et al., 2024b). Grouping Accuracy is particularly important when downstream pipelines rely on template clusters rather than individual parses.

Since Parsing Accuracy and Grouping Accuracy are based on message counts, they can be overly influenced by highly frequent templates, such as heartbeat messages. This motivates the use of template-level evaluation (Khan et al., 2022; Jiang et al., 2024a). Let $\mathcal{T}_{gt}$ and $\mathcal{T}_p$ be the sets of ground-truth and predicted templates, and let $\mathcal{T}_{\text{correct}}$ be the set of correctly identified templates. This is often reported with oracle template correction following common evaluation guidelines (Khan et al., 2022). Define template-level precision and recall as

$$P_{\text{TA}} = \frac{|\mathcal{T}_{\text{correct}}|}{|\mathcal{T}_p|}, \qquad R_{\text{TA}} = \frac{|\mathcal{T}_{\text{correct}}|}{|\mathcal{T}_{gt}|}.$$

The F1-score of Template Accuracy is defined as

$$\text{FTA} = 2 \cdot \frac{P_{\text{TA}} \cdot R_{\text{TA}}}{P_{\text{TA}} + R_{\text{TA}}},$$

which complements Parsing Accuracy and Grouping Accuracy by emphasizing template coverage and uniqueness (Khan et al., 2022; Jiang et al., 2024a). Recent large-scale benchmarks further recommend additional template-level grouping measures, such as the F1-score of Group Accuracy, to mitigate the sensitivity of message-level grouping metrics when template frequencies are imbalanced (Jiang et al., 2024b). Overall, these metrics capture whether generated logs support faithful structural parsing beyond mere syntactic correctness.

Fidelity. For logs, fidelity emphasizes how well generated templates, variables, and metadata preserve linguistic realism and operational plausibility. When models generate or reconstruct variables, Variable Precision, Variable Recall, and the Variable F1-score assess whether the bound runtime variables are correct:

$$\text{VP} = \frac{|V_p \cap V_t|}{|V_p|}, \quad \text{VR} = \frac{|V_p \cap V_t|}{|V_t|}, \text{VF1} = 2 \cdot \frac{\text{VP} \cdot \text{VR}}{\text{VP} + \text{VR}},$$

where V_p and V_t are the predicted and true variable sets (Li et al., 2024d).

For template and text realism, overlap-based metrics such as BLEU and ROUGE can be used between generated and reference log templates or texts. In addition, embedding-based semantic similarity is often reported to reduce sensitivity to paraphrases and mixed natural language or code phrasing (Li et al., 2024d).

Metadata fidelity is also crucial. For example, Log Level Accuracy measures the fraction of logs with exactly correct severity levels:

$$\text{L-ACC} = \frac{N_{\text{correct\_level}}}{N},$$

Furthermore, the Average Ordinal Distance captures how far predicted levels deviate on an ordinal severity scale:

$$\text{AOD} = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{\text{Dis}(l_{\text{pred}}^{(i)}, l_{\text{gt}}^{(i)})}{\text{MaxDis}} \right),$$

where Dis($\cdot$) is the ordinal distance between levels, such as the ranking from error to trace, and MaxDis is the maximum possible distance on the chosen scale (Li et al., 2024d).

Utility. For logs, downstream utility reflects their contribution to system understanding and diagnosis. A common evaluation approach is to determine whether synthetic or augmented logs improve or preserve performance on downstream log analysis tasks, such as parsing or anomaly detection, under standard pipelines and fixed test sets (Huo et al., 2023).

A direct way to quantify utility is the change in downstream model performance when synthetic logs are used for training or augmentation:

$$\Delta_{\mathcal{M}} = \mathcal{M}\big(f(\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{syn}})\big) - \mathcal{M}\big(f(\mathcal{D}_{\text{real}})\big),$$

where $\mathcal{M}$ is a task metric, such as the F1 score or accuracy, and f is the downstream model or procedure evaluated on a fixed test set.

In human-centered evaluations, developers may additionally rate the usefulness and readability of generated logs and explanations on Likert scales to capture practical diagnostic value (Liu et al., 2024d).

5.3 Trustworthy Metrics for Semi-structured Data

5.3.1 Graph Data

Privacy. Graph-structured data raises distinct privacy risks because nodes and edges often correspond to individual users and their relationships. These risks are formalized using differential privacy notions specialized to graph adjacency (Kasiviswanathan et al., 2013).

The formal definition of privacy in graph generation is based on the framework of differential privacy, which is characterized by two key parameters, ε and δ. A randomized mechanism M satisfies the (ε, δ)- differential privacy inequality if for any two neighboring datasets D and D' that differ by a single record, and for any possible output set S, the probability that the mechanism produces an output in S for dataset D is bounded by the following condition:

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S] + \delta.$$

In this mathematical framework, ε represents the privacy budget or privacy loss, which quantifies the maximum allowable change in the output probability distribution when one individual's information is modified. A smaller ε value indicates a stronger privacy guarantee because it ensures that the outputs are more similar regardless of whether a specific record is present. The parameter δ represents the probability that the privacy constraint might be violated, and it is typically required to be a very small value to ensure that the guarantee remains robust (Dwork & Roth, 2014).

When this definition is applied to graph structured data, the concept of neighboring datasets is specialized to either node or edge adjacency (Kasiviswanathan et al., 2013). Node level differential privacy defines neighbors as two graphs where one is obtained from the other by removing a single node along with all its incident edges. In contrast, edge level differential privacy considers graphs to be neighbors if they differ by exactly one edge. Under these adjacency notions, the parameters ε and δ serve as quantitative metrics for the risk that a specific user or relationship could be inferred from the generated graph (Kasiviswanathan et al., 2013; Dwork & Roth, 2014).

Similarly, edge-level central differential privacy treats neighboring graphs as those differing by exactly one edge, a concept known as edge adjacency (Kasiviswanathan et al., 2013). Under this adjacency notion, the same (ε, δ)-differential privacy inequality applies (Dwork & Roth, 2014):

$$\Pr[M(G) \in S] \leq e^{\varepsilon} \Pr[M(G') \in S] + \delta, \ \forall S \subseteq \mathcal{O}.$$

In this case, the presence or absence of any single relationship or edge is protected.

For more granular protection without a trusted curator, we can work in the local model of differential privacy, where each participant randomizes their contribution before sharing. A standard formalization of local differential privacy is given via a likelihood-ratio bound on the per-user privatization channel (Duchi et al., 2014). Instantiating this standard definition for per-edge reports, such as an edge indicator or local adjacency information, we use the following notion of edge-level local differential privacy. A mechanism M is ε-edge-level local differential privacy if for any two possible edge values e and e' and any output set $S \subseteq \mathcal{O}$, the following holds:

$$\Pr[M(e) \in S] \leq e^{\varepsilon} \Pr[M(e') \in S] \qquad \forall\, e, e', S \subseteq \mathcal{O}.$$

In many local differential privacy settings, one focuses on the pure case described above (Duchi et al., 2014). One can also consider an (ε, δ) extension by adding a δ term as in approximate differential privacy (Dwork & Roth, 2014), but we maintain the common pure form for clarity.

A common implementation in the central model uses global sensitivity and Laplace noise (Dwork & Roth, 2014):

$$\Delta f = \max_{G \sim G'} \|f(G) - f(G')\|_1,$$

$$\eta \sim \mathrm{Lap}\Big(0, \tfrac{\Delta f}{\varepsilon}\Big),$$

where f is a numeric query, such as degree-histogram bins. The Laplace mechanism yields pure ε-differential privacy, which means δ is equal to zero, under the chosen adjacency notion (Dwork & Roth, 2014). In graph settings, Δf can be very large for degree-related queries under node adjacency, since adding or removing a node can affect many incident edges and consequently many degree counts. Practical approaches therefore impose bounded-degree assumptions or apply degree clipping and projection to a bounded-degree graph family. These techniques keep Δf and the required noise level manageable (Kasiviswanathan et al., 2013).

In our evaluation, the privacy parameters ε and δ for central differential privacy, and ε for pure local differential privacy, serve as quantitative privacy metrics. A smaller ε and a smaller δ indicate stronger privacy protection, which typically comes at the cost of utility (Dwork & Roth, 2014).

Finally, it is common to require δ to be negligible in an appropriate problem or security parameter. It is important to avoid regimes where δ is non-trivially large, such as on the order of one over n for population size n (Dwork & Roth, 2014). In graph settings where nodes correspond to protected individuals, a practical heuristic is to select δ to be far smaller than one over the number of vertices in the graph when using approximate differential privacy. The exact choice remains dependent on the specific application and threat model.

Robustness. Beyond privacy, graph generation must be robust to structural hallucinations, which refers to the production of plausible but incorrect graphs (Richardeau et al., 2025). We use complementary checks to capture this risk.

The Syntactic Correctness Rate is defined as the fraction of generated outputs that can be parsed into valid graph structures under the required grammar or format:

$$\sigma_{SCR} = \frac{1}{|\mathcal{G}_{\text{gen}}|} \sum_{G \in \mathcal{G}_{\text{gen}}} \mathbb{1}\{\textsc{ParseOK}(G) = 1\}.$$

A low value of σ_{SCR} indicates format-breaking outputs and sensitivity to changes in prompting or decoding. This is particularly relevant when the generator is prompted to output a concrete graph representation, such as edge lists, which must be parsed into a graph object for downstream evaluation. Such generate-then-parse pipelines are common in research involving graph generation with LLMs (Richardeau et al., 2025).

The Graph Atlas Distance is employed to directly measure topological deviation for well-specified targets, such as canonical atlas graphs. Following the methodology proposed in (Richardeau et al., 2025), this metric is defined as follows:

$$\text{GAD} = \frac{1}{k} \sum_{i=1}^{k} d_{\text{GED}}(G_i, a_i),$$

where $G_1, \dots, G_k$ represent the outputs generated from a fixed set of prompts whose ground-truth targets are the corresponding canonical atlas graphs $a_1, \dots, a_k$. In this formulation, d_{GED} denotes the graph edit distance as described in (Richardeau et al., 2025). A larger value for the Graph Atlas Distance implies more severe structural hallucination in the generated results.

To reduce sensitivity to rare and extreme edit distances which might otherwise dominate the mean, we additionally report a capped version of the Graph Atlas Distance known as robust aggregation. This metric is calculated as follows:

$$\text{GAD}^{\text{cap}} = \frac{1}{k} \sum_{i=1}^{k} \min\big\{ d_{\text{GED}}(G_i, a_i),\, C \big\},$$

where C caps extreme values. In this context, the uncapped average corresponds to the standard definition of the Graph Atlas Distance proposed in (Richardeau et al., 2025), while the cap serves as a robustification measure during the evaluation process.

The Degree-distribution Deviation captures distributional drift by measuring a lightweight deviation between normalized degree histograms. This metric is defined by the following expression:

$$D_{L_2}(G) = \big\| h_{\text{gen}}(G) - h_{\text{ref}} \big\|_2,$$

where $h_{\text{gen}}(G)$ represents the normalized degree histogram of the generated graph G and h_{ref} is the target histogram. This target may be derived from the ground-truth graph or a reference dataset. A higher value of this deviation indicates that the generated graphs may appear reasonable but diverge statistically from the reference model. Comparing degree-based statistics through distributional distances, such as Maximum Mean Discrepancy, Kullback Leibler divergence, or Kolmogorov Smirnov tests, is standard practice in the evaluation of graph generation (Liao et al., 2020; Liu et al., 2024b). In this study, we utilize an L_2 histogram deviation for simplicity. We emphasize that degree-based deviations provide useful although not sufficient auxiliary signals. Consequently, they should be interpreted alongside topology-sensitive measures such as the Graph Atlas Distance or the graph edit distance (Richardeau et al., 2025).

5.3.2 JSON Data

Privacy. Privacy is a critical concern for generated JSON data, such as electronic health record resources in JSON format, that contains personally identifiable information or other sensitive data. In pipelines based on sanitization, privacy can be evaluated through established de-identification methods. Furthermore, the quality of privacy-enhancing post-processing is essential. This includes techniques like entity relexicalization, which ensures consistent surrogate substitutions to maintain coherence for downstream applications (Singh et al., 2025).

These document-level and corpus-level scores measure the accuracy of sensitive entity detection and removal. Since even a single leak of sensitive data can lead to a major privacy breach, researchers often adopt stricter definitions of recall (Singh et al., 2025). All-or-Nothing Recall represents this strictness by treating the completeness of detection within a document as a binary outcome. The value is 1 only if no instance of the target entity type is missed and it is 0 otherwise (Scaiano et al., 2016; Singh et al., 2025).

$$\mathrm{Recall}_{\mathrm{AON}}(t, d) = \mathbb{I}\Big(E_{t,d} \subseteq \hat{E}_{t,d}\Big) = \mathbb{I}\Big(E_{t,d} \setminus \hat{E}_{t,d} = \emptyset\Big).$$

In this equation, for an entity type t and document d, $E_{t,d}$ represents the set of ground-truth sensitive entity instances of type t. Additionally, $\hat{E}_{t,d}$ denotes the set of instances that are correctly detected and redacted, and $\mathbb{I}(\cdot)$ is the indicator function. This document-level definition reflects the fact that a single missed instance can invalidate privacy protection for that specific entity type.

Clinical Model Consistency measures whether relexicalized data avoids introducing systematic biases when it is used in clinical decision-making models. These biases may include factors such as race, ethnicity, region, or age group. This metric also assesses whether the data preserves real-world model behavior (Singh et al., 2025). Specifically, if a clinical model achieves a performance metric value X when trained and evaluated on real data, poor relexicalization may result in a metric of $X + \delta_X$:

$$|\delta_X| = \big|X_{\mathrm{relex}} - X_{\mathrm{real}}\big|.$$

A smaller value of $|\delta_X|$ indicates higher-quality relexicalization. This ensures that the downstream clinical utility and behavior remain close to those of models that are trained and evaluated on real data.

5.3.3 Log Data

Privacy. We recommend evaluating privacy in logs by explicitly capturing the residual disclosure risk that remains in the released log text and the utility loss induced by anonymization. This approach reflects the practical trade-off between privacy and utility that is widely observed in log anonymization and task-aware anonymization benchmarking (Aghili et al., 2025; Loiseau et al., 2025). Inspired by evaluation practices in text anonymization, we organize privacy evaluation for logs along three complementary axes. These axes include exposure-oriented privacy risk, downstream utility degradation, and human-centered effectiveness (Pilán et al., 2022; Ren et al., 2025; Loiseau et al., 2025). Logs are naturally related to text anonymization because they are categorized as semi-structured text.

Sensitive Attribute Exposure measures an observable indicator of privacy risk by counting the number of sensitive attribute instances that remain in a log message after anonymization. Let S_S be a predefined set of sensitive attribute types, such as IP addresses and MAC addresses. This set is derived from regulations, empirical analysis, or industry consensus for software logs (Aghili et al., 2025). Let $\psi(\ell)$ be a sensitive-information detector that extracts a multiset of spans from the log message ℓ, and let $\mathrm{type}(x)$ return the attribute type of span x. We define the exposure as follows:

$$\mathrm{Exposure}(\ell; S_S, \psi) = \sum_{x \in \psi(\ell)} \mathbb{I}\big(\mathrm{type}(x) \in S_S\big).$$

Optionally, a dataset-level exposure score can be computed by averaging over a log subset $\mathcal{D} \subseteq \mathcal{L}$:

$$\mathrm{Exposure}(\mathcal{D}; S_S, \psi) = \frac{1}{|\mathcal{D}|} \sum_{\ell \in \mathcal{D}} \mathrm{Exposure}(\ell; S_S, \psi).$$

Exposure provides a lightweight and annotation-free signal of remaining identifiable spans. This aligns with identifier-removal effectiveness measures commonly used in the evaluation of anonymized text. When token or span-level ground truth annotations exist, researchers may further report precision, recall, and F1 scores for sensitive-span removal (Ren et al., 2025; Pilán et al., 2022). Note that Exposure is dependent on the detector. That is, it reflects both the quality of anonymization and the quality of detection, and it should be interpreted accordingly.

To quantify the utility cost of anonymization, we measure the performance drop on a downstream task Downstream Model Performance Degradation. Such tasks include anomaly detection, failure diagnosis, and log parsing. Let $\mathcal{A}$ be a fixed learning algorithm and $\mathcal{M}$ be a task metric, such as F1 or Accuracy. A task-aware evaluation can be conducted by training the downstream model once on original training logs and then swapping the original and anonymized inputs at evaluation time. This follows the task-sensitivity perspective in anonymization benchmarking (Loiseau et al., 2025):

$$g = \mathrm{Train}(\mathcal{A}, \mathcal{D}_{\mathrm{orig}}^{\mathrm{train}}),$$

$$\Delta_{\mathcal{M}} = \mathcal{M}\big(g; \mathcal{D}_{\text{test, orig}}\big) - \mathcal{M}\big(g; \mathcal{D}_{\text{test, anon}}\big).$$

Optionally, one may also evaluate a retraining-oriented setting by training on anonymized logs and comparing the results against the original training baseline, depending on specific deployment constraints.

The Qualitative Effectiveness Score captures expert judgment on how well anonymization protects privacy while keeping logs usable in practice. Examples of usability include readability for debugging and sufficiency for incident response. Such human-centered assessments are commonly collected through Likert-scale surveys in log-privacy studies (Aghili et al., 2025). The average score is calculated as:

$$\mathrm{Score}_{\mathrm{Qualitative}} = \frac{1}{|R|} \sum_{i=1}^{|R|} r_i,$$

where R is the set of responses for a question, such as perceived privacy effectiveness or perceived utility preservation, and r_i is the numerical value of response i, such as a value from 1 to 5.

5.4 Evaluation Practice Gap

We analyze representative methods from Section 5.1 and categorize their reported evaluation protocols in Table 5.

Our coverage analysis identifies two systematic gaps in semi-structured data evaluation—**Privacy** and **Utility**. These gaps appear to stem primarily from how existing studies scope and design their experiments, rather than from fundamental shortcomings in the available metric frameworks.

Generation method	**Validity**	**Fidelity**	**Utility**	**Privacy**
Graph data				
LLM4GraphGen (Yao et al., 2024)	✓	△	×	×
Generate-on-Graph (GoG) (Xu et al., 2024b)	×	×	△	×
Ontology-grounded constrained decoding (Feng et al., 2024)	△	△	△	×
GraphJudge (Huang et al., 2025a)	△	△	△	×
GAG (Ji et al., 2025)	✓	✓	✓	×
GraphMaster (Du et al., 2025)	△	✓	✓	×
PrivGraph (Yuan et al., 2023)	×	△	△	✓
JSON				
JSON Schema discovery (Mior, 2024)	×	✓	△	×
JSONSchemaBench (Geng et al., 2025)	✓	×	✓	×
Schema Reinforcement Learning (Lu et al., 2025)	✓	×	✓	×
ThinkJSON (Agarwal et al., 2025a)	✓	✓	×	×
RedactOR (Singh et al., 2025)	△	△	✓	✓
Log data				
LogBench (Li et al., 2024d)	×	✓	×	×
AUCAD (Zhang et al., 2025a)	△	△	△	×
Protecting Privacy in Software Logs (Aghili et al., 2025)	△	△	✓	✓

Table 5: Whether representative semi-structured generation methods explicitly evaluate each dimension in their experimental sections. ✓: explicitly evaluated; △: partially/indirectly covered; ×: not reported or not applicable.

Privacy. Across the graph, JSON, and log generation methods that were reviewed, privacy is rarely treated as a primary evaluation dimension. This property is typically measured only in studies where the main contribution is explicitly focused on preserving privacy. For example, this includes graph generation based on differential privacy or clinical de identification. This pattern shows that most papers on general purpose generation prioritize demonstrating feasibility and fidelity while leaving privacy risks unmeasured. A likely explanation is that rigorous privacy evaluation requires specific experimental commitments that many authors consider to be outside the scope of their work. These commitments include an explicit threat model as well as a concrete and reproducible protocol such as the use of differential privacy budgets with ε and δ parameters. As a result, privacy measurement remains an optional feature that is limited to literature focused on privacy rather than becoming a standard reporting dimension for semi structured generation.

Utility. While utility evaluation appears more frequently than privacy, its use in practice remains very inconsistent across the methods we reviewed. Graph synthesis papers typically measure utility through downstream learning performance such as training graph neural networks. Similarly, methods related to knowledge graph question answering treat end task accuracy as the main evidence. In contrast, research on generating JSON and log data often emphasizes structural correctness or semantic similarity without systematically showing improvements in downstream applications. This variety shows that utility is naturally difficult to standardize because it depends on the task and requires complex experimental controls. These controls include data mixing strategies, fixed training and testing splits, and specific downstream models to allow for fair comparisons. As a result, many studies choose to use validity or fidelity measures that are easier to calculate instead of using a consistent and complete utility protocol. This leads to evidence that is fragmented and can only be compared partially across different research areas.

5.5 Usages

Graph Construction and Analysis. LLM-guided graph synthesis supports four main applications and an emerging evaluation method.

First, in validator-in-the-loop knowledge graph curation, candidate triples such as facts proposed by extraction pipelines or completion models are verified by validators based on LLMs. These systems perform consistency checks and use retrieval to verify information against external sources. This process filters noisy statements and reduces the need for large-scale human validation (Boylan et al., 2024).

Second, for text-conditioned graph simulation, frameworks such as the GraphAgent-Generator, which is also known as GAG, use large-scale agent simulations based on LLMs to create dynamic social graphs with text attributes. These approaches improve structural accuracy at both small and large scales and can handle large networks through parallel processing (Ji et al., 2025).

Third, synthetic graphs and latent structures derived from LLMs serve as data augmentation for downstream graph neural networks, which are often called GNNs. For example, DemoGraph uses black-box LLMs to generate latent knowledge graphs from text prompts. It integrates these structures into the original training graph to solve problems with limited data and noise in real-world applications. The value of this augmentation is usually measured using performance metrics from downstream tasks such as node classification (Feng et al., 2025b; Ji et al., 2025).

Finally, graph-structured hallucination analysis examines the reliability of LLMs when they produce structured outputs. A study by Richardeau et al. (2025) prompts these models to reproduce standard real-world networks such as Zachary's karate club and Les Miserables. The study also asks models to generate random graphs like Erdos Renyi graphs. Researchers measure graph hallucinations using structural metrics including degree sequence statistics, spectral distances, and the Graph Atlas Distance which is based on graph edits. This work connects structural differences with external leaderboards for hallucinations.

JSON for Tool Use and Schema Automation. LLMs frequently generate JSON as a structured interface for tool invocation, workflow automation, and data exchange between different services. These are contexts where outputs must strictly follow predefined schemas. Constrained decoding is also known as grammar-constrained generation or schema-constrained generation. This process enforces adherence to specifications such as JSON Schema, regular expressions, or context-free grammars during the decoding stage. This mechanism reduces parse failures and improves the reliability of outputs that are consumable by machines in agent pipelines. Recent benchmarks on real-world schemas further describe the trade-offs between efficiency, coverage, and quality in these systems for constrained decoding (Geng et al., 2025).

JavaScript Object Notation is widely used for data interchange across application programming interfaces and data integration workflows. When schemas are missing or incomplete, large language models can improve automatically discovered schemas for this format. These models generate natural language descriptions and assign meaningful names to components that are reusable. They also filter properties that were inferred but are not useful. This process helps with validation and the use of the data in later stages (Mior, 2024).

Log for Observability and Developer Assistance. Semi structured log generation is used to create realistic and parsable telemetry for incident drills and anomaly benchmarks. This technology also assists developers with automatic logging tasks such as decisions about placement, logging levels, and message content. Furthermore, it helps share exemplars that preserve privacy. However, recent benchmarks reveal critical limitations. While LLMs often produce log messages that seem correct semantically, they frequently fail to meet strict quality requirements and project conventions at a large scale. LogBench organizes the evaluation of log statement generation and identifies significant gaps under common automatic metrics and generalization settings (Li et al., 2024d). Similarly, AL Bench emphasizes real-world constraints through dynamic evaluation. It demonstrates that code containing generated log statements often fails to compile and that log outputs at runtime can differ significantly from the ground truth (Tan et al., 2025).

These findings encourage adaptation for specific domains instead of using generic prompting. To address this, research on the AUCAD framework shows that training on alignment datasets created specifically for log generation can perform much better than generic solutions based on LLMs. This provides a practical path for adoption in software engineering (Zhang et al., 2025a).

6 Vision–Language Data

Vision-Language data consists of multimodal artifacts where visual signals are naturally paired with linguistic descriptions. This category forms the foundational training corpus for modern Vision-Language Models and Multimodal LLMs. Recent methodologies increasingly use both real and synthetic data to reduce costs, privacy concerns, and the scarcity associated with large-scale aligned corpora (Mohammadkhani et al., 2025).

Specifically, Vision-Language corpora include image-text pairs, interleaved documents such as webpages containing mixed images and text, and video-text sequences. All of these formats provide paired visual content and textual descriptions (Sun et al., 2024).

In the context of LLMs, multimodal models such as Emu (Sun et al., 2024) encode visual signals into continuous embeddings and interleave them with text tokens. These models then train a single transformer using a unified autoregressive objective, which involves predicting the next text token or regressing the next visual embedding within the multimodal sequence (Sun et al., 2024). Recent architectures such as Emu3 further advance this paradigm by tokenizing images, text, and videos into a shared discrete token space. This approach applies pure next-token prediction over the mixed multimodal stream (Wang et al., 2024c).

Cross-modal alignment operates at multiple levels of granularity. Fundamentally, Vision-Language data provides weak or global pairing between a visual unit such as an image or video clip and a textual unit such as a caption, transcript, or surrounding narrative. Beyond this global alignment, certain datasets encode fine-grained grounding signals that connect specific regions or spatio-temporal segments to linguistic entities. This allows models to connect localized perception with compositional semantic reasoning. Representative examples include region-to-phrase correspondences with bounding boxes in Flickr30k Entities (Plummer et al., 2015), dense region descriptions and object-level grounding in Visual Genome (Krishna et al., 2016), and spatio-temporal tube grounding within the VidSTG benchmark for the Spatio-Temporal Video Grounding task (Zhang et al., 2020).

6.1 Generation Methods

6.1.1 Image-Text Data

LLM-based image-text generation can be broadly categorized into two paradigms based on the generation mechanism. The first paradigm is native autoregressive generation. In this approach, the multimodal model produces visual representations within a unified token or embedding stream. This includes methods such as continuous visual-embedding regression in Emu or discrete multimodal token prediction in Emu3 (Sun et al., 2024; Wang et al., 2024c). The second paradigm is external diffusion control. In this setup, the LLM functions as a controller for a separate diffusion backbone. The model iteratively diagnoses mismatches and guides edits to improve how well the output follows the prompt (Wu et al., 2023).

Native Autoregressive Generation. This paradigm treats multimodal generation as a sequence modeling task using mixed visual and textual representations. In this framework, images and other data types are represented as continuous embeddings or discrete tokens. These elements are combined with text within a single autoregressive stream (Sun et al., 2024; Wang et al., 2024c; Team, 2025).

For example, Emu encodes images into visual embeddings that are mixed with text tokens. The model uses a transformer to predict the next text token or estimate the next visual embedding in a unified sequence (Sun et al., 2024). Emu3 further develops this approach by converting images, text, and videos into a shared space of discrete tokens. This allows the model to work entirely through next-token prediction over the mixed sequence (Wang et al., 2024c).

These unified designs support both multimodal understanding and generation within one autoregressive model. By starting with different multimodal prefixes, a single model can perform various tasks such as image captioning, visual question answering, and image generation (Sun et al., 2024; Wang et al., 2024c). Team (2025) also shows the effectiveness of early-fusion modeling when applied directly to long sequences of interleaved image and text tokens.

External Diffusion Control. In contrast to generating visual representations directly within the LLM, this paradigm uses the model as a high-level controller. This controller routes intent and conditioning signals to specialized generative tools such as diffusion models (Pan et al., 2024; Koh et al., 2023).

Systems such as Kosmos-G (Pan et al., 2024) first align the outputs of a multimodal model to a CLIP-anchored conditioning interface through supervised alignment. They then apply score-distillation instruction tuning where a frozen diffusion decoder provides training signals. This process enables the controller to produce representations that guide high-fidelity image synthesis.

Similarly, GILL (Koh et al., 2023) connects a frozen text-only LLM with pretrained image generation backbones using lightweight mapping modules. These modules translate the hidden representations of the model into the embedding space of the generator. This allows downstream decoders such as diffusion-based generators to render the requested images.

This modular design separates intent understanding and planning from high-fidelity rendering. In this setup, the LLM manages the planning while a diffusion backbone executes the rendering. This separation enables precise control through an explicit conditioning interface. In systems like Kosmos-G, the visual generator can be upgraded or replaced while the controller remains largely the same. In mapping-based designs such as GILL, replacing the generator typically requires retraining the bridging module rather than the entire LLM (Pan et al., 2024; Koh et al., 2023).

In summary, while native autoregressive generation provides a unified space for both reasoning and rendering, external diffusion control makes better use of specialized vision models and established generation tools.

6.1.2 Video-Text Modeling

Video-text generation extends the scope of LLM synthesis from static visual domains to spatiotemporal media. Similar to the image domain, contemporary methodologies generally fall into two distinct paradigms. The first is native spatiotemporal generation, where the model directly synthesizes video and often audio representations within a unified multimodal sequence. The second is planner-based diffusion control, where the LLM manages high-level video structure for execution by a separate rendering engine.

Native Spatiotemporal Generation. Approaches such as VideoPoet (Kondratyuk et al., 2024) and Emu3 (Wang et al., 2024c) treat video generation as a native language modeling task over spatiotemporal tokens. In the case of VideoPoet, this also includes audio. VideoPoet uses a decoder-only transformer to process multimodal inputs including images, video clips, text, and audio. The model is trained using multimodal generative objectives in an autoregressive manner (Kondratyuk et al., 2024). Similarly, Emu3 converts video frames into a discrete latent space and learns to predict the next token. This approach effectively unifies video understanding and generation under a single backbone (Wang et al., 2024c).

By representing video and sometimes audio as discrete tokens and training a decoder-only transformer with autoregressive next-token objectives, these methods unify multimodal conditioning and generation within a language-model style backbone. This strategy avoids the use of diffusion-based generation heads. In practice, these systems still rely on specific tokenizers or decoders and sometimes use additional modules such as token-space super-resolution. Understanding capabilities such as captioning depend on the mixture of tasks and the post-training setup. Furthermore, the models learn long-range temporal dependencies, such as motion coherence and scene continuity, end-to-end through sequence modeling.

Planner-based Diffusion Control. In planner-based architectures such as FlowZero (Lu et al., 2023), LVD (Lian et al., 2024), and VideoDirectorGPT (Lin et al., 2024), the LLM functions as a director rather than a renderer. It translates natural language prompts into structured and interpretable intermediate representations. These representations include dynamic scene syntaxes or spatiotemporal layouts such as per-frame scene descriptions, object layouts, bounding-box trajectories, and background motion patterns. These plans then condition a separate diffusion-based video generator.

For instance, FlowZero uses a dynamic scene syntax generated by a LLM to guide frame-wise diffusion and improve temporal smoothness. This includes coherent object motion as well as controllable background and camera motion patterns (Lu et al., 2023). LVD similarly uses the LLM as a spatiotemporal planner that outputs dynamic scene layouts, which are typically frame-consistent bounding-box sequences. These layouts are injected into the video diffusion model to enforce spatial relations and motion consistency (Lian et al., 2024). VideoDirectorGPT further decomposes long prompts into editable multi-scene video plans and consistency groupings. It generates scene-level and frame-level layouts before invoking downstream diffusion modules, which enables multi-scene narrative structure and character consistency (Lin et al., 2024).

By separating high-level planning from low-level frame synthesis, planner-based control provides transparent and editable handles over visual content and motion dynamics. This paradigm also offers natural anchor points for integrating external constraints such as timeline editing, user edits, or guardrails at the planning stage before invoking the diffusion generator.

Across both paradigms, there is a converging trend to treat structural integrity and provenance as native components of the generation pipeline rather than retroactive additions. First, temporal alignment and long-horizon consistency are increasingly treated as primary objectives in long-form video generation. Second, to address generative authenticity, researchers are integrating provenance mechanisms such as Content Credentials with cryptographically signed metadata following the C2PA standard (Coalition for Content Provenance and Authenticity, 2025). They are also including robust invisible watermarking for video. This involves watermarking designed for latent video diffusion models (Jang et al., 2025) as well as visual-audio watermarking for manipulation localization and copyright protection (Zhang et al., 2024e). These tools enable machine-checkable attribution and manipulation tracing. Consequently, modern video-text systems are evolving from loosely coupled toolchains into more integrated frameworks that jointly optimize expressivity, structural consistency, and governance.

6.2 Quality Metrics for Vision-Language Data

6.2.1 Image-Text

Validity. Validity ensures that generated vision-language content is both well-formed and verifiable. This means that the content can be parsed under a predefined machine-readable structure and checked for basic consistency constraints. In practice, a valid output must follow the target schema and maintain consistent internal references. This includes features such as stable identifiers or pointers between turns when the format requires them. Common structural failures such as malformed schemas can be reduced through formally constrained decoding. For instance, grammar-based engines such as DOMINO align grammatical constraints with the subword vocabulary of the model to guarantee the structural correctness of machine-readable outputs (Beurer-Kellner et al., 2024).

Given a set of vision-language generations $\mathcal{M}_{\text{gen}} = \{m_1, \ldots, m_N\}$ and a schema S, we define the Well-Formed Rate as:

$$\text{WFR} = \frac{1}{|\mathcal{M}_{\text{gen}}|} \sum_{m \in \mathcal{M}_{\text{gen}}} V(m, S),$$

where $V(m, S) \in \{0, 1\}$ indicates whether m conforms to S.

Fidelity. Semantic-level fidelity measures the logical and contextual alignment between text and non-text modalities within a generated interleaved sequence $m = \{(t_\ell, v_\ell)\}_{\ell=1}^{L}$. In this notation, ℓ represents the individual steps within a single vision-language sample. A common approach is to score each pair using an alignment function $\mathcal{S}_{\text{align}}$. In practice, this function is often implemented in one of two ways. The first way uses a strong multimodal judge model such as GPT-4o. The second way uses embedding-space alignment such as cosine similarity between CLIP text and image features, which is also known as text alignment (Ham et al., 2024). We aggregate step-wise alignment as follows:

$$\text{ITA}(m) = \frac{1}{L} \sum_{\ell=1}^{L} \mathcal{S}_{\text{align}}(t_\ell, v_\ell).$$

The CoMM evaluation framework also reports an Image-Text Alignment score using strong judge models for a similar purpose(Chen et al., 2025b).

Beyond global alignment, instance-level fidelity measures how accurately specific attributes are preserved. Following standard protocols for evaluating subject-driven generation (Ruiz et al., 2023), we compute Subject Fidelity between a source subject image V_{subj} and a generated image V_{gen}. This is calculated using an image embedding extractor $E_{\text{img}}(\cdot)$ such as the CLIP image encoder or DINO:

$$\text{Fidelity}_{\text{subj}} = \text{sim}\big(E_{\text{img}}(V_{\text{gen}}),\ E_{\text{img}}(V_{\text{subj}})\big),$$

In this equation, $\mathrm{sim}(\cdot,\cdot)$ is typically cosine similarity. We capture Prompt Fidelity, which refers to how well the model follows the prompt, using CLIP-style alignment between text and images (Ruiz et al., 2023):

$$\mathrm{Fidelity}_{\mathrm{prompt}} = \mathrm{sim}\big(E_{\mathrm{img}}(V_{\mathrm{gen}}),\, E_{\mathrm{text}}(T_{\mathrm{prompt}})\big).$$

When reference data is available, we also report standard text metrics such as ROUGE and METEOR. For image evaluation, we use metrics such as the Frechet Inception Distance and the Inception Score to measure image quality. We also use the Structural Similarity Index for style consistency and the Peak Signal-to-Noise Ratio for reconstruction when applicable. These metrics are measured against a reference text T_{ref} or a reference image V_{ref} (Chen et al., 2025b).

We use the Caption Hallucination Assessment with Image Relevance (CHAIR) metric to measure how often models mention objects that are not in the image (Rohrbach et al., 2019). We extract mentions from captions and compare them to eighty object categories from the Microsoft Common Objects in Context dataset. A mention is a hallucination if it does not appear in the ground truth labels.

The metric provides two scores. The object level rate is the fraction of all mentions that are hallucinated. The sentence level rate is the fraction of all sentences with at least one hallucination. Lower scores indicate higher grounded fidelity.

$$\mathrm{CHAIR}_i = \frac{\text{Number of hallucinated objects}}{\text{Total object mentions}}$$

$$\mathrm{CHAIR}_s = \frac{\text{Number of sentences with hallucinations}}{\text{Total number of sentences}}$$

Utility. Utility measures the value of a model for downstream tasks such as image captioning and visual question answering. It also applies to tool-augmented workflows. This is typically assessed through performance on downstream tasks and evaluations by automatic or human judges. Unified early-fusion models such as Chameleon demonstrate interleaved reasoning and generation over mixed-modal sequences (Team, 2025). Similarly, Anole provides an open-source autoregressive model along with a training framework (Chern et al., 2024). Interleaved image and text training data such as CoMM (Chen et al., 2025b) and instruction-driven multi-turn dialogues such as InterSyn (Feng et al., 2025a) further support supervised instruction tuning and evaluation.

Let $D = \{(q_i, a_i)\}_{i=1}^{N}$ denote a benchmark dataset consisting of N question and answer pairs where q_i is the ith question and a_i is its corresponding ground truth answer. We use the notation $|D|$ to represent the size of the dataset which is equal to N. Let M be the model or the inference function that maps an input question q_i to a predicted answer.

We use the indicator function which equals one if its argument is true and zero otherwise. The equality predicate $M(q_i) = a_i$ indicates an exact match comparison between the predicted answer and the reference answer. This process is optionally performed after applying a standard normalization procedure such as lowercasing and punctuation stripping when the task is open ended. The accuracy for question answering is computed as the average indicator value over all samples in the dataset:

$$\mathrm{Acc}_{\mathrm{QuestionAnswering}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(M(q_i) = a_i)$$

For human evaluation of utility, we adopt a pairwise preference protocol following the relative evaluation setting in Chameleon (Team, 2025). Given the same input, annotators are shown two anonymized responses from our model M and a baseline B in a randomized order. The annotators then select one of the following options: M is better, B is better, or about the same. We define a win as the case where annotators prefer the output of M due to better task fulfillment and usefulness. Similarly, a loss occurs when B is preferred and a tie occurs when the two outputs are judged to be comparable. We compute the win rate by awarding one point for a win and half a point for a tie (Team, 2025):

$$\mathrm{Win}_{\mathrm{Rate}} = \frac{W + 0.5 \times T}{W + T + L},$$

where W, T, and L denote the number of wins, ties, and losses of M against B respectively.

Diversity. Diversity measures the range of entities, attributes, styles, and layouts while preserving the agreement between text and images. We measure diversity at three different levels. The first level is distributional diversity over the entire set. The second level is intra-sequence diversity within a single sample. The third level is judge-based diversity over multimodal outputs.

From an architectural perspective, MM-Interleaved improves multi-image interleaved generation by synchronizing access to detailed visual features during the generation process (Tian et al., 2024). Chameleon uses a unified token space (Team, 2025). InterSyn increases coverage with multi-turn dialogues driven by instructions. These dialogues are created through a process known as iterative refinement during data creation (Feng et al., 2025a).

Let $\mathcal{V}$ be the set of generated images. Following Salimans et al. (2016), we evaluate sample quality and diversity using the Inception Score. Specifically, we apply a pretrained Inception classifier to each generated image $v \in \mathcal{V}$ to obtain a conditional label distribution $p(y \mid v)$.

Images that contain meaningful and recognizable objects tend to yield low entropy conditional label distributions where the classifier shows high confidence. In contrast, a generator that avoids mode collapse should produce varied images such that the marginal label distribution $p(y) = \mathbb{E}_{v \in \mathcal{V}}[p(y \mid v)]$ has high entropy. This ensures that predictions are spread across many different classes (Salimans et al., 2016). Combining these two goals, the Inception Score is defined as:

$$\mathrm{IS}(\mathcal{V}) = \exp\Big(\mathbb{E}_{v \in \mathcal{V}}\big[D_{\mathrm{KL}}(p(y \mid v) \,\|\, p(y))\big]\Big).$$

Equivalently, $\log \mathrm{IS}(\mathcal{V}) = H(p(y)) - \mathbb{E}_{v \in \mathcal{V}}[H(p(y \mid v))]$. The second term encourages sharpness and recognizability through low conditional entropy. The first term encourages diversity by rewarding marginal distributions with high entropy and D_{KL} means KL Divergence. This represents broad coverage over the semantic categories predicted by the Inception classifier (Salimans et al., 2016).

Let $m_v = \{v_\ell\}_{\ell=1}^{L}$ be a visual sequence such as a series of generated images or video frames. In this sequence v_ℓ is the visual element at position ℓ and L represents the total length of the sequence. Let E be a feature extractor that transforms a visual input into an embedding vector. This extractor can be an image encoder such as Contrastive Language Image Pretraining or Self Distillation with no Labels. The term sim refers to a similarity function between embeddings and this value is typically determined by cosine similarity.

We define the Intra-Sequence Diversity as one minus the average pairwise similarity among all unordered pairs in the sequence as follows:

$$\mathrm{Div}_{\mathrm{seq}}(m_v) = 1 - \frac{2}{L(L-1)} \sum_{\ell=1}^{L} \sum_{r=\ell+1}^{L} \mathrm{sim}\big(E(v_\ell), E(v_r)\big).$$

Let $\mathcal{M}_{\mathrm{generations}}$ be a set of multimodal generations and let $|\mathcal{M}_{\mathrm{generations}}|$ represent the total number of items in that set. For each generated sample m in the set of generations the term $\mathcal{J}_m$ represents the diversity score. This score is assigned by a judge model or a human rater or an automatic rater to evaluate the diversity of the output. We calculate the aggregated judge based diversity by taking the average of these scores over the entire set of generations to obtain the final diversity score.

$$\mathrm{Score}_{\mathrm{diversity}} = \frac{1}{|\mathcal{M}_{\mathrm{generations}}|} \sum_{m \in \mathcal{M}_{\mathrm{generations}}} \mathcal{J}_m$$

6.2.2 Video-Text

Validity. For video-text samples, we define validity using a set of verifiable sub-properties. These properties include schema-level well-formedness, temporal correctness, and cross-modal binding consistency. Let $\mathcal{M}$ be the set of N generated video and text samples where the size of the set is N. Each sample m_i consists of a video v_i and a paired text t_i and the number of modalities is two.

Let $\mathbb{S}$ denote a machine readable schema such as a typed JavaScript Object Notation or Yet Another Markup Language schema with field constraints used to validate structured outputs. The function denoted by the term conforms checks whether sample m_i adheres to schema $\mathbb{S}$ and returns true or false. Let $\mathbb{I}$ be the indicator function which returns one if its argument is true and zero otherwise. We define the Schema Adherence Rate as the fraction of generated samples that pass the schema validation:

$$\text{SchemaAdherenceRate} = \frac{1}{|\mathcal{M}|} \sum_{m_i \in \mathcal{M}} \mathbb{I}[\text{conforms}(m_i, \mathbb{S}) = \text{true}]$$

In the evaluation of temporal correctness, we follow the established protocols used in temporal language grounding such as TALL (Gao et al., 2017). We measure correctness using recall at various Intersection over Union thresholds. In this framework, $T^{(j)}_{\text{pred},i}$ represents the j-th ranked predicted interval for sample i based on the model score, and $T_{\text{gt},i}$ represents the ground truth interval. We define the Recall @ K metric for a given Intersection over Union threshold δ as follows:

$$\text{R@K}(\delta) = \frac{1}{|\mathcal{M}|} \sum_{m_i \in \mathcal{M}} \mathbb{I}\Big(\max_{1 \le j \le K} \text{IoU}\big(T^{(j)}_{\text{pred},i}, T_{\text{gt},i}\big) \ge \delta \Big).$$

In practice, we report this metric for standard choices such as K values of 1, 5, or 10 and threshold values of 0.3, 0.5, or 0.7. Additionally, a comprehensive temporal score can be calculated by taking the average across a specific set of thresholds Δ:

$$\text{R@K-Avg} = \frac{1}{|\Delta|} \sum_{\delta \in \Delta} \text{R@K}(\delta).$$

Regarding cross-modal binding consistency, we treat cross-modal binding as a validity property that can be verified through representation agreement. Let f_T and f_V be text and video encoders, and let ϕ_{sim} represent cosine similarity. For each sample, we assume that a grounded video segment v_i^{seg} is referenced by text t_i. The Cross-Modal Consistency score is then defined as follows:

$$\text{CMC} = \frac{1}{|\mathcal{M}|} \sum_{m_i \in \mathcal{M}} \phi_{\text{sim}}\big(f_T(t_i), f_V(v_i^{\text{seg}})\big).$$

This equation naturally extends to aligned speech or text regions by using automatic speech recognition transcripts or optical character recognition snippets when they are available.

Fidelity. Regarding video fidelity, this property reflects both spatio-temporal coherence and visual realism. Let F be the number of frames in v_i and let τ represent the frame indices. To evaluate object identity consistency over tracked object regions $\{o_{i,\tau}\}$, we use a visual feature extractor f_E such as CLIP and a similarity function ϕ_{sim}. The identity consistency score is defined as follows:

$$\mathcal{F}_{\text{ID}}(m_i) = \frac{1}{F-1} \sum_{\tau=1}^{F-1} \phi_{\text{sim}}\big(f_E(o_{i,\tau}), f_E(o_{i,\tau+1})\big).$$

Holistic realism and distributional faithfulness are often summarized by the Frechet Video Distance. This metric was originally proposed for the evaluation of generative video models (Unterthiner et al., 2019). It is now widely reported in modern text-to-video evaluations including VideoPoet and various planner-based systems (Kondratyuk et al., 2024; Lin et al., 2024).

Additionally, we can aggregate frame-level or clip-level quality scores from specialized predictors to define a general video quality metric:

$$\mathcal{F}_{\mathrm{VQ}}(m_i) = \frac{1}{F}\sum_{\tau=1}^{F}\Psi(v_{i,\tau}),$$

In this equation, Ψ represents one or more automated evaluators that target dimensions such as imaging and aesthetics. These evaluators are typically organized in standardized suites such as VBench (Huang et al., 2024b).

Utility. Utility represents how reliably synthetic video and text data supports controllable generation and downstream usage. For planner style pipelines that generate structured plans or prompts such as VideoDirectorGPT (Lin et al., 2024), we let P represent a set of plans or prompts and $v_{\mathrm{out}}^{(j)}$ be the output video conditioned on a specific plan p_j. Inspired by planner-based pipelines such as VideoDirectorGPT, we define a controllability oriented Task Success Rate as follows:

$$\mathcal{U}_{\mathrm{Control}} = \frac{1}{|P|}\sum_{p_j\in P}\mathbb{I}\big(\mathrm{eval}_{\mathrm{task}}(v_{\mathrm{out}}^{(j)}, p_j) = \mathrm{true}\big),$$

In this formulation, the task evaluation function can be implemented through automatic checkers, learned judges, or human evaluation. The choice of method depends on the controllability constraints that are encoded by the plan p_j.

For cases involving specific styles or functions, we use an embedding-based proxy for usefulness:

$$\mathcal{U}_{\mathrm{Func}} = \phi_{\mathrm{sim}}\big(f_T(t_{\mathrm{style}}), f_V(v_{\mathrm{out}})\big),$$

This metric measures whether the generated video matches a target style or condition within a shared representation space.

Diversity. Regarding video diversity, this metric reflects both motion variety and semantic breadth. We define $\mathcal{V} = \{v \mid (v,t)\in\mathcal{M}\}$ as the set of generated videos.

To evaluate dynamic variety, we measure motion magnitude using a motion score based on optical flow(Teed & Deng, 2020). This score is calculated using the following equation:

$$\mathrm{motion\_score}(v) = \frac{1}{T-1}\sum_{\tau=1}^{T-1}\frac{1}{|\Omega|}\sum_{\mathbf{x}\in\Omega}\left\|\mathbf{f}_{\tau\to\tau+1}(\mathbf{x})\right\|_2,$$

In this formulation, $f_{\tau\to\tau+1}$ represents the optical flow field from frame τ to frame $t+1$ over the pixel domain Ω, and T represents the number of frames. In practice, we calculate optical flow using methods such as RAFT and aggregate the average flow magnitude over all frames to determine the motion score for each video (Liao et al., 2024).

In terms of semantic diversity, we use a metric similar to the Inception Score over videos. This approach is widely adopted in the evaluation of video generation. Following the standard Inception Score formulation (Salimans et al., 2016), we define the metric as follows:

$$\mathcal{D}_{\mathrm{sem}} = \exp\big(\mathbb{E}_{v\sim\mathcal{V}}\left[D_{KL}\big(p(y|v)\,\|\,p(y)\big)\right]\big),$$

In this definition, $p(y|v)$ is the label distribution predicted by a pretrained classifier(Saito et al., 2017). The term $p(y)$ represents the marginal label distribution.

6.3 Trustworthy Metrics for Vision-Language Data

6.3.1 Image-Text

Safety. Safety aims to prevent the generation of harmful content across different modalities. It also focuses on reducing multimodal attack surfaces such as image-based prompt attacks and jailbreaking conditioned on images (Liu et al., 2024c). The MM-SafetyBench framework investigates these prompt attacks and demonstrates that images relevant to a query can significantly increase the success rate of an attack. This finding supports the hypothesis that query-relevant images activate the alignment module for vision and language. Since this module is often trained without specific safety constraints, its activation can weaken the ability of the model to identify harmful requests (Liu et al., 2024c).

Furthermore, the benchmark notes that a low attack success rate is not always clear. It might indicate that the model is properly aligned for safety, or it could simply mean that the model fails to understand the malicious query. To address this uncertainty, MM-SafetyBench emphasizes refusal behavior. This approach helps to distinguish between a model that recognizes and refuses a harmful request and one that simply fails to comply because of a lack of understanding. In practice, achieving a low attack success rate while keeping the expected refusal behavior remains a difficult task (Liu et al., 2024c).

The benchmark also shows that adding a safety prompt can significantly lower attack success rates for models that are better at following instructions. For other models, these improvements are smaller. The authors suggest that the effectiveness of safety prompts depends on how well the underlying model follows instructions (Liu et al., 2024c).

Following MM-SafetyBench, we report Attack Success Rate (ASR) and Refusal Rate (RR) on a query set D:

$$\mathrm{ASR} = \frac{\sum_{q \in D} I(q)}{|D|}, \qquad \mathrm{RR} = \frac{\sum_{q \in D} R(q)}{|D|}.$$

Here $I(q) = 1$ if and only if the model engages with the disallowed request under query q, for example by providing prohibited content or actionable instructions; otherwise $I(q) = 0$ (Liu et al., 2024c). We set $R(q) = 1$ if and only if the model begins with a refusal to satisfy the unsafe query, using the refusal definition in MM-SafetyBench; otherwise $R(q) = 0$ (Liu et al., 2024c).

In implementation, refusal detection can be instantiated with a predefined refusal-pattern matcher, and it can be strengthened with a judge Φ_S for more robust refusal identification. We report RR over all queries in D. To quantify over-sensitivity at the system level, we additionally evaluate a benign subset D_{benign} constructed separately, and report $\mathrm{RR}_{\text{benign}}$, the refusal rate on benign queries. This serves as a practical system-level proxy inspired by safety-awareness benchmarks such as MMSafeAware (Wang et al., 2025), which include benign subsets to capture harmless requests that are mistakenly treated as unsafe. MMSafeAware focuses on safety awareness and recognition, whereas $\mathrm{RR}_{\text{benign}}$ operationalizes the downstream refusal behavior under benign inputs.

Provenance. Regarding provenance, this property tracks the verifiable origin, attribution claims, and modification history for multimodal outputs. C2PA version 2.2 provides signed content credentials that support an auditable provenance chain (Coalition for Content Provenance and Authenticity, 2025). In addition, SynthID-Image enables invisible watermarking for imagery generated by artificial intelligence to facilitate detection and traceability (Gowal et al., 2025; Google DeepMind, 2025). We define the credential validation rate over the set of generated multimodal samples as follows:

$$\mathrm{Rate}_{\text{validate}} = \frac{1}{|\mathcal{M}_{\text{gen}}|} \sum_{m \in \mathcal{M}_{\text{gen}}} V_{\text{c2pa}}(m),$$

In this definition, the indicator function V_{c2pa} results in a value of either zero or one. The value is zero when no C2PA manifest or credentials are present. The value is one when the validator returns a manifest state that is at least Valid, or Trusted under a stricter setting, depending on the chosen trust model. To clarify, the manifest states of Well-Formed, Valid, and Trusted reflect three distinct properties. These include the well-formedness and cryptographic integrity of the manifest, the verification of the signature, and the trust status of the signing credential under the adopted trust configuration (Coalition for Content Provenance and Authenticity, 2025).

6.3.2 Video-Text

Safety. The Harmful Content Rate measures the failure to prevent video generation that violates policies during adversarial evaluation. This approach is inspired by T2VSafetyBench, which creates malicious prompts such as real-world prompts, prompts generated by GPT-4, and prompts based on jailbreak attacks. That framework evaluates videos using a GPT-4 based assessment together with manual review for validation. We define the Harmful Content Rate using a safety evaluation pipeline Φ_S. This pipeline may involve judging by LLMs along with optional human review and detector-based checks (Miao et al., 2024).

We define this metric as follows:

$$\text{HCR} = \frac{|\{v \in \mathcal{V}_{\text{adv}} \mid \text{is\_unsafe}(v, \Phi_S) = \text{true}\}|}{|\mathcal{V}_{\text{adv}}|},$$

In this equation, $\mathcal{V}_{\text{adv}}$ represents the set of all successfully generated videos under adversarial prompts. Beyond measuring the number of unsafe generations, T2VSafetyBench provides three findings that are useful for system-level safety evaluation. First, no single model performs well across all safety aspects. Second, assessments by GPT-4 correlate well with manual reviews, which supports the use of large-scale automatic judging. Finally, a trade-off exists between model capability and safety. This suggests that safety risks may increase as video generation technology improves (Miao et al., 2024).

Provenance. Watermark-based provenance evaluates the robustness and the trade-off between payload and quality for generated videos. When using an original watermark w that consists of L bits, an attack $\mathcal{A}$, and a recovery procedure $\mathcal{R}$, we measure robustness using bit accuracy. For a generated video v, this is defined as follows:

$$\mathcal{P}_{\text{Rob}} = 1 - \frac{d_H\big(w, \mathcal{R}(\mathcal{A}(v))\big)}{L}.$$

We define the payload and fidelity cost using the following equations:

$$\mathcal{P}_{\text{Load}} = L, \qquad \mathcal{P}_{\text{Cost}} = \Delta\big(V_{\text{ori}}, v\big).$$

In these expressions, $\mathcal{P}_{\text{Load}}$ represents the payload per video in bits. The variable V_{ori} represents the corresponding baseline video that was generated without a watermark using the same prompt and seed. The term Δ measures the perceptual distortion that is introduced by the process of embedding the watermark. For example, this can be calculated by averaging the frame-wise Learned Perceptual Image Patch Similarity over sampled frames to measure frame-level distortion. One can also use the Frechet Video Distance to evaluate quality degradation at the level of the distribution.

These estimates can be used together with signed credentials and invisible watermarking for video content. Specific examples include LVMark for latent video diffusion model watermarking and V2A-Mark for visual and audio watermarking with tamper localization. These tools support traceability and detection in downstream tasks. Furthermore, they can complement provenance mechanisms that are cryptographically verifiable such as C2PA (Coalition for Content Provenance and Authenticity, 2025; Jang et al., 2025; Zhang et al., 2024e).

6.4 Evaluation Practice Gap

We analyze representative methods from Section 6.1 and categorize their reported evaluation protocols in Table 6.

Diversity. Compared with fidelity, diversity is less consistently evaluated in research related to vision and language generation. A practical reason is that commonly reported diversity proxies such as the Inception Score combine quality and coverage. However, stronger measures such as intra sequence embedding diversity, judge based diversity, or video diversity based on motion entropy require additional modeling choices. These measures also require reliable feature extractors or judges and careful control of prompt distributions. In addition, diversity is closely linked with alignment. Increasing variety in a simple way can reduce the agreement between text and images. For this reason, many papers choose to prioritize faithful rendering over broad coverage. As a result, diversity is frequently treated as a qualitative claim or a secondary proxy. This leaves a gap for standardized and reproducible diversity panels that separate breadth from alignment.

Generation method	**Validity**	**Fidelity**	**Diversity**	**Utility**	**Safety**	**Provenance**
Kosmos-G (Pan et al., 2024)	×	✓	×	△	×	×
GILL (Koh et al., 2023)	×	✓	×	✓	×	×
LVD (Lian et al., 2024)	△	✓	×	✓	×	×
FlowZero (Lu et al., 2023)	△	✓	×	△	×	×
VideoDirectorGPT (Lin et al., 2024)	✓	✓	✓	✓	×	×
SynthID-Image (Gowal et al., 2025)	×	✓	×	×	×	✓

Table 6: Whether representative vision–language methods explicitly evaluate each dimension in their experimental sections. ✓: explicitly evaluated; △: partially/indirectly covered; ×: not reported or not applicable.

Safety. Although vision–language models are becoming widely used, most evaluations still focus mainly on fidelity metrics such as realism and alignment. In contrast, safety is often treated as a secondary concern. It is usually addressed only through indirect measures such as brief qualitative discussions or data filtering rather than being measured in a systematic way. One major challenge is that multimodal safety involves a wide range of potential attacks. Examples include visual prompt injections and jailbreaks that are conditioned on images. Furthermore, a low success rate for attacks can be difficult to interpret. This result might mean that the model is safe or it might simply mean that the model failed to understand the malicious query. Finally, safety evaluation has two goals. These include stopping harmful responses and avoiding the refusal of safe requests. This balance makes it both difficult and costly to standardize safety reports.

Provenance. Provenance metrics are rarely reported in standard generation papers. This is mainly because measuring provenance requires additional infrastructure such as cryptographic credentials or watermarking systems. It also requires testing the model against changes in the real world such as compression, cropping, re encoding, or editing. However, as generative models become more capable and synthetic content spreads more widely, establishing verifiable provenance through robust and measurable watermarks is becoming essential for tracking content, ensuring accountability, and building trust.

6.5 Usages

Recent methodologies increasingly use synthetic vision and language data across four primary functional paradigms. This trend is largely independent of the underlying generation backbone. These paradigms include supervised finetuning, preference based alignment and reward modeling, data curation and remediation for weak supervision, and evaluation centered on safety or structure. In these contexts, synthetic image and text, video and text, and document vision and language samples serve not only as supplementary data but also as scalable supervision sources, explicit alignment signals, and controllable probes for assessing robustness and safety.

Supervised Finetuning. Significant research effort has been directed toward constructing large-scale and structured multimodal instruction datasets through synthetic captioning and automated prompt curation to drive supervised finetuning. Broadly speaking, these initiatives scale supervised finetuning along two principal axes. The first axis is modality coverage, which extends from static images to video. The second axis is semantic granularity, which progresses from generic captions to instance-level and OCR-aware instructions. Advancements along both of these axes have demonstrated strong correlations with improvements in downstream performance.

In the video domain, LLaVA-Video-178K introduces a largely synthetic instruction-following corpus that was assisted by GPT-4o and included human involvement in the pipeline. This dataset comprises detailed captions, open-ended question answering, and multiple-choice tasks, which yields consistent improvements in video large multimodal model training (Zhang et al., 2025e). To address text-rich imagery, TextSquare, also known as Square-10M, leverages closed-source multimodal LLMs to scale synthetic image and text instructions to the order of tens of millions. When this approach is utilized for large-scale supervised finetuning, it exhibits near-monotonic performance gains relative to the scale of the data (Tang et al., 2025).

Similarly, LLaVAR-2 integrates human-annotated captions with filtered instructions generated by GPT-4o to curate a high-quality and text-centric supervised fine-tuning dataset (Zhou et al., 2024). For the purpose of targeting instance-level grounding, Inst-IT synthesizes explicit visual-prompted instructions for continuous supervised fine-tuning. This method enhances instance-grounded understanding and demonstrates effective transfer to generic benchmarks (Peng et al., 2024). Concurrently, dense and high-fidelity synthetic captions have proven effective for supervision. For example, ShareGPT4V and ShareGPT4Video demonstrate that large-scale synthetic captioning directly benefits both supervised finetuning and unified large vision-language model pre-training (Chen et al., 2023b; 2024a).

Preference alignment and reward modeling. Beyond standard Supervised Finetuning, synthetic data plays a pivotal role in constructing pairwise preferences and training reward models. For video generation, VideoDPO proposes an automatic pipeline that generates multiple videos for each prompt and scores them using a multi-dimensional metric called OmniScore. This pipeline forms score-ranked preference pairs consisting of the best and worst samples. These pairs are further re-weighted during Direct Preference Optimization to prioritize distinctive and high-impact samples (Liu et al., 2024a).

Similarly, the work titled Improving Video Generation with Human Feedback leverages multi-dimensional human preferences regarding synthetic videos to train a VideoReward model. This reward model then guides generators through both Direct Preference Optimization objectives and reward-weighted inference (Liu et al., 2025a). In the domain of vision and language, VLFeedback provides more than 82,000 feedback instances annotated by artificial intelligence. These instances include preference labels and rationales across helpfulness, visual faithfulness, and ethical and safety aspects. Applying Direct Preference Optimization to VLFeedback to train the Silkie model improves helpfulness, visual faithfulness, and safety-related metrics. Experimental results show that applying this optimization method to such feedback significantly enhances the helpfulness, visual faithfulness, and safety of the models (Li et al., 2024b).

Data curation, error correction, and failure-targeted synthesis. Synthetic data is also instrumental in refining noisy real-world data, repairing weak labels, and generating hard negatives to expose model failures. For example, CapFusion leverages a LLM to consolidate and refine information from both noisy web-based image and text pairs and model-generated synthetic captions. This process produces higher-quality and more scalable supervision for multimodal pre-training (Yu et al., 2024b).

In the area of structured graphics, CHOCOLATE provides a human-annotated benchmark and a typology of factual errors in chart captions. This work also establishes the task of factual error correction for chart captions. It further introduces ChartVE, which is a reference-free visual entailment metric for chart and caption factuality. Additionally, the authors propose C2TFec, an interpretable two-stage framework that converts charts into tables before using a LLM to rectify factual inconsistencies (Huang et al., 2024a).

For document understanding, SynthDoc synthesizes bilingual document images containing text, tables, and charts. This study demonstrates that training OCR-free parsers similar to the Donut model on such data improves pre-training read tasks and downstream robustness, even when language inconsistencies are present (Ding et al., 2024).

Safety alignment and evaluation. Beyond general capabilities, synthetic data is essential for red teaming techniques. These techniques involve generating harmful and helpful responses along with preference pairs to stress-test the safety of vision and language models. For instance, SPA-VL creates 100,000 image-instruction quadruples that feature chosen and rejected responses across six specific harm domains. This dataset is designed to support safety training based on Proximal Policy Optimization or Direct Preference Optimization (Zhang et al., 2025d). Beyond alignment based on preferences red teaming also requires stress testing failure modes that could lead to behavior that is not safe in an indirect manner. This includes visual scenes containing a large amount of text where optical character recognition and misunderstandings of the layout can result in interpretations that are harmful. Regarding evaluation, OCRBench v2 extends text-rich scenarios with human-verified questions to strictly examine the optical character recognition and structure understanding capabilities of large multimodal models (Fu et al., 2025).

Based on the discussion above, combining these distinct roles reveals that synthetic vision and language data has evolved into a fundamental infrastructure for post-training. This data not only scales supervised instruction learning but also provides explicit signals for alignment. Furthermore, it enables the repair of weak cross-modal supervision and supports rigorous safety and structural evaluations within an ecosystem driven by LLMs.

7 Agent Data

Within the framework of digital twins and embodied artificial intelligence, we analyze agent data generated by LLMs from the perspective of the final data product utilized in practical applications. This approach is motivated by the perspective that world models function as internal simulators that capture environment dynamics and enable forward as well as counterfactual rollouts to support perception, prediction, and decision making (Li et al., 2025b).

We classify the practical data products used in digital twins and embodied artificial intelligence into three primary categories. The first category is environment and task data, which describes world setups and task-oriented scenario configurations (Ruan et al., 2025). The second category consists of control and decision data, which captures policies, action traces, and other control decisions for steering agents and simulations. This includes LLM multi-agent parametrization trajectories and sequential control plans in digital twin simulations (Xia et al., 2024). The third category is perception and telemetry data, which focuses on observed sensor streams and logs.

A key feature of this taxonomy is that it classifies data based on its primary use. For instance, mixed artifacts such as trajectory logs are assigned to a category based on whether they primarily serve to design testbeds, train agent behaviors, or monitor system health. Based on this data-product perspective, egocentric streams that are typical of embodied agents and exocentric telemetry common in digital twins can be integrated into a single taxonomy according to their main application. This classification remains compatible with the broader world model perspective that connects perception, dynamics, and control across different domains (Li et al., 2025b).

7.1 Generation Methods

Environment and task data. Environment and task data include the specifications required to initialize an episode. These specifications define the appearance of the environment, the entities or agents that are present, and the goals and constraints that constitute a valid simulation run.

In urban traffic simulation settings, LLMs parse free-form requests into structured keywords such as a dictionary of scenario fields. These keywords then drive the generation of scriptable configurations and parameters for urban mobility testing (Li et al., 2024c). Multimodal pipelines extend this concept to rare and difficult situations by generating realistic corner cases along with runnable tests that target the long tail of driving scenarios (Lu et al., 2024). Other complementary pipelines convert operational design domain descriptions into scripts compatible with ScenarioRunner for simulation-based testing (Danso & Büker, 2025).

Beyond traffic applications, semantic digital twins use LLMs to connect domain concepts to mission-level plans and recovery strategies. In these systems, the digital twin provides the domain concepts and interaction rules, while the LLMs generate structured action descriptions and recovery behaviors that can be executed and monitored within the twin (Naeem et al., 2025). At the enterprise level, work oriented toward Industry 5.0 proposes an Interactive-DT framework. In this framework, LLMs act as interactive interfaces and intelligent agents across the edge, digital twin, and service layers. This supports the construction and operation of digital twins, collaboration between cloud and edge systems, and advanced data analytics. This research also identifies integration challenges such as unreliability driven by hallucinations which may have safety impacts, as well as bias, inference speed constraints, interoperability, and secure deployment according to standards (Chen et al., 2025a).

In the area of embodied artificial intelligence, long-horizon planners such as L3M+P generate PDDL problems and maintain world-state graphs that can be instantiated into many concrete tasks and goals (Agarwal et al., 2025b). Safety-focused planners such as SELP translate natural language tasks into temporal logic specifications and use equivalence voting to improve the robustness and consistency of the mapping from natural language to logic. Following this, constrained planning enforces the resulting safety constraints during execution (Wu et al., 2025). Self-correcting frameworks such as the T3 Planner verify plans against spatio-temporal logic and repair them when needed. This process produces verified plan and trajectory traces that can be recorded as supervision for evaluation or learning under explicit constraints (Li & Zhao, 2025). Large-scale multi-agent benchmarks such as PARTNR also belong to this category. In these benchmarks, LLMs help design and decompose collaboration tasks, ground them into simulators, and export them as standard task collections together with planner baselines (Chang et al., 2024).

Control and decision data. Control and decision data describe how agents act once an environment and task have been fixed. In digital twin settings, LLM agents can operate in a closed loop with the simulator. These agents read simulation data through a data interface and output parameterized application programming interface or function calls, which are often serialized as JSON, through a control interface to adjust simulation parameters. They then iteratively summarize outcomes for the next cycle.

Running such agents yields cycle-level traces that include simulation data, control parameters, and agent summaries. These traces are recorded as simulation logs and can be compiled into a concise sequential control or parametrization plan for offline analysis and what-if evaluation (Xia et al., 2024). In safety-critical infrastructures such as power grids, control policies coordinated by LLMs are executed inside digital twin sandboxes where numerical solvers validate stability and safety. The resulting control sequences and trajectories support stress tests and control studies without affecting the real system (Zhang et al., 2025c).

For human-centered digital twins, persona simulations based on LLMs can act as silicon samples for digital experimentation. Academics may use them for pilot experiments to identify impactful stimuli, while firms may explore strategies for customer insight and product development (Toubia et al., 2025). Persona-based behavior-chain benchmarks such as BehaviorChain quantify the extent to which current LLMs can faithfully simulate continuous human behavior (Li et al., 2025a).

In embodied artificial intelligence, LLMs can also augment demonstration and trajectory data more directly. For instance, LLM Trainer performs offline demonstration annotation and online keypose retargeting to adapt demonstrations to new scenes and generate additional imitation trajectories with minimal human input (George & Farimani, 2025). Frameworks such as ELLMER integrate high-level language-driven task decomposition with low-level execution using vision or force feedback and retrieval-augmented code examples to solve long-horizon tasks. The resulting executions naturally produce rich interaction traces that can be recorded for analysis (Mon-Williams et al., 2025). Program-structured approaches such as Instruct2Act and ProgPrompt leverage executable or program-like representations to ground instructions in available actions as well as perception and planning loops in some systems. This enables a more modular analysis of which components are necessary (Huang et al., 2023; Singh et al., 2022). When execution logs contain both actions and rich observations, we treat them as control and decision data in this survey if their main use is to study or improve behavior rather than perception.

Perception and telemetry data. Perception and telemetry data focus on what is observed and recorded in digital twin and embodied environments, as well as on how these observations are generated and labeled. In digital twin pipelines for defect inspection, DefectTwin integrates a LLM driven multimodal pipeline to analyze multimodal inputs such as images. This system generates detailed defect descriptions and utilizes user interaction and feedback loops to support defect analysis and maintenance workflows (Ferdousi et al., 2024). In digital twins for autonomous driving based on camera and LiDAR sensors, a LLM interface enables online scenario editing through natural language prompts. These prompts allow for actions such as adding or removing assets, changing positions, or modifying appearances. This supports photorealistic and physically interactive simulations with sensor visualization and real-time interfacing with autonomous driving software stacks (Samak et al., 2025).

Similar workflows support egocentric data categories for embodied agents. LLM agents can produce Python scripts executable in Blender that retrieve and arrange 3D assets under spatial constraints derived from scene graphs. These agents render images and iteratively refine scenes through feedback from vision language models (Hu et al., 2024). LLMs can also be trained to generate Blender scripts for programmatic 3D object creation. The pipeline can render multiple views such as several images from different angles to increase visual diversity without manual modeling effort (Du et al., 2024). End-to-end pipelines based on mobile LiDAR enable rapid reconstruction of digital twin assets and can incorporate semantic enhancement guided by large vision language models into the reconstructed 3D representation. The resulting twins can be exported through formats such as OpenUSD for immersive inspection and downstream editing in platforms such as NVIDIA Omniverse (Gholizadeh HamlAbadi et al., 2025).

Across these examples, perception and telemetry data are primarily used for transfer learning, domain adaptation, and scalable evaluation using methods such as language-conditioned scoring or anomaly descriptions. In embodied setups, data consist of interaction traces in which observations are explicitly paired with agent actions, often following a partially observable Markov decision process formulation (Li et al., 2025b). In contrast, exocentric telemetry in digital twins is often structured for tracking past behaviors, monitoring current behaviors, and predicting future behaviors to support decision making as well as operations and maintenance (Deng et al., 2021; Bofill et al., 2023).

7.2 Quality Metrics for Agent Data

This section formalizes four metric families for agent data, which include validity, fidelity, diversity, and utility, and provides explicit definitions for each. We draw evaluation semantics from multi-agent embodied benchmarks such as PARTNR, text-to-traffic-scene evaluation on CARLA (Dosovitskiy et al., 2017), and world model or video assessments (Chang et al., 2024; Ruan et al., 2025; Li et al., 2025b). We consider trajectories defined as:

$$\tau = (s_0, a_0, r_1, s_1, \ldots, a_{T-1}, r_T, s_T)$$

and a dataset of generated trajectories denoted by:

$$\mathcal{A}_{\text{gen}} = \{\tau_i\}_{i=1}^{N}, \qquad \tau_i = (s_0^{(i)}, a_0^{(i)}, r_1^{(i)}, s_1^{(i)}, \ldots, a_{T_i-1}^{(i)}, r_{T_i}^{(i)}, s_{T_i}^{(i)})$$

In this formulation, s_t represents the state or observation at step t, a_t represents the action including tool or application programming interface calls, and r_{t+1} denotes the reward or feedback associated with the transition from (s_t, a_t) to s_{t+1}.

Validity. Validity measures whether generated agent data respect basic syntax, structural constraints, and executable preconditions. The action executability rate over all generated trajectories is defined as follows:

$$\text{ExecRate} = \frac{1}{\sum_{i=1}^{N} T_i} \sum_{i=1}^{N} \sum_{t=0}^{T_i-1} \mathbf{1}\Big\{\text{action } a_t^{(i)} \text{ executes without error in } s_t^{(i)}\Big\}.$$

Task success is a binary indicator of whether all goal predicates are satisfied at the end of an episode. Over N episodes, the success rate is calculated by the following formula:

$$\text{SR}_{\text{valid}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\text{all task goals satisfied in episode } i\}.$$

To capture partial completion, the percent complete metric reports the achieved fraction of goal predicates at termination. For a goal set G and an achieved subset $\hat{G}(\tau_i)$, the metric is defined as:

$$\text{PC}(\tau_i) = \frac{|\hat{G}(\tau_i)|}{|G|}.$$

The PARTNR benchmark defines task evaluation functions using propositions together with dependencies and constraints such as temporal constraints. It computes percent completion as the ratio of satisfied propositions. Success is achieved when the percent completion value equals 1.0 (Chang et al., 2024). In text-to-traffic-scene generation for CARLA, generation correctness is evaluated through text matching. This method uses a binary matched or unmatched criterion between the input prompt and the rendered scene, and the results are averaged over repeated generations (Ruan et al., 2025).

Fidelity. Fidelity measures the degree to which synthetic observations match reference distributions and structures. This metric is widely used in evaluations related to world models and video prediction (Li et al., 2025b).

The Frechet Inception Distance compares Gaussian fits of real and generated feature embeddings (Heusel et al., 2018). We let μ_X and Σ_X represent the mean and covariance of embeddings for real samples, $\mathrm{Tr}(\cdot)$ denotes the matrix trace and we let μ_Y and Σ_Y represent the mean and covariance for synthetic samples. The distance is calculated as follows:

$$\mathrm{FID} = \|\mu_X - \mu_Y\|_2^2 + \mathrm{Tr}\Big(\Sigma_X + \Sigma_Y - 2\big(\Sigma_X^{1/2}\Sigma_Y\Sigma_X^{1/2}\big)^{1/2}\Big).$$

The Frechet Video(FVD) applies the same mathematical form to video embeddings that capture temporal dynamics, where lower values indicate higher fidelity (Unterthiner et al., 2019).

Structural Similarity compares luminance, contrast, and structure between an image x and a reference image y (Wang et al., 2004). The metric is defined by the following equation:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}.$$

where C_1 and C_2 are small positive constants introduced to avoid numerical instability when the denominators are close to zero.

Mean Squared Error and Peak Signal-to-Noise Ratio are expressed as:

$$\mathrm{MSE} = \frac{1}{M}\sum_{i=1}^{M}(x_i - y_i)^2,$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right),$$

In these formulas, M represents the number of pixels or dimensions, and MAX denotes the maximum possible value of the signal.

Learned Perceptual Image Patch Similarity computes a perceptual distance between deep feature activations (Zhang et al., 2018). For layer-wise normalized features $\hat{f}^l_{h,w,x}$ and $\hat{f}^l_{h,w,y}$ as well as learned weights w_l, the distance is calculated as:

$$\mathrm{LPIPS}(x, y) = \sum_l \frac{1}{H_l W_l}\sum_{h,w}\Big\|w_l \odot \big(\hat{f}^l_{h,w,x} - \hat{f}^l_{h,w,y}\big)\Big\|_2^2.$$

where the symbol $\odot$ denotes the Hadamard element wise product. This operation represents channel wise scaling performed by the learned weight vector w_l.

The evaluation metrics mentioned above assume either distributional access to real samples, as in the cases of Frechet Inception Distance and Frechet Video Distance, or the availability of paired references such as Structural Similarity, Mean Squared Error, Peak Signal-to-Noise Ratio, and Learned Perceptual Image Patch Similarity. However, in many conditional generation settings, paired pixel-level ground truth is not available. In these situations, we utilize semantic prompt and outcome alignment as a reference-free proxy.

When pixel-level ground truth is unavailable, we use prompt–outcome semantic alignment as a reference-free fidelity proxy that is widely applicable when pixel-level ground truth is unavailable is prompt and outcome semantic alignment. This metric measures whether a synthesized outcome is consistent with the conditioning prompt at the level of high-level semantics. Given a textual prompt or specification p and a generated outcome o, such as a rendered image or keyframe produced from a generated scenario, we obtain cross-modal embeddings using a pretrained vision and language model such as CLIP (Radford et al., 2021).

We let $f_T(\cdot)$ and $f_V(\cdot)$ denote the text and vision encoders, and we let

$$e_p = f_T(p), \qquad e_o = f_V(o)$$

be the resulting embeddings. We define the prompt and outcome semantic alignment score by cosine similarity:

$$\mathrm{SemAlign}(p, o) = \frac{e_p^\top e_o}{\|e_p\|_2 \, \|e_o\|_2},$$

where $\| \cdot \|_2$ is the ℓ_2 norm and higher values indicate stronger semantic agreement. Over N prompt and outcome pairs $\{(p_i, o_i)\}_{i=1}^N$, the mean semantic fidelity is calculated as:

$$\mathrm{SemFid} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{SemAlign}(p_i, o_i).$$

The mean semantic fidelity summarizes the semantic consistency between conditions and synthesized outcomes without requiring a reference image for each prompt. This approach is closely related to CLIP-based and reference-free evaluation metrics used in vision and language generation (Hessel et al., 2022). When a discrete correctness signal is preferred, such as a matched or unmatched criterion, one may threshold the semantic alignment score and report the resulting match rate:

$$\mathrm{MatchRate}(\tau) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\mathrm{SemAlign}(p_i, o_i) \geq \tau\},$$

where τ is a similarity threshold and $\mathbf{1}\{\cdot\}$ is the indicator function.

Diversity. Diversity captures the coverage of modes, long-tail concepts, and non-templated variety within the generated data. In the context of text-to-traffic-scene generation, diversity can be quantified using agent diversity(AD) and road diversity(RD). Both of these metrics are computed as the ratio of unique elements to total elements across repeated generations. Specifically, agent diversity accounts for variations in agent type, action, and relative position, while road diversity is calculated based on unique road identifiers (Ruan et al., 2025).

For observation-level evaluation of generated videos or world model rollouts, the VBench framework decomposes video generation assessment into complementary axes such as subject consistency and temporal Quality. An example of temporal Quality is motion smoothness. This framework also separately evaluates the consistency between the video and the condition, which refers to how well the output adheres to the given conditions or prompts. These axis-wise scores provide fine-grained evidence regarding strengths and weaknesses across the prompt and condition space (Huang et al., 2024b; Li et al., 2025b).

Utility. Utility measures downstream effectiveness during training or evaluation with synthetic agent data. The success rate over N evaluation episodes with binary success indicators s_i taking values of zero or one is defined by the following equation:

$$\mathrm{SR}_{\mathrm{eval}} = \frac{1}{N} \sum_{i=1}^{N} s_i.$$

To quantify efficiency, we define L_i as the number of environment steps used in episode i:

$$\mathrm{SimSteps} = \frac{1}{N} \sum_{i=1}^{N} L_i.$$

In collaborative settings, task offloading measures the division of labor. We let $n_i^{(r)}$ be the number of sub-tasks or propositions completed by the robot in episode i and n_i be the total number of completed sub-tasks. The metric is expressed as:

$$\mathrm{Offloading} = \frac{1}{N} \sum_{i=1}^{N} \frac{n_i^{(r)}}{n_i}.$$

For partial task completion at the episode level, we reuse the percent complete metric:

$$\mathrm{PC}(\tau_i) = \frac{|\hat{G}(\tau_i)|}{|G|}.$$

In SafeBench-style CARLA driving evaluation, the collision rate (CR) is defined as the expected collision indicator (or count) over evaluation scenarios τ:

$$\mathrm{CR} = \mathbb{E}_{\tau \sim P}[c(\tau)]\,,$$

where $c(\tau)$ denotes the collision signal in scenario τ. In many episodic driving setups where an episode terminates upon the first collision (or collisions are binarized), $c(\tau) \in \{0, 1\}$ and CR can be interpreted as the proportion of episodes that contain at least one collision. Lower values indicate better safety performance (Xu et al., 2022; Zhang et al., 2024b).The overall score is a composite metric that aggregates driving metrics related to safety, functionality, and etiquette by using weights specified by the benchmark. In practice, the overall score is commonly interpreted as a measure of ego-vehicle driving performance where higher values are preferred. Conversely, adversarial scenario generation or selection may instead aim to reduce the overall score by constructing more challenging conditions.

Reinforcement learning studies also report the discounted return per episode:

$$G_t = \sum_{k=0}^{\infty} \gamma^k \, r_{t+k+1},$$

This return, as well as the average return across episodes, is used to assess whether synthetic data improves policy performance or sample efficiency.

7.3 Trustworthy Metrics for Agent Data

In practice, the evaluation of agent data generated by LLMs has primarily focused on safety. This focus examines whether trajectories, scenarios, and plans avoid dangerous states and behaviors that break established rules. Other dimensions of trustworthiness, such as fairness and privacy, are significantly less developed in current benchmarks for embodied agents and digital twins. Therefore, we treat these aspects as open directions for future research and focus our current discussion on safety metrics.

Safety. Safety measures whether trajectories, scenarios, and plans generated by LLMs avoid hazardous states and rule-breaking behaviors and satisfy explicit safety constraints, independent of task completion. In practice, safety is often assessed through infraction and collision metrics defined by simulators in driving benchmarks, such as suites based on CARLA like SafeBench. It is also evaluated through the satisfaction of formal specifications in pipelines for constraint-enforced planning and motion planning (Xu et al., 2022; Wu et al., 2025; Li & Zhao, 2025). Systems for text-to-scenario generation further emphasize the production of standardized scenario files and simulator testing reports that include monitored evaluation indicators (Cai et al., 2025).

Rule violations include basic traffic infractions such as running red lights, driving the wrong way, lane-keeping violations, and speeding. A generic per-episode violation rate can be calculated as:

$$\mathrm{RVR} = \frac{1}{N} \sum_{i=1}^{N} \frac{V_i}{E_i},$$

where V_i represents the total number of rule violations in episode i, and E_i represents an exposure term such as the number of decision steps T_i or the distance traveled.

In practice, CARLA-based benchmarks frequently report specific categories of violation metrics such as collisions, red-light running, stop-sign running, out-of-road distance, and lane invasion rather than a single aggregated count (Xu et al., 2022). An exposure-normalized violation rate for a specific infraction type t can be defined as follows:

$$\mathrm{RVR}^{(t)} = \frac{\sum_{i=1}^{N} V_i^{(t)}}{\sum_{i=1}^{N} \mathrm{dist}_i},$$

where $V_i^{(t)}$ is the number of infractions of type t in episode i. This style of reporting is also used in CARLA Leaderboard results where multiple infraction types are shown as infractions per kilometer (CARLA Autonomous Driving Leaderboard, 2025b). Fine-grained violation counts for each specific type additionally enable a more detailed diagnosis than aggregate success or return (Cai et al., 2025; Xu et al., 2022).

Route incompleteness measures the extent to which the planned route remains unfinished at the end of a scenario:

$$\mathrm{RI} = 1 - \frac{\text{distance completed}}{\text{planned route length}}.$$

Higher values for this metric indicate early termination or a failure to follow the designated route. In CARLA-style evaluations, route progress or completion is commonly reported together with violation and collision metrics. This approach helps to distinguish safe task completion from instances of early stopping or unsafe driving (Xu et al., 2022).

Speed-related compliance and infractions capture whether an agent drives too slowly or too fast relative to legal or context-dependent bounds. A simple minimum-speed compliance rate is defined as follows:

$$\mathrm{MSCR} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(v(t) \geq v_{\min}(t)),$$

where $v_{\min}(t)$ is a context-dependent minimum-speed requirement such as a speed determined by nearby traffic. The CARLA Leaderboard explicitly penalizes the failure to maintain minimum speed as part of its evaluation criteria (CARLA Autonomous Driving Leaderboard, 2025a). More generally, one may also track speed-limit compliance within specific bounds when such information is available in the simulator or map.

Comfort and smoothness often serve as indicators for aggressive or risky maneuvers. CARLA-based diagnostic reports may include kinematics-based indicators such as average acceleration and yaw velocity (Xu et al., 2022). For example, if $a(t)$ represents the acceleration and $\omega(t)$ represents the yaw velocity, these can be quantified as follows:

$$\begin{aligned} \mathrm{ACC} &= \mathbb{E}_{\tau \sim P}[\mathrm{acc}(\tau)], \\ \mathrm{YV} &= \mathbb{E}_{\tau \sim P}[y(\tau)]. \end{aligned}$$

Furthermore, more sensitive smoothness indicators can be computed, such as root-mean-square jerk and the hard-braking rate:

$$\mathrm{Jerk}_{\mathrm{RMS}} = \sqrt{\frac{1}{T-2} \sum_{t=2}^{T-1} \left\| \frac{a(t+1) - a(t-1)}{2\Delta t} \right\|^2},$$

$$\mathrm{HardBrakeRate} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(a_x(t) < -\tau),$$

where $a_x(t)$ is longitudinal acceleration and τ is a braking threshold. These smoothness metrics help to characterize the trade-off between safety and efficiency while complementing collision and completion metrics.

Formal safety satisfaction measures the fraction of executions that satisfy a temporal-logic specification ϕ, such as those defined by Linear Temporal Logic (LTL) or Signal Temporal Logic (STL). If $\sigma^{(i)}$ represents the execution trace for episode i, then the safety satisfaction rate is defined as follows:

$$\mathrm{SafetySat} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\sigma^{(i)} \models \phi).$$

The SELP framework improves the safety rate by mapping natural language to Linear Temporal Logic and enforcing constrained decoding, while the T^3 Planner utilizes Signal Temporal Logic verification in the loop to increase the satisfaction rates of motion plans (Wu et al., 2025; Li & Zhao, 2025).

Hazard rejection and risk are safety-specific metrics for embodied LLM agents that capture how these agents respond to explicitly hazardous tasks. Given a set of labeled hazardous tasks $\mathcal{H}$ and a set of safe tasks $\mathcal{S}$, the hazard rejection rate and risk rate are defined as follows:

$$\text{Rejection} = \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \mathbb{I}(\text{agent refuses } h),$$
$$\text{Risk} = \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \mathbb{I}(\text{agent executes } h).$$

SafeAgentBench reports a low rejection rate and non-trivial risk for current embodied LLM agents, which motivates the explicit tracking of these quantities (Yin et al., 2025).

Time-to-Collision is a widely used proximity-based surrogate safety indicator in traffic safety assessment. Given the relative longitudinal distance $d(t)$ and closing speed $\dot{d}(t) < 0$, the instantaneous Time-to-Collision and its minimum over an episode are defined as follows:

$$\text{TTC}(t) = \begin{cases} \dfrac{d(t)}{-\dot{d}(t)}, & \dot{d}(t) < 0, \\ +\infty, & \dot{d}(t) \geq 0, \end{cases}$$
$$\text{minTTC} = \min_{t} \text{TTC}(t).$$

Lower values for the minimum Time-to-Collision indicate a higher risk of collision (Ward et al., 2015; Sharath & Mehran, 2021).

Minimum Distance to Collision is another commonly used proximity-based indicator. It is defined by using the positions of the ego agent $\mathbf{p}_{\text{ego}}(t)$ and the other agent $\mathbf{p}_{\text{other}}(t)$ as follows:

$$\text{MDC} = \min_{t} \left\| \mathbf{p}_{\text{ego}}(t) - \mathbf{p}_{\text{other}}(t) \right\| .$$

Lower values for the Minimum Distance to Collision indicate higher collision risk (Gao et al., 2025).

These proximity measures based on Time-to-Collision and Minimum Distance to Collision have also been adopted when evaluating traffic scenarios generated or selected with the aid of LLMs (Gao et al., 2025).

Violation diagnosis accuracy evaluates the reliability of automated detectors, including auditors based on multimodal LLMs, when used to identify safety violations or accidents from logs, images, or narratives. This accuracy is typically summarized by standard classification metrics such as precision, recall, and the F1 score (Skender et al., 2025). Additionally, the SeeUnsafe framework proposes a multimodal LLM agent for traffic accident analysis and introduces an Information Matching Score to align structured model responses with ground truth data (Zhang et al., 2025b).

7.4 Evaluation Practice Gap

We analyze representative methods from Section 7.1 and categorize their reported evaluation protocols in Table 7.

Diversity. Regarding diversity, the scope of evaluation in current agent data generation remains limited. Our analysis shows that only generation from text to traffic scenes explicitly reports diversity metrics. In contrast, other representative methods do not provide quantitative measures of coverage or variation that is not based on templates under repeated sampling. Consequently, although these methods allow for comparisons of success rates or efficiency, it remains difficult to determine from existing experiments whether the generated data meaningfully expands the coverage of behavior modes, tasks, or scenarios.

Generation method	Validity	Fidelity	Diversity	Utility	Safety
TTSG(Ruan et al., 2025)	✓	△	✓	✓	✓
PARTNR (Chang et al., 2024)	✓	×	×	✓	×
SELP (Wu et al., 2025)	✓	×	×	✓	✓
Grid-Agent (Zhang et al., 2025c)	✓	×	×	✓	△
T^3 Planner (Li & Zhao, 2025)	✓	△	×	✓	✓

Table 7: Whether representative agent-data generation methods explicitly evaluate each dimension in their experimental sections. ✓: explicitly evaluated; △: partially/indirectly covered; ×: not reported or not applicable.

Fidelity. Regarding distribution level fidelity, evaluations are mostly focused on proxies for consistency and correctness instead of direct tests for distributional alignment. In our representative set, the generation from text to traffic scenes reports prompt scene matching as a consistency proxy. In contrast, methods centered on planning and control focus mainly on success rates, completion, and constraint satisfaction. These methods include SELP(Wu et al., 2025), PARTNR(Chang et al., 2024), and Grid Agent(Zhang et al., 2025c). These studies generally do not report metrics that compare synthetic trajectories or plans to reference data distributions. As a result, while consistency proxies are partially covered, the statistical proximity of synthetic agent data to reference distributions is still mostly unvalidated.

Diagnostic Safety. Concerning the diagnosis of safety issues, current reporting relies heavily on broad benchmark metrics such as those related to collisions in evaluations using the CARLA simulator. It also relies on the implicit satisfaction of specifications such as adherence to linear temporal logic or signal temporal logic. These methods are used instead of detailed and specific diagnostics for different types of violations. In our representative set, the generation of traffic scenes reports signals related to collisions. Meanwhile, the SELP(Wu et al., 2025) and T^3 Planner (Li & Zhao, 2025) methods mainly focus on the satisfaction of formal constraints. In contrast, other approaches mainly report success at the task level. This includes the completion of tasks or the resolution of violations. However, they do not provide an independent breakdown of safety violations for each specific type. As a result, although current evaluations can show whether safety constraints are met, they provide limited diagnostic detail for identifying and explaining specific modes of safety failure.

7.5 Usages

Having discussed the generation of agent data, we now focus on its practical applications. This section examines how trajectories, task specifications, and telemetry streams are used to train, evaluate, and refine agents based on LLMs in both embodied and digital twin environments. We follow the same three-part structure established previously by categorizing these applications into environment and task data for testbeds, control and decision data for supervision, and perception and telemetry data for evaluation signals.

Environment and task data usages. Environment and task data primarily serve as a rigorous testbed for reasoning, planning, and safety verification. In the context of lifelong planning, maintaining a persistent memory of the world through world-state graphs is crucial. This approach enables agents to repeatedly instantiate and solve PDDL planning problems from natural language tasks as the environment evolves, which supports the repeatable evaluation of long-horizon symbolic planning (Agarwal et al., 2025b).

When safety is paramount, these data specifications act as strict constraints. Safety-focused pipelines convert natural language instructions into formal temporal logic, such as Linear Temporal Logic, and then utilize constrained decoding guided by automata to ensure that generated plans adhere to these formal constraints step by step (Wu et al., 2025).

Beyond static enforcement, this data also drives self-correction. Systems often pair planners based on LLMs with logic-based verifiers that iteratively reject and repair unsafe actions (Li & Zhao, 2025). This iterative loop can be logged to form trial-and-error traces consisting of failed plans, verifier diagnostics such as constraint violations or robustness scores, and corrected revisions. These traces are useful for evaluation and may be repurposed for future training.

For multi-agent systems, collaboration benchmarks provide grounded and simulator-verifiable tasks. These datasets effectively transform individual environments into collaborative evaluation suites designed to test how well agents coordinate, track task progress, and recover from errors through language (Chang et al., 2024). Similarly, in digital twin settings, formal task descriptions define the distinct operational boundaries for system controllers. These specifications are used to stress-test governance policies and verify that fallback strategies function correctly under diverse conditions.

Control and decision data usages. Control and decision data provide direct supervision for language-driven policies and are reused in both imitation and feedback-based learning. On the demonstration side, recent frameworks ground decisions from LLMs in collections of trajectories. For example, combining language understanding with value estimates learned from robot interaction data allows a model to choose actions that are consistent with successful past behavior (Ahn et al., 2022). Programmatic control follows a related pattern where natural language is translated into executable policies that can be directly run on robotic platforms (Liang et al., 2023). More generally, executing such policies naturally produces traces including states, actions, and failures that could be repurposed for analysis, dataset construction, or distillation into smaller controllers. Large and heterogeneous robot logs have been pooled to train instruction-following visuomotor policies that generalize across different robots, which enables analyses of how dataset diversity and composition affect language-conditioned generalization (Collaboration et al., 2025).

The same data also drive reinforcement-style updates that are mediated by language. Automated reward design utilizes LLMs to write and refine reward functions, which are frequently formatted as executable code. These functions are validated through downstream policy optimization and evaluation in the target environment, which is typically performed in simulation to enable rapid iteration over reward hypotheses (Ma et al., 2024a; Xie et al., 2024).

In offline settings, policies can be conditioned on language embeddings derived from LLMs and trained via offline reinforcement learning on static logs. This approach enables generalization to unseen commands without the need for additional interaction (Morad et al., 2024). To reduce the effort required for human labeling, artificial intelligence feedback protocols use multimodal critics to score trajectories after they occur. This process converts archived trials into training data for behavior shaping. For example, recent work adapts video and language models into language-conditioned robotic reward functions that score executions directly from video (Yang et al., 2024c). Complementary to critic-based feedback, methods for one-video reward inference can compute dense rewards from a single demonstration video using methods such as semantic point correspondence and automated point tracking. This allows for trajectory evaluation and dataset filtering for downstream policy synthesis (Shi et al., 2025a).

In digital twins, control logs from simulation play a similar role. These logs can be relabeled, scored, and reused to refine control policies and to analyze how controllers driven by LLMs behave under distribution shifts or rare events.

Perception and telemetry data usage. Perception and telemetry data focus less on direct control and more on evaluation and reward learning. Streams aligned with actions, such as egocentric or first-person observations from cameras and proprioception, provide raw material for judging behavior and system state. Similarly, exocentric system events, including third-person logs from external cameras, map-based states, and simulator records, serve a similar purpose. Methods that utilize LLMs as judges evaluate diverse artifacts and multimodal inputs along multiple criteria. The reliability of these assessments is supported by careful design of prompts and inputs as well as strategies for consistency and bias mitigation (Gu et al., 2025; Li et al., 2024a).

Vision and language reward learning can adapt video and language models into language-conditioned reward functions using successful and failed execution videos. This process produces reward scores and signals that can guide planning or reinforcement learning (Yang et al., 2024c). Furthermore, LLMs can propose parameterized reward features and iteratively refine reward parameters using execution feedback. This refinement is achieved by minimizing ranking inconsistency between the model and the learned reward function (Zeng et al., 2024). In the domain of driving evaluation, frameworks based on LLMs can transform multi-source driving logs, such as surround-view videos and CAN bus signals, into structured driving contexts. These systems then output structured assessments of driving behaviors across safety, intelligence, and comfort, which are validated in simulation (You et al., 2025).

Large and multi-domain observation corpora also support 4D world modeling, including geometric understanding and camera-conditioned video generation. These corpora provide temporally aligned video streams with rich multimodal signals such as depth maps, camera poses, optical flow, and foreground masks. Such data enables models to learn scene dynamics and generate videos that adhere to specified camera trajectories. OmniWorld presents a unified resource for 4D world modeling that spans reconstruction and future-oriented prediction needs. It introduces a benchmark centered on 3D geometric prediction and camera-controlled video generation, which is built from a newly collected game subset called OmniWorld-Game and curated public datasets across diverse domains (Zhou et al., 2025).

8 Open Challenges and Future Directions

While this survey establishes a unified taxonomy for evaluating data generation driven by LLMs across modalities, the field is rapidly transitioning from static dataset creation to dynamic and self-improving synthesis ecosystems. Current metrics were largely designed for the era of fixed human-annotated corpora and they face significant challenges in this new regime. Building on our established framework of quality and trustworthiness, we highlight directions where methodological advancement is required to ensure the sustainability and reliability of synthetic data.

8.1 From Static Snapshots to Dynamic Feedback-Loop Evaluation

Most metrics reviewed in this survey, such as diversity measured through self-similarity or fidelity measured through distributional distance, provide only a static snapshot of a single generation round. However, practical applications are increasingly moving toward recursive training loops where synthetic data are used to train models that subsequently generate new data.

The main challenge is that these static snapshots fail to capture long-term system dynamics. A dataset may achieve high scores for diversity in the first iteration yet still drive model collapse. This is a degenerative process where distributional tails disappear and modes are over-amplified after several training and generation cycles.

A key future direction is to move from point-wise evaluation to longitudinal trajectory monitoring. There is a need for dynamic metrics that track the derivatives of quality over time. Such metrics would detect the loss of support coverage, the contraction of the feature space, or the drift of error patterns across iterations. These temporal metrics could serve as early warning signals that trigger interventions, such as mixing in real data, rebalancing domains, or adjusting sampling strategies, before model collapse becomes irreversible. More broadly, this shift requires meta-evaluation protocols to test whether existing metrics are sufficiently sensitive to serve as stability controllers in feedback loops.

8.2 Redefining Fidelity: From Surface Mimicry to Process Verifiability

In our taxonomy, fidelity has traditionally measured how closely generated data match the distribution of real world human data. For open-ended natural language tasks such as dialogue, this human-centric distributional similarity remains central. However, for reasoning-intensive domains, this definition is becoming a bottleneck. As reasoning-oriented LLMs begin to match or surpass average human performance, enforcing strict adherence to human distributions may penalize correct but novel solutions. In fields such as mathematics, programming, or symbolic logic, sounding human is less important than being objectively correct.

For such modalities, fidelity metrics must evolve from measuring mimicry, which is the distributional similarity to human artifacts, toward measuring verifiability. Promising directions include scalable and automated process-level rewards. These rewards may include execution feedback for code, formal proof checkers for mathematics, or logical consistency probes for chain-of-thought traces. The goal is to assess the correctness and internal coherence of the reasoning process rather than its stylistic resemblance to human baselines.

8.3 Navigating the Trust and Utility Pareto Frontier

Our framework distinguishes between Quality, which includes downstream utility, and Trustworthiness, which covers privacy, safety, and fairness. While these are often treated as separate pillars in a taxonomy, they frequently act as competing objectives during deployment.

Mechanisms designed to increase trustworthiness often impose an alignment tax on synthetic corpora. For instance, aggressive safety filtering may not only remove harmful content but also disproportionately eliminate rare concepts, which effectively shrinks tail coverage and reduces diversity. Similarly, adding differential privacy noise to tabular data can improve privacy guarantees but degrade predictive performance. Strict alignment in conversational agents may also reduce harmful outputs at the cost of elevated refusal rates on benign queries, which is a phenomenon known as over-refusal.

Future work should move beyond optimizing individual metrics in isolation and instead quantify and navigate these trade-offs. Concretely, we need frameworks that characterize the Pareto optimal frontier over pipeline configurations. This is the set of operating points where no trust metric can be improved without degrading utility or vice versa. This would enable explicit and application dependent choices, such as estimating how much downstream performance must be sacrificed to attain a target privacy guarantee defined by ϵ and δ, or how much refusal rate slack is acceptable to achieve a desired safety level. More broadly, this calls for multi-objective optimization and selection strategies that treat trust and utility as co-dependent variables rather than independent checkboxes.

9 Conclusion

Current research efforts are predominantly directed toward leveraging Large Language Models (LLMs) for data generation, while the critical role of the "**Data Auditor**"—responsible for evaluating the quality of synthetic data—has been relatively overlooked. Ensuring high data quality is essential for transforming scarce data into a controllable resource, suitable not only for model training but also for direct real-world applications.

In this survey, centered on synthetic data, we aim to bridge fragmented progress and provide a systematic understanding of evaluation methods through our proposed LLM Data Auditor framework. Beyond summarizing typical generation methods across various modalities, we categorize representative metrics into two primary dimensions: Quality and Trustworthiness. By applying this evaluation system, we identify significant gaps in current assessment practices. For instance, fairness evaluation in tabular data generation remains notably underdeveloped. Consequently, our framework serves as both a comprehensive reference and a diagnostic tool to pinpoint missing evaluation dimensions across different data modalities.

Our findings suggest that, regardless of the modality, there is an urgent need to refine post-generation evaluation metrics. A holistic assessment is necessary to prevent synthetic data from excelling in one dimension while failing in others. Furthermore, we highlight inherent trade-offs between certain metrics, such as the tension between privacy and fidelity in tabular data, which further underscores the necessity of a multi-dimensional evaluation system.

For future work, the evaluation framework proposed in this survey can serve as a foundation for developing comprehensive benchmarks to assess existing generation methods. Additionally, exploring the transition from static to dynamic evaluation systems will be crucial for the continued advancement of data generation. While we acknowledge that our collection of metrics may not be exhaustive, the Quality-Trustworthiness framework is designed to be open and extensible, allowing for the integration of new and valuable metrics as the field evolves.

We hope this survey serves as a foundation for high-quality and trustworthy LLM-based data generation, enabling the community to develop more robust and reliable data generation systems.

References

Oluwanifemi Adebayo Moses Adekanye. LLM-powered synthetic environments for self-driving scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 23721–23723, 2024. doi: 10.1609/aaai.v38i21.30540. URL https://ojs.aaai.org/index.php/AAAI/article/view/30540.

Harshvardhan Aditya, Siddansh Chawla, Gunika Dhingra, Parijat Rai, Saumil Sood, Tanmay Singh, Zeba Mohsin Wase, Arshdeep Bahga, and Vijay K. Madisetti. Evaluating privacy leakage and memorization attacks on large language models (llms) in generative ai applications. *Journal of Software Engineering and Applications*, 17(5):421–447, 2024. doi: 10.4236/jsea.2024.175023. URL https://www.scirp.org/journal/paperinformation?paperid=133625.

Bhavik Agarwal, Ishan Joshi, and Viktoria Rojkova. Think inside the json: Reinforcement strategy for strict llm schema adherence, 2025a. URL https://arxiv.org/abs/2502.14905.

Krish Agarwal, Yuqian Jiang, Jiaheng Hu, Bo Liu, and Peter Stone. L3m+p: Lifelong planning with large language models, 2025b. URL https://arxiv.org/abs/2508.01917.

Roozbeh Aghili, Heng Li, and Foutse Khomh. Protecting privacy in software logs: What should be anonymized?, 2025. URL https://arxiv.org/abs/2409.11313.

Wasi Uddin Ahmad, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Vahid Noroozi, Somshubra Majumdar, and Boris Ginsburg. Opencodeinstruct: A large-scale instruction tuning dataset for code llms, 2025. URL https://arxiv.org/abs/2504.04030.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022. URL https://say-can.github.io/.

Gretel AI. Synthetically generated reasoning dataset (gsm8k-inspired) with enhanced diversity using gretel navigator and meta-llama/meta-llama-3.1-405b, 9 2024. URL https://huggingface.co/datasets/gretelai/gretel-math-gsm8k-v1.

Ahmed M. Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 290–306. PMLR, 2022. URL https://proceedings.mlr.press/v162/alaa22a.html.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models, 2024. URL https://arxiv.org/abs/2402.16827.

Anthropic. Enhancing model safety through pretraining data filtering, 2025. URL https://alignment.anthropic.com/2025/pretraining-data-filtering/.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://arxiv.org/abs/2204.05862.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023. URL https://fairmlbook.org/.

Austin A. Barr, Joshua Quan, Eddie Guo, and Emre Sezgin. Large language models generating synthetic clinical datasets: A feasibility and comparative analysis with real-world perioperative data. *Frontiers in Artificial Intelligence*, 8:1533508, 2025. doi: 10.3389/frai.2025.1533508. URL https://www.frontiersin.org/articles/10.3389/frai.2025.1533508.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Guiding llms the right way: Fast, non-invasive constrained generation, 2024. URL https://arxiv.org/abs/2403.06988.

Jherson Bofill, Mideth Abisado, Jocelyn Villaverde, and Gabriel Avelino Sampedro. Exploring digital twin-based fault monitoring: Challenges and opportunities. *Sensors*, 23(16), 2023. ISSN 1424-8220. doi: 10.3390/s23167087. URL https://www.mdpi.com/1424-8220/23/16/7087.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators, 2023. URL https://arxiv.org/abs/2210.06280.

Jack Boylan, Shashank Mangla, Dominic Thorn, Demian Gholipour Ghalandari, Parsa Ghaffari, and Chris Hokamp. Kgvalidator: A framework for automatic validation of knowledge graph construction, 2024. URL https://arxiv.org/abs/2404.15923.

Tim Bray. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259, December 2017. URL https://www.rfc-editor.org/info/rfc8259.

Xuan Cai, Xuesong Bai, Zhiyong Cui, Danmu Xie, Daocheng Fu, Haiyang Yu, and Yilong Ren. Text2scenario: Text-driven scenario generation for autonomous driving test. *arXiv preprint arXiv:2503.02911*, 2025. URL https://arxiv.org/abs/2503.02911.

CARLA Autonomous Driving Leaderboard. Evaluation criteria for the leaderboard 2.0, 2025a. URL https://leaderboard.carla.org/evaluation_v2_0/.

CARLA Autonomous Driving Leaderboard. Carla autonomous driving leaderboard (leaderboard table), 2025b. URL https://leaderboard.carla.org/leaderboard/.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium*, pp. 2633–2650, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavier Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John M. Turner, Eric Undersander, and Tsung-Yen Yang. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks, 2024. URL https://arxiv.org/abs/2411.00081.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation, 2023. URL https://github.com/sahil280114/codealpaca.

Bocheng Chen, Guangjing Wang, Hanqing Guo, Yuanda Wang, and Qiben Yan. Understanding multi-turn toxic behaviors in open-domain chatbots. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID 2023, pp. 282–296. ACM, October 2023a. doi: 10.1145/3607199.3607237. URL http://dx.doi.org/10.1145/3607199.3607237.

Chong Chen, Kuanhong Zhao, Jiewu Leng, Chao Liu, Junming Fan, and Pai Zheng. Integrating large language model and digital twins in the context of industry 5.0: Framework, challenges and opportunities. *Robotics and Computer-Integrated Manufacturing*, 94:102982, 2025a. ISSN 0736-5845. doi: https://doi.org/10.1016/j.rcim.2025.102982. URL https://www.sciencedirect.com/science/article/pii/S0736584525000365.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023b. URL https://arxiv.org/abs/2311.12793.

Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, Li Yuan, Yu Qiao, Dahua Lin, Feng Zhao, and Jiaqi Wang. Sharegpt4video: Improving video understanding and generation with better captions, 2024a. URL https://arxiv.org/abs/2406.04325.

Wei Chen, Lin Li, Yongqi Yang, Bin Wen, Fan Yang, Tingting Gao, Yu Wu, and Long Chen. Comm: A coherent interleaved image-text dataset for multimodal understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Chen_CoMM_A_Coherent_Interleaved_Image-Text_Dataset_for_Multimodal_Understanding_and_CVPR_2025_paper.pdf.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models, 2024b. URL https://arxiv.org/abs/2401.01335.

Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation, 2024. URL https://arxiv.org/abs/2407.06135.

Coalition for Content Provenance and Authenticity. Content credentials: C2PA technical specification, version 2.2. Specification, May 2025. URL https://c2pa.org/specifications/specifications/2.2/specs/C2PA_Specification.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: boosting language models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024. URL https://dl.acm.org/doi/abs/10.5555/3692070.3692454.

Aaron Agyapong Danso and Ulrich Büker. Automated generation of test scenarios for autonomous driving using llms. *Electronics*, 14(16), 2025. ISSN 2079-9292. doi: 10.3390/electronics14163177. URL https://www.mdpi.com/2079-9292/14/16/3177.

DataCebo. Detection: Single table, 2024. URL https://docs.sdv.dev/sdmetrics/data-metrics/metrics-in-beta/detection-single-table.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. doi: 10.48550/arXiv.2501.12948. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

Tianhu Deng, Keren Zhang, and Zuo-Jun (Max) Shen. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *Journal of Management Science and Engineering*, 6(2):125–134, 2021. ISSN 2096-2320. doi: https://doi.org/10.1016/j.jmse.2021.03.003. URL https://www.sciencedirect.com/science/article/pii/S2096232021000238.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation, 2023. URL https://arxiv.org/abs/2311.01460.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.212/.

Chuanghao Ding, Xuejing Liu, Wei Tang, Juan Li, Xiaoliang Wang, Rui Zhao, Cam-Tu Nguyen, and Fei Tan. Synthdoc: Bilingual documents synthesis for visual document understanding, 2024. URL https://arxiv.org/abs/2408.14764.

Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11165–11197. PMLR, 2024. URL https://proceedings.mlr.press/v235/dohmatob24b.html.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. URL https://arxiv.org/abs/1711.03938.

dottxt-ai. outlines-core, 2025. URL https://github.com/dottxt-ai/outlines-core.

Enjun Du, Xunkai Li, Tian Jin, Zhihan Zhang, Rong-Hua Li, and Guoren Wang. Graphmaster: Automated graph synthesis via llm agents in data-limited environments, 2025. URL https://arxiv.org/abs/2504.00711.

Yuhao Du, Shunian Chen, Wenbo Zan, Peizhao Li, Mingxuan Wang, Dingjie Song, Bo Li, Yan Hu, and Benyou Wang. Blenderllm: Training large language models for computer-aided design with self-improvement, 2024. URL https://arxiv.org/abs/2412.14203.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy, data processing inequalities, and statistical minimax rates, 2014. URL https://arxiv.org/abs/1302.3203.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Liancheng Fang, Aiwei Liu, Hengrui Zhang, Henry Peng Zou, Weizhi Zhang, and Philip S. Yu. Tabgen-icl: Residual-aware in-context example selection for tabular data generation, 2025. URL https://arxiv.org/abs/2502.16414.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models(llms) on tabular data: Prediction, generation, and understanding – a survey, 2024. URL https://arxiv.org/abs/2402.17944.

Michael Feldman, Sorelle Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact, 2015. URL https://arxiv.org/abs/1412.3756.

Xiaohan Feng, Xixin Wu, and Helen Meng. Ontology-grounded automatic knowledge graph construction by llm under wikidata schema. In *Proceedings of the HI-AI@KDD Workshop*, pp. 117–135. CEUR Workshop Proceedings, 2024. URL https://arxiv.org/abs/2412.20942.

Yukang Feng, Jianwen Sun, Chuanhao Li, Zizhen Li, Jiaxin Ai, Fanrui Zhang, Yifan Chang, Sizhuo Zhou, Shenglin Zhang, Yu Dai, and Kaipeng Zhang. A high-quality dataset and reliable evaluation for interleaved image-text generation, 2025a. URL https://arxiv.org/abs/2506.09427.

Yushi Feng, Tsai Hor Chan, Guosheng Yin, and Lequan Yu. Democratizing large language model-based graph data augmentation via latent knowledge graphs, 2025b. URL https://arxiv.org/abs/2502.13555.

Rahatara Ferdousi, M. Anwar Hossain, Chunsheng Yang, and Abdulmotaleb El Saddik. Defecttwin: When llm meets digital twin for railway defect inspection, 2024. URL https://arxiv.org/abs/2409.06725.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N. Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for rlhf, 2024. URL https://arxiv.org/abs/2410.14872.

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, Jingqun Tang, Wei Chen, Lianwen Jin, Yuliang Liu, and Xiang Bai. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning, 2025. URL https://arxiv.org/abs/2501.00321.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query, 2017. URL https://arxiv.org/abs/1705.02101.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16477–16508, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.910. URL https://aclanthology.org/2023.acl-long.910/.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023b. URL https://arxiv.org/abs/2211.10435.

Yuan Gao, Mattia Piccinini, Korbinian Möller, Amr Alanwar, and Johannes Betz. From words to collisions: LLM-guided evaluation and adversarial generation of safety-critical driving scenarios. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2025. doi: 10.48550/arXiv.2502.02145. URL https://arxiv.org/abs/2502.02145.

Noam Gat. LM Format Enforcer, 2025. URL https://github.com/noamgat/lm-format-enforcer.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models, 2020. URL https://arxiv.org/abs/2009.11462.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. Jsonschemabench: A rigorous benchmark of structured outputs for language models, 2025. URL https://arxiv.org/abs/2501.10868.

Abraham George and Amir Barati Farimani. Llm trainer: Automated robotic data generating via demonstration augmentation using llms, 2025. URL https://arxiv.org/abs/2509.20070.

Kamran Gholizadeh HamlAbadi, Monica Vahdati, Haiwei Dong, and Abdulmotaleb El Saddik. Ai-enhanced creation of digital twins from iphone lidar for immersive xr experiences in nvidia omniverse. In *Proceedings of the International Workshop on Intelligent Immersification in the Metaverse: AI-Driven Immersive Multimedia*, I2M-MM '25, pp. 25–33, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400718441. doi: 10.1145/3728487.3759232. URL https://doi.org/10.1145/3728487.3759232.

Google DeepMind. Synthid, 2025. URL https://deepmind.google/models/synthid.

Sven Gowal, Rudy Bunel, Florian Stimberg, David Stutz, Guillermo Ortiz-Jimenez, Christina Kouridi, Mel Vecerik, Jamie Hayes, Sylvestre-Alvise Rebuffi, Paul Bernard, Chris Gamble, Miklós Z. Horváth, Fabian Kaczmarczyck, Alex Kaskasoli, Aleksandar Petrov, Ilia Shumailov, Meghana Thotakuri, Olivia Wiles, Jessica Yung, Zahra Ahmed, Victor Martin, Simon Rosen, Christopher Savčak, Armin Senoner, Nidhi Vyas, and Pushmeet Kohli. Synthid-image: Image watermarking at internet scale, 2025. URL https://arxiv.org/abs/2510.09263.

Mandeep Goyal and Qusay H. Mahmoud. A systematic review of synthetic data generation techniques using generative AI. *Electronics*, 13(17):3509, 2024. doi: 10.3390/electronics13173509. URL https://www.mdpi.com/2079-9292/13/17/3509.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025. URL https://arxiv.org/abs/2411.15594.

Cusuh Ham, Matthew Fisher, James Hays, Nicholas I. Kolkin, Yuchen Liu, Richard Zhang, and Tobias Hinz. Personalized residuals for concept-driven text-to-image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8186–8195. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00782. URL https://doi.org/10.1109/CVPR52733.2024.00782.

Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. Evaluating large language models in generating synthetic hci research data: a case study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM, 2023. doi: 10.1145/3544548.3580688. URL https://dl.acm.org/doi/10.1145/3544548.3580688.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning, 2016. URL https://arxiv.org/abs/1610.02413.

Pinjia He, Jieming Zhu, Zibin Zheng, and Michael R. Lyu. Drain: An online log parsing approach with fixed depth tree. In *2017 IEEE International Conference on Web Services (ICWS)*, pp. 33–40, 2017. doi: 10.1109/ICWS.2017.13. URL https://ieeexplore.ieee.org/document/8029742.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. URL https://arxiv.org/abs/2104.08718.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model, 2024. URL https://arxiv.org/abs/2403.07691.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.naacl-main.287/.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scene as blender code, 2024. URL https://arxiv.org/abs/2403.01248.

Haoyu Huang, Chong Chen, Zeang Sheng, Yang Li, and Wentao Zhang. Can llms be good graph judge for knowledge graph construction?, 2025a. URL https://arxiv.org/abs/2411.17388.

Kung-Hsiang Huang, Mingyang Zhou, Hou Pong Chan, Yi R. Fung, Zhenhailong Wang, Lingyu Zhang, Shih-Fu Chang, and Heng Ji. Do lvlms understand charts? analyzing and correcting factual errors in chart captioning, 2024a. URL https://arxiv.org/abs/2312.10160.

Siyuan Huang, Zhengkai Jiang, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model, 2023. URL https://arxiv.org/abs/2305.11176.

Yingbing Huang, Deming Chen, and Abhishek K. Umrawal. Jam: Controllable and responsible text generation via causal reasoning and latent vector manipulation, 2025b. URL https://arxiv.org/abs/2502.20684.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

HuggingFaceFW. Fineweb-edu classifier (model card). https://huggingface.co/HuggingFaceFW/fineweb-edu-classifier, 2024.

Yintong Huo, Yichen Li, Yuxin Su, Pinjia He, Zifan Xie, and Michael R. Lyu. Autolog: A log sequence synthesis framework for anomaly detection. In *38th IEEE/ACM International Conference on Automated Software Engineering, ASE 2023, Luxembourg, September 11-15, 2023*, pp. 497–509. IEEE, 2023. doi: 10.1109/ASE56229.2023.00133. URL https://doi.org/10.1109/ASE56229.2023.00133.

Shadi Iskander, Sofia Tolmach, Ori Shapira, Nachshon Cohen, and Zohar Karnin. Quality matters: Evaluating synthetic data for tool-using LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4958–4976, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.285. URL https://aclanthology.org/2024.emnlp-main.285/.

MinHyuk Jang, Youngdong Jang, JaeHyeok Lee, Feng Yang, Gyeongrok Oh, Jongheon Jeong, and Sangpil Kim. Lvmark: Robust watermark for latent video diffusion models, 2025. URL https://arxiv.org/abs/2412.09122.

Jiarui Ji, Runlin Lei, Jialing Bi, Zhewei Wei, Xu Chen, Yankai Lin, Xuchen Pan, Yaliang Li, and Bolin Ding. Llm-based multi-agent systems are scalable graph generative models, 2025. URL https://arxiv.org/abs/2410.09824.

Zhihan Jiang, Jinyang Liu, Zhuangbin Chen, Yichen Li, Junjie Huang, Yintong Huo, Pinjia He, Jiazhen Gu, and Michael R. Lyu. Lilac: Log parsing using llms with adaptive parsing cache, 2024a. URL https://arxiv.org/abs/2310.01796.

Zhihan Jiang, Jinyang Liu, Junjie Huang, Yichen Li, Yintong Huo, Jiazhen Gu, Zhuangbin Chen, Jieming Zhu, and Michael R. Lyu. A large-scale evaluation for log parsing techniques: How far are we?, 2024b. URL https://arxiv.org/abs/2308.10828.

Jing Jin and Houfeng Wang. Select high-quality synthetic QA pairs to augment training data in MRC under the reward guidance of generative language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 14543–14554, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1267/.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling, 2016. URL https://arxiv.org/abs/1602.02410.

Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference (TCC 2013)*, volume 7785 of *Lecture Notes in Computer Science*, pp. 457–476. Springer, 2013. doi: 10.1007/978-3-642-36594-2_26. URL https://dl.acm.org/doi/10.1007/978-3-642-36594-2_26.

Zanis Ali Khan, Donghwan Shin, Domenico Bianculli, and Lionel C. Briand. Guidelines for assessing the accuracy of log message template identification techniques. In *Proceedings of the 44th International Conference on Software Engineering (ICSE '22)*, pp. 1095–1106. Association for Computing Machinery, 2022. doi: 10.1145/3510003.3510101. URL https://dl.acm.org/doi/10.1145/3510003.3510101.

Jinhee Kim, Taesung Kim, and Jaegul Choo. Epic: Effective prompting for imbalanced-class data synthesis in tabular data classification via large language models, 2025. URL https://arxiv.org/abs/2404.12404.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models, 2023. URL https://arxiv.org/abs/2305.17216.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL https://arxiv.org/abs/2205.11916.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Josh Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation, 2024. URL https://arxiv.org/abs/2312.14125.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016. URL https://arxiv.org/abs/1602.07332.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations – democratizing large language model alignment, 2023. URL https://arxiv.org/abs/2304.07327.

Peng Lai, Jianjie Zheng, Sijie Cheng, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. Beyond the surface: Enhancing llm-as-a-judge alignment with human via internal representations, 2025. URL https://arxiv.org/abs/2508.03550.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024. URL https://arxiv.org/abs/2403.13787.

Duy Le, Kris Zhao, Mengying Wang, and Yinghui Wu. Graphlingo: Domain knowledge exploration by synchronizing knowledge graphs and large language models. In *IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024. URL https://ieeexplore.ieee.org/document/10597884.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback, 2024. URL https://arxiv.org/abs/2309.00267.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: A comprehensive survey on llm-based evaluation methods, 2024a. URL https://arxiv.org/abs/2412.05579.

Jia Li and Guoxiang Zhao. T^3 planner: A self-correcting llm framework for robotic motion planning with temporal logic, 2025. URL https://arxiv.org/abs/2510.16767.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014/.

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, Lingpeng Kong, and Qi Liu. VLFeedback: A large-scale AI feedback dataset for large vision-language models alignment. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6227–6246, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.358. URL https://aclanthology.org/2024.emnlp-main.358/.

Rui Li, Heming Xia, Xinfeng Yuan, Qingxiu Dong, Lei Sha, Wenjie Li, and Zhifang Sui. How far are llms from being our digital twins? a benchmark for persona-based behavior chain simulation. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025a. URL https://aclanthology.org/2025.findings-acl.813/.

Shuyang Li, Talha Azfar, and Ruimin Ke. Chatsumo: Large language model for automating traffic scenario generation in simulation of urban mobility, 2024c. URL https://arxiv.org/abs/2409.09040.

Xinqing Li, Xin He, Le Zhang, Min Wu, Xiaoli Li, and Yun Liu. A comprehensive survey on world models for embodied ai, 2025b. URL https://arxiv.org/abs/2510.16732.

Yichen Li, Yintong Huo, Zhihan Jiang, Renyi Zhong, Pinjia He, Yuxin Su, Lionel C. Briand, and Michael R. Lyu. Exploring the effectiveness of llms in automated logging statement generation: An empirical study. *IEEE Trans. Softw. Eng.*, 50(12):3188–3207, December 2024d. ISSN 0098-5589. doi: 10.1109/TSE.2024.3475375. URL https://doi.org/10.1109/TSE.2024.3475375.

Yue Li, Xin Yi, Dongsheng Shi, Yongyi Cui, Gerard de Melo, and Linlin Wang. From injection to defense: Constructing edit-based fingerprints for large language models, 2025c. URL https://arxiv.org/abs/2509.03122.

Zhecheng Li, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. DRS: Deep question reformulation with structured output. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12869–12882, Vienna, Austria, July 2025d. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.666. URL https://aclanthology.org/2025.findings-acl.666/.

Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. Llm-grounded video diffusion models, 2024. URL https://arxiv.org/abs/2309.17444.

Hao Liang, Xiaochen Ma, Zhou Liu, Zhen Hao Wong, Zhengyang Zhao, Zimo Meng, Runming He, Chengyu Shen, Qifeng Cai, Zhaoyang Han, Meiyi Qiang, Yalin Feng, Tianyi Bai, Zewei Pan, Ziyi Guo, Yizhen Jiang, Jingwen Deng, Qijie You, Peichao Lai, Tianyu Guo, Chi Hsu Tsai, Hengyi Feng, Rui Hu, Wenkai Yu, Junbo Niu, Bohan Zeng, Ruichuan An, Lu Ma, Jihao Huang, Yaowei Zheng, Conghui He, Linpeng Tang, Bin Cui, Weinan E, and Wentao Zhang. Dataflow: An llm-driven framework for unified data preparation and workflow automation in the era of data-centric ai, 2025. URL https://arxiv.org/abs/2512.16676.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023. URL https://arxiv.org/abs/2209.07753.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. Controllable text generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2408.12599.

Mingxiang Liao, Hannan Lu, Xinyu Zhang, Fang Wan, Tianyu Wang, Yuzhong Zhao, Wangmeng Zuo, Qixiang Ye, and Jingdong Wang. Evaluation of text-to-video generation models: A dynamics perspective, 2024. URL https://arxiv.org/abs/2407.01094.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Charlie Nash, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard S. Zemel. Efficient graph generation with graph recurrent attention networks, 2020. URL https://arxiv.org/abs/1910.00760.

Han Lin, Abhay Zala, Jaemin Cho, and Mohit Bansal. Videodirectorgpt: Consistent multi-scene video generation via llm-guided planning, 2024. URL https://arxiv.org/abs/2309.15091.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback, 2025a. URL https://arxiv.org/abs/2501.13918.

Junteng Liu, Yuanxiang Fan, Zhuo Jiang, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong Weng, Aili Chen, Shiqi Chen, Yunan Huang, Mozhi Zhang, Pengyu Zhao, Junjie Yan, and Junxian He. Synlogic: Synthesizing verifiable reasoning data at scale for learning logical reasoning and beyond, 2025b. URL https://arxiv.org/abs/2505.19641.

Runtao Liu, Haoyu Wu, Zheng Ziqiang, Chen Wei, Yingqing He, Renjie Pi, and Qifeng Chen. Videodpo: Omni-preference alignment for video diffusion generation, 2024a. URL https://arxiv.org/abs/2412.14167.

Shang Liu, Hao Du, Yang Cao, Bo Yan, Jinfei Liu, and Masatoshi Yoshikawa. Pgb: Benchmarking differentially private synthetic graph generation algorithms, 2024b. URL https://arxiv.org/abs/2408.02928.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LVI*, pp. 386–403, Berlin, Heidelberg, 2024c. Springer-Verlag. ISBN 978-3-031-72991-1. doi: 10.1007/978-3-031-72992-8_22. URL https://doi.org/10.1007/978-3-031-72992-8_22.

Yilun Liu, Shimin Tao, Weibin Meng, Jingyu Wang, Wenbing Ma, Yanqing Zhao, Yuhang Chen, Hao Yang, Yanfei Jiang, and Xun Chen. Interpretable online log analysis using large language models with prompt strategies, 2024d. URL https://arxiv.org/abs/2308.07610.

Gabriel Loiseau, Damien Sileo, Damien Riquet, Maxime Meyer, and Marc Tommasi. Tau-eval: A unified evaluation framework for useful and private text anonymization. In Ivan Habernal, Peter Schulam, and Jörg Tiedemann (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 216–227, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-334-0. doi: 10.18653/v1/2025.emnlp-demos.16. URL https://aclanthology.org/2025.emnlp-demos.16/.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11065–11082, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.658. URL https://aclanthology.org/2024.findings-acl.658/.

Yunbo Long, Liming Xu, and Alexandra Brintrup. Llm-tablogic: Preserving inter-column logical relationships in synthetic tabular data via prompt-guided latent diffusion, 2025. URL https://arxiv.org/abs/2503.02161.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests, 2018. URL https://arxiv.org/abs/1610.06545.

Qiujing Lu, Meng Ma, Ximiao Dai, Xuanhan Wang, and Shuo Feng. Realistic corner case generation for autonomous vehicles with multimodal large language model, 2024. URL https://arxiv.org/abs/2412.00243.

Yaxi Lu, Haolun Li, Xin Cong, Zhong Zhang, Yesai Wu, Yankai Lin, Zhiyuan Liu, Fangming Liu, and Maosong Sun. Learning to generate structured output with schema reinforcement learning, 2025. URL https://arxiv.org/abs/2502.18878.

Yu Lu, Linchao Zhu, Hehe Fan, and Yi Yang. Flowzero: Zero-shot text-to-video synthesis with llm-driven dynamic scene syntax, 2023. URL https://arxiv.org/abs/2311.15813.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7c04aea54c2a60a632a47bd451cd2849-Paper-Conference.pdf.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct, 2025b. URL https://arxiv.org/abs/2306.08568.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models, 2024a. URL https://arxiv.org/abs/2310.12931.

Zeyang Ma, Dong Jae Kim, and Tse-Hsun Chen. Librelog: Accurate and efficient unsupervised log parsing using open-source large language models, 2024b. URL https://arxiv.org/abs/2408.01585.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models, 2024. URL https://arxiv.org/abs/2410.12832.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023. URL https://arxiv.org/abs/2303.08896.

Margherita Martorana, Tobias Kuhn, Lise Stork, and Jacco van Ossenbruggen. Zero-shot topic classification of column headers: Leveraging llms for metadata enrichment, 2024. URL https://arxiv.org/abs/2403.00884.

Shikib Mehri and Maxine Eskenazi. Usr: An unsupervised and reference free evaluation metric for dialog generation, 2020. URL https://arxiv.org/abs/2005.00456.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward, 2024. URL https://arxiv.org/abs/2405.14734.

Yibo Miao, Yifan Zhu, Yinpeng Dong, Lijia Yu, Jun Zhu, and Xiao-Shan Gao. T2vsafetybench: Evaluating the safety of text-to-video generative models, 2024. URL https://arxiv.org/abs/2407.05965.

Marko Miletic and Murat Sariyar. Large language models for synthetic tabular health data: A benchmark study. *Studies in health technology and informatics*, 316:963–967, 08 2024. doi: 10.3233/SHTI240571. URL https://pubmed.ncbi.nlm.nih.gov/39176952/.

Michael J. Mior. Large language models for json schema discovery, 2024. URL https://arxiv.org/abs/2407.03286.

Mohammad Ghiasvand Mohammadkhani, Saeedeh Momtazi, and Hamid Beigy. A survey on bridging VLMs and synthetic data. *Authorea Preprints*, 2025. doi: 10.36227/techrxiv.174741263.32891073. URL https://www.techrxiv.org/doi/full/10.36227/techrxiv.174741263.32891073.

R. Mon-Williams et al. Embodied large language models enable robots to complete complex tasks in unpredictable environments. *Nature Machine Intelligence*, 2025. URL https://www.nature.com/articles/s42256-025-01005-x.

Steven Morad, Ajay Shankar, Jan Blumenkamp, and Amanda Prorok. Language-conditioned offline rl for multi-robot navigation, 2024. URL https://arxiv.org/abs/2407.20164.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of llms via principled synthetic logic corpus, 2024. URL https://arxiv.org/abs/2411.12498.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633, 2025. ISSN 2169-3536. doi: 10.1109/access.2025.3589503. URL http://dx.doi.org/10.1109/ACCESS.2025.3589503.

Mehreen Naeem, Andrew Melnik, and Michael Beetz. Grounding language models with semantic digital twins for robotic planning, 2025. URL https://arxiv.org/abs/2506.16493.

Jaehyun Nam, Kyuyoung Kim, Seunghyuk Oh, Jihoon Tack, Jaehyung Kim, and Jinwoo Shin. Optimized feature generation for tabular data via llms with decision tree reasoning, 2024. URL https://arxiv.org/abs/2406.08527.

Yatin Nandwani, Vineet Kumar, Dinesh Raghu, Sachindra Joshi, and Luis A. Lastras. Pointwise mutual information based metric and decoding strategy for faithful generation in document grounded dialogs, 2023. URL https://arxiv.org/abs/2305.12191.

Dang Nguyen, Sunil Gupta, Kien Do, Thin Nguyen, and Svetha Venkatesh. Generating realistic tabular data with large language models, 2024. URL https://arxiv.org/abs/2410.21717.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann

Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models, 2024. URL https://arxiv.org/abs/2310.02992.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning, 2024. URL https://arxiv.org/abs/2402.13950.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. Fineweb2: One pipeline to scale them all — adapting pre-training data processing to every language, 2025. URL https://arxiv.org/abs/2506.20920.

Wujian Peng, Lingchen Meng, Yitong Chen, Yiweng Xie, Yang Liu, Tao Gui, Hang Xu, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. Inst-it: Boosting multimodal instance understanding via explicit visual prompt instruction tuning, 2024. URL https://arxiv.org/abs/2412.03565.

Vasileios C. Pezoulas, Dimitrios I. Zaridis, Eugenia Mylona, Christos Androutsos, Kosmas Apostolidis, Nikolaos S. Tachos, and Dimitrios I. Fotiadis. Synthetic data generation methods in healthcare: A review on open-source tools and methods. *Computational and Structural Biotechnology Journal*, 23:2892–2910, 2024. doi: 10.1016/j.csbj.2024.07.005. URL https://www.sciencedirect.com/science/article/pii/S2001037024002393.

Vitou Phy, Yang Zhao, and Akiko Aizawa. Deconstruct to reconstruct a configurable evaluation metric for open-domain dialogue systems, 2020. URL https://arxiv.org/abs/2011.00483.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. The text anonymization benchmark (TAB): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101, December 2022. doi: 10.1162/coli_a_00458. URL https://aclanthology.org/2022.cl-4.19/.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2641–2649, 2015. doi: 10.1109/ICCV.2015.303. URL https://ieeexplore.ieee.org/document/7410660.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. Receval: Evaluating reasoning chains via correctness and informativeness, 2023. URL https://arxiv.org/abs/2304.10703.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery, 2018. URL https://arxiv.org/abs/1803.09518.

Prime Intellect Team. Synthetic-2 release: Four million collaboratively generated verified reasoning traces, July 2025. URL https://www.primeintellect.ai/blog/synthetic-2-release.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5811–5826, Online, 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-main.468/.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on*

Machine Learning (ICML), volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Natraj Raman and Sameena Shah. Synthetic text generation using hypergraph representations, 2023. URL https://arxiv.org/abs/2309.06550.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, December 2023. doi: 10.1162/coli_a_00486. URL https://aclanthology.org/2023.cl-4.2/.

Yaxuan Ren, Krithika Ramesh, Yaxing Yao, and Anjalie Field. How do we measure privacy in text? a survey of text anonymization metrics, 2025. URL https://arxiv.org/abs/2512.01109.

Gurvan Richardeau, Samy Chali, Erwan Le Merrer, Camilla Penzo, and Gilles Tredan. Llms prompted for graphs: Hallucinations and generative capabilities, 2025. URL https://arxiv.org/abs/2409.00159.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning, 2019. URL https://arxiv.org/abs/1809.02156.

Bo-Kai Ruan, Hao-Tang Tsui, Yung-Hui Li, and Hong-Han Shuai. Traffic scene generation from natural language description for autonomous vehicles with large language model, 2025. URL https://arxiv.org/abs/2409.09575.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510. IEEE, 2023. doi: 10.1109/CVPR52729.2023.02155. URL https://doi.org/10.1109/CVPR52729.2023.02155.

Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping, 2017. URL https://arxiv.org/abs/1611.06624.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. URL https://arxiv.org/abs/1606.03498.

Tanmay Vilas Samak, Chinmay Vilas Samak, Bing Li, and Venkat Krovi. When digital twins meet large language models: Realistic, interactive, and editable simulation for autonomous driving, 2025. URL https://arxiv.org/abs/2507.00319.

Soumya Sanyal, Harman Singh, and Xiang Ren. Fairr: Faithful and robust deductive reasoning over natural language, 2022. URL https://arxiv.org/abs/2203.10261.

Ali Satvaty, Suzan Verberne, and Fatih Turkmen. Undesirable memorization in large language models: A survey, 2025. URL https://arxiv.org/abs/2410.02650.

Martin Scaiano, Grant Middleton, Luk Arbuckle, Varada Kolhatkar, Liam Peyton, Moira Dowling, Debbie S Gipson, and Khaled El Emam. A unified framework for evaluating the risk of re-identification of text de-identification tools. *Journal of Biomedical Informatics*, 63:174–183, October 2016. doi: 10.1016/j.jbi.2016.07.015. URL https://www.sciencedirect.com/science/article/pii/S1532046416300697.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. Curated llm: Synergy of llms and data curation for tabular augmentation in low-data regimes, 2024. URL https://arxiv.org/abs/2312.12112.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.439. URL https://aclanthology.org/2020.emnlp-main.439/.

Yashothara Shanmugarasa, Ming Ding, Mahawaga Arachchige Pathum Chamikara, and Thierry Rakotoarivelo. SoK: The privacy paradox of large language models: Advancements, privacy risks, and mitigation. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '25, pp. 425–441. ACM, 2025. doi: 10.1145/3708821.3733888. URL https://arxiv.org/abs/2506.12699.

Mysore N. Sharath and Babak Mehran. A literature review of performance metrics of automated driving systems for on-road vehicles. *Frontiers in Future Transportation*, 2:759125, 2021. doi: 10.3389/ffutr.2021.759125. URL https://www.frontiersin.org/articles/10.3389/ffutr.2021.759125.

Junyao Shi, Joshua Smith, Jianing Qian, and Dinesh Jayaraman. Points2reward: Robotic manipulation rewards from just one video. *Robotics: Science and Systems (RSS)*, 2025a. URL https://scholar.google.com/citations?view_op=view_citation&hl=en&user=XAhGkq8AAAAJ&citation_for_view=XAhGkq8AAAAJ:eQOLeE2rZwMC.

Ruxue Shi, Yili Wang, Mengnan Du, Xu Shen, and Xin Wang. A comprehensive survey of synthetic tabular data generation. *arXiv preprint arXiv:2504.16506*, 2025b. doi: 10.48550/arXiv.2504.16506. URL https://arxiv.org/abs/2504.16506.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2017. URL https://arxiv.org/abs/1610.05820.

Wai Man Si, Michael Backes, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, Savvas Zannettou, and Yang Zhang. Why so toxic? measuring and triggering toxic behavior in open-domain chatbots, 2022. URL https://arxiv.org/abs/2209.03463.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models, 2022. URL https://arxiv.org/abs/2209.11302.

Praphul Singh, Charlotte Dzialo, Jangwon Kim, Sumana Srivatsa, Irfan Bulu, Sri Gadde, and Krishnaram Kenthapadi. Redactor: An llm-powered framework for automatic clinical data de-identification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 510–530, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-industry.36. URL https://aclanthology.org/2025.acl-industry.36/.

Ilhan Skender, Kailin Tong, Selim Solmaz, and Daniel Watzenig. Investigating traffic accident detection using multimodal large language models, 2025. URL https://arxiv.org/abs/2509.19096.

Aivin V. Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers, 2023. URL https://arxiv.org/abs/2302.02041.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15725–15788, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.840. URL https://aclanthology.org/2024.acl-long.840/.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality, 2024. URL https://arxiv.org/abs/2307.05222.

Anirudh Sundar, Christopher Richardson, and Larry Heck. gtbls: Generating tables from text by conditional question answering, 2024. URL https://arxiv.org/abs/2403.14457.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Boyin Tan, Junjielong Xu, Zhouruixing Zhu, and Pinjia He. Al-bench: A benchmark for automatic logging, 2025. URL https://arxiv.org/abs/2502.03160.

Jingqun Tang, Chunhui Lin, Zhen Zhao, Shu Wei, Binghong Wu, Qi Liu, Yangfan He, Kuan Lu, Hao Feng, Yang Li, Siqi Wang, Lei Liao, Wei Shi, Yuliang Liu, Hao Liu, Yuan Xie, Xiang Bai, and Can Huang. Textsquare: Scaling up text-centric visual instruction tuning, 2025. URL https://arxiv.org/abs/2404.12803.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. Ruber: an unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8. URL https://dl.acm.org/doi/10.5555/3504035.3504124.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2025. URL https://arxiv.org/abs/2405.09818.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020. URL https://arxiv.org/abs/2003.12039.

Changyao Tian, Xizhou Zhu, Yuwen Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Yuntao Chen, Lewei Lu, Tong Lu, Jie Zhou, Hongsheng Li, Yu Qiao, and Jifeng Dai. Mm-interleaved: Interleaved image-text generative modeling via multi-modal feature synchronizer, 2024. URL https://arxiv.org/abs/2401.10208.

Together Computer. Redpajama: an open dataset for training large language models, 2023. URL https://huggingface.co/datasets/togethercomputer/RedPajama-Data-V2.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset, 2024. URL https://arxiv.org/abs/2402.10176.

Olivier Toubia, George Z. Gui, Tianyi Peng, Daniel J. Merlau, Ang Li, and Haozhe Chen. Twin-2k-500: A dataset for building digital twins of over 2,000 people based on their answers to over 500 questions. *Marketing Science*, 2025. doi: 10.1287/mksc.2025.0262. URL https://pubsonline.informs.org/doi/10.1287/mksc.2025.0262.

Toan V. Tran and Li Xiong. Differentially private tabular data synthesis using large language models, 2024. URL https://arxiv.org/abs/2406.01457.

Jeanine Treffers-Daller, Patrick Parslow, and Shirley Williams. Back to basics: how measures of lexical diversity can help discriminate between CEFR levels. *Applied Linguistics*, 39(3):302–327, 2018. doi: 10.1093/applin/amw009. URL https://doi.org/10.1093/applin/amw009.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019. URL https://arxiv.org/abs/1812.01717.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models, 2018. URL https://arxiv.org/abs/1610.02424.

vLLM Project. Structured Outputs, 2024. URL https://docs.vllm.ai/en/v0.8.2/features/structured_outputs.html.

Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, and Yunhong Wang. A survey on data synthesis and augmentation for large language models, 2024a. URL https://arxiv.org/abs/2410.12896.

Wenxuan Wang, Xiaoyuan Liu, Kuiyi Gao, Jen-tse Huang, Youliang Yuan, Pinjia He, Shuai Wang, and Zhaopeng Tu. Can't see the forest for the trees: Benchmarking multimodal safety awareness for multimodal llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. URL https://aclanthology.org/2025.acl-long.832/.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models, 2024b. URL https://arxiv.org/abs/2307.10635.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024c. URL https://arxiv.org/abs/2409.18869.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023a. URL https://arxiv.org/abs/2203.11171.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023b. URL https://arxiv.org/abs/2212.10560.

Yuxin Wang, Duanyu Feng, Yongfu Dai, Zhengyu Chen, Jimin Huang, Sophia Ananiadou, Qianqian Xie, and Hao Wang. Harmonic: Harnessing llms for tabular data synthesis and privacy protection, 2024d. URL https://arxiv.org/abs/2408.02927.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. URL https://ieeexplore.ieee.org/document/1284395.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T. Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. Codeclm: Aligning language models with tailored synthetic data, 2024e. URL https://arxiv.org/abs/2404.05875.

James R. Ward, Gabriel Agamennoni, Stewart Worrall, Asher Bender, and Eduardo Nebot. Extending time to collision for probabilistic reasoning in general traffic scenarios. *Transportation Research Part C: Emerging Technologies*, 51:66–82, 2015. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2014.11.002. URL https://www.sciencedirect.com/science/article/pii/S0968090X14003180.

Maurice Weber, Daniel Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. Redpajama: an open dataset for training large language models, 2024. URL https://arxiv.org/abs/2411.12372.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. Magicoder: Empowering code generation with oss-instruct, 2024. URL https://arxiv.org/abs/2312.02120.

Haoyang Wen, Jiang Guo, Yi Zhang, Jiarong Jiang, and Zhiguo Wang. On synthetic data strategies for domain-specific generative retrieval, 2025. URL https://arxiv.org/abs/2502.17957.

Martin Weyssow, Aton Kamanda, Xin Zhou, and Houari Sahraoui. Codeultrafeedback: An llm-as-a-judge dataset for aligning large language models to coding preferences, 2024. URL https://arxiv.org/abs/2403.09032.

Tsung-Han Wu, Long Lian, Joseph E. Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting llm-controlled diffusion models, 2023. URL https://arxiv.org/abs/2311.16090.

Yi Wu, Zikang Xiong, Yiran Hu, Shreyash S. Iyengar, Nan Jiang, Aniket Bera, Lin Tan, and Suresh Jagannathan. Selp: Generating safe and efficient task plans for robot agents with large language models, 2025. URL https://arxiv.org/abs/2409.19471.

Yuchen Xia, Daniel Dittler, Nasser Jazdi, Haonan Chen, and Michael Weyrich. Llm experiments with simulation: Large language model multi-agent system for simulation model parametrization in digital twins, 2024. URL https://arxiv.org/abs/2405.18092.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tUM39YTRxH.

Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Dongmei Zhang, and Surajit Chaudhuri. Table-llm-specialist: Language model specialists for tables using iterative generator-validator fine-tuning, 2024. URL https://arxiv.org/abs/2410.12164.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024a. URL https://openreview.net/forum?id=CfXh93NDgH.

Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles, 2022. URL https://arxiv.org/abs/2206.09682.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan, 2019. URL https://arxiv.org/abs/1907.00503.

Shengzhe Xu, Cho-Ting Lee, Mandar Sharma, Raquib Bin Yousuf, Nikhil Muralidhar, and Naren Ramakrishnan. Why llms are bad at synthetic table generation (and what to do about it), 2025. URL https://arxiv.org/abs/2406.14541.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Kang Liu, and Jun Zhao. Generate-on-graph: Treat llm as both agent and kg in incomplete knowledge graph question answering, 2024b. URL https://arxiv.org/abs/2404.14741.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024a. URL https://arxiv.org/abs/2409.12122.

June Yong Yang, Geondo Park, Joowon Kim, Hyeongwon Jang, and Eunho Yang. Language-interfaced tabular oversampling via progressive imputation and self-authentication. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=8F6bws5JBy.

Shuo Yang, Chenchen Yuan, Yao Rong, Felix Steinbauer, and Gjergji Kasneci. P-ta: Using proximal policy optimization to enhance tabular data augmentation via large language models, 2025. URL https://arxiv.org/abs/2406.11391.

Yanting Yang, Minghao Chen, Qibo Qiu, Jiahao Wu, Wenxiao Wang, Binbin Lin, Ziyu Guan, and Xiaofei He. Adapt2reward: Adapting video-language models to generalizable robotic rewards via failure prompts. In *European Conference on Computer Vision (ECCV)*, 2024c. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/07430.pdf.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Peiran Yao and Denilson Barbosa. Accurate and nuanced open-qa evaluation through textual entailment, 2024. URL https://arxiv.org/abs/2405.16702.

Yang Yao, Xin Wang, Zeyang Zhang, Yijian Qin, Ziwei Zhang, Xu Chu, Yuekui Yang, Wenwu Zhu, and Hong Mei. Exploring the potential of large language models in graph generation, 2024. URL https://arxiv.org/abs/2403.14358.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting, 2018. URL https://arxiv.org/abs/1709.01604.

Sheng Yin, Xianghe Pang, Yuanzhuo Ding, Menglan Chen, Yutong Bi, Yichen Xiong, Wenhao Huang, Zhen Xiang, Jing Shao, and Siheng Chen. Safeagentbench: A benchmark for safe task planning of embodied llm agents, 2025. URL https://arxiv.org/abs/2412.13178.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, Yudong Wang, Zijian Wu, Shuaibin Li, Fengzhe Zhou, Hongwei Liu, Songyang Zhang, Wenwei Zhang, Hang Yan, Xipeng Qiu, Jiayu Wang, Kai Chen, and Dahua Lin. Internlm-math: Open math large language models toward verifiable reasoning, 2024. URL https://arxiv.org/abs/2402.06332.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models, 2018. URL https://arxiv.org/abs/1802.08773.

Shanhe You, Xuewen Luo, Xinhe Liang, Jiashu Yu, Chen Zheng, and Jiangtao Gong. A comprehensive llm-powered framework for driving intelligence evaluation, 2025. URL https://arxiv.org/abs/2503.05164.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024a. URL https://arxiv.org/abs/2309.12284.

Qiying Yu, Quan Sun, Xiaosong Zhang, Yufeng Cui, Fan Zhang, Yue Cao, Xinlong Wang, and Jingjing Liu. Capsfusion: Rethinking image-text data at scale, 2024b. URL https://arxiv.org/abs/2310.20550.

Quan Yuan, Zhikun Zhang, Linkang Du, Min Chen, Peng Cheng, and Mingyang Sun. Privgraph: Differentially private graph data publication by exploiting community information, 2023. URL https://arxiv.org/abs/2304.02401.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

Yuwei Zeng, Yao Mu, and Lin Shao. Learning reward for robot skills using large language models via self-alignment, 2024. URL https://arxiv.org/abs/2405.07162.

Hao Zhang, Dongjun Yu, Lei Zhang, Guoping Rong, Yongda Yu, Haifeng Shen, He Zhang, Dong Shao, and Hongyu Kuang. Aucad: Automated construction of alignment dataset from log-related issues for enhancing llm-based log generation. In *Proceedings of the 16th International Conference on Internetware*, Internetware 2025, pp. 413–425. ACM, June 2025a. URL http://dx.doi.org/10.1145/3755881.3755889.

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024a. URL https://arxiv.org/abs/2310.09656.

Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles, 2024b. URL https://arxiv.org/abs/2405.14062.

Kaituo Zhang, Zhimeng Jiang, and Na Zou. Cleansing the artificial mind: A self-reflective detoxification framework for large language models, 2026. URL https://arxiv.org/abs/2601.11776.

Li Zhang, Hainiu Xu, Yue Yang, Shuyan Zhou, Weiqiu You, Manni Arora, and Chris Callison-Burch. Causal reasoning of entities and events in procedural texts. In *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 407–423. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-eacl.31.

Mingming Zhang, Zhiqing Xiao, Guoshan Lu, Sai Wu, Weiqiang Wang, Xing Fu, Can Yi, and Junbo Zhao. Aigt: Ai generative table based on prompt, 2024c. URL https://arxiv.org/abs/2412.18111.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Ruixuan Zhang, Beichen Wang, Juexiao Zhang, Zilin Bian, Chen Feng, and Kaan Ozbay. When language and vision meet road safety: Leveraging multimodal large language models for video-based traffic accident analysis. *Accident Analysis & Prevention*, 219:108077, September 2025b. ISSN 0001-4575. doi: 10.1016/j.aap.2025.108077. URL http://dx.doi.org/10.1016/j.aap.2025.108077.

Tianhui Zhang, Bei Peng, and Danushka Bollegala. Improving diversity of commonsense generation by large language models via in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9226–9242, Miami, Florida, USA, November 2024d. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.540. URL https://aclanthology.org/2024.findings-emnlp.540/.

Xuanyu Zhang, Youmin Xu, Runyi Li, Jiwen Yu, Weiqi Li, Zhipei Xu, and Jian Zhang. V2a-mark: Versatile deep visual-audio watermarking for manipulation localization and copyright protection, 2024e. URL https://arxiv.org/abs/2404.16824.

Yan Zhang, Ahmad Mohammad Saber, Amr Youssef, and Deepa Kundur. Grid-agent: An llm-powered multi-agent system for power grid control, 2025c. URL https://arxiv.org/abs/2508.05702.

Yongting Zhang, Lu Chen, Guodong Zheng, Yifeng Gao, Rui Zheng, Jinlan Fu, Zhenfei Yin, Senjie Jin, Yu Qiao, Xuanjing Huang, Feng Zhao, Tao Gui, and Jing Shao. Spa-vl: A comprehensive safety preference alignment dataset for vision language model, 2025d. URL https://arxiv.org/abs/2406.12030.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data, 2025e. URL https://arxiv.org/abs/2410.02713.

Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences, 2020. URL https://arxiv.org/abs/2001.06891.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Mingyu Zheng, Zhifan Feng, Jia Wang, Lanrui Wang, Zheng Lin, Yang Hao, and Weiping Wang. Tabledreamer: Progressive and weakness-guided data synthesis from scratch for table instruction tuning, 2025. URL https://arxiv.org/abs/2506.08646.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Shijie Zhou, Ruiyi Zhang, Yufan Zhou, and Changyou Chen. A high-quality text-rich image instruction tuning dataset via hybrid instruction generation, 2024. URL https://arxiv.org/abs/2412.16364.

Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, Mingyu Liu, Dingning Liu, Jiange Yang, Zhoujie Fu, Junyi Chen, Chunhua Shen, Jiangmiao Pang, Kaipeng Zhang, and Tong He. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling, 2025. URL https://arxiv.org/abs/2509.12201.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges, 2025. URL https://arxiv.org/abs/2310.17631.

A Notation Table

In this section, we provide the main notations we used in the survey.

Symbol	Meaning
Shared	
$E(\cdot)$	Encoder / embedding function
$\mathrm{sim}(\cdot, \cdot)$	Similarity (e.g., cosine)
Text / Dialogue	
$\mathcal{T}$	Set of generated texts
t_i	i-th generated text, $t_i \in \mathcal{T}$
p_i	Prompt / instruction for t_i
$\mathcal{D}$	Labeled evaluation dataset
$y_i, \hat{y}_i$	Gold / predicted label
w_m, L	m-th token; sequence length
τ	Acceptability threshold
$\mathcal{X}$	Corpus (set of instances)
Symbolic / Reasoning	
$\mathcal{R}$	Set of reasoning examples (q_i, c_i, y_i)
q_i	Question / problem for example i
c_i	Chain-of-thought / rationale for example i
$c_{i,t}$	t-th step in chain c_i
y_i	Final answer / conclusion for example i
$c_{i,k}$	k-th sampled chain for the same question q_i
$y_{i,k}$	Final answer induced by chain $c_{i,k}$
K	Number of sampled chains per question
Tabular	
$X_{\mathrm{real}}, X_{\mathrm{syn}}$	Real / synthetic tables
x_i	Row (record) in a table
y_i	Label / target for x_i (if any)
d	Number of features (columns)
Graphs / JSON / Logs	
$\mathcal{G}$	Set of graphs (real or generated)
G	A graph; $V(G), E(G)$: nodes / edges
$\mathcal{J}$	Set of JSON / structured records
J	A JSON key–value object
$\mathcal{L}$	Set of log entries or sequences
ℓ_i	A log entry or a log sequence
Vision-Language	
$\mathcal{M}$	Set of multi-modal samples
m_i	i-th sample, e.g., (v_i, t_i)
v_i	Visual / audio / video input paired with text
k	Number of modalities in m_i
Agent / Interaction	
$\mathcal{A}$	Set of agent trajectories / episodes
$\mathcal{A}_{\mathrm{gen}}$	Set of generated trajectories $\{\tau_i\}_{i=1}^{N}$
N	Number of trajectories / episodes in $\mathcal{A}_{\mathrm{gen}}$
τ_i	i-th trajectory $(s_0, a_0, r_1, s_1, \ldots, a_{T_i-1}, r_{T_i}, s_{T_i})$
T_i	Horizon (number of decision steps) of trajectory τ_i
s_t	State / observation at step t
a_t	Action (incl. tool/API call) at step t
r_{t+1}	Reward / feedback for transition (s_t, a_t, s_{t+1})

Table 8: Unified notation for core data objects across modalities.

B Related Work

We summarize related works and conduct a comparison with LLM Data Auditor across several dimensions, including Primary Focus, Organization, Quality, Trustworthy Evaluation, and Scope.

Table 9: Comparative Analysis of Surveys in Synthetic Data. While prevailing surveys structure the field by engineering workflows (Long et al., 2024), training lifecycles (Wang et al., 2024a), or specific modalities (Shi et al., 2025b), this work establishes a metric-centric taxonomy focused on the intrinsic evaluation and governance of cross-modality data.

Dimension	**Long et al. (2024)** Long et al. (2024)	**Wang et al. (2024)** Wang et al. (2024a)	**Shi et al. (2025)** Shi et al. (2025b)	**This Work**
Primary Focus	**The Engineering Pipeline**. Focuses on the operational workflow of constructing data, emphasizing generation techniques, curation strategies, and downstream application.	**The Model Lifecycle**. Focuses on the utility of synthetic data across distinct training stages, spanning pre-training, supervised fine-tuning, and alignment.	**The Generative Method**. Focuses on methodological families within the structured data domain, contrasting GANs, Variational Autoencoders, and Diffusion models.	**The Data Artifact**. Focuses on the intrinsic properties of the generated product, prioritizing rigorous auditing standards and data governance protocols.
Organization	**Procedure-Oriented**. Taxonomy defined by operational modules including prompt engineering, task decomposition, and heuristic filtering.	**Stage-Oriented**. Taxonomy structured around the LLM development lifecycle and downstream competencies such as reasoning and coding.	**Model-Centric**. Taxonomy categorized by the underlying generative architecture and associated post-processing techniques for tabular structures.	**Metric-Oriented**. Taxonomy defined by a unified evaluation coordinate system separating Quality dimensions from Trustworthiness dimensions.
Quality Evaluation	**Extrinsic Utility**. Quality is frequently judged by downstream gains, complemented by lightweight intrinsic proxies.	**Benchmark-Based**. Effectiveness is assessed via success rates on capability-specific evaluation suites and standard public benchmarks.	**Statistical Fidelity**. Evaluation prioritizes distributional resemblance to real data, machine learning efficacy, and column-wise statistical alignment.	**Intrinsic Verification**. Quality is defined through proactive audits of Validity, Fidelity, Diversity, and Utility prior to model training.
Trustworthy Evaluation	**Challenge-Oriented**. Hallucination and bias are discussed primarily as open challenges or limitations.	**Alignment-Oriented**. Safety is framed within the context of RLHF alignment.	**Privacy-Specific**. Analysis heavily concentrates on privacy guarantees.	**Foundational Pillar**. Elevates mainstream trustworthiness to a primary dimension orthogonal to utility metrics.
Scope	**Text-Dominant**. Primarily covers natural language processing tasks.	**Broad**. This work covers multiple modalities such as text, code, and vision.	**Tabular Data Specialized**. Addresses the constraints inherent to tabular data.	**Cross-Modality**. Applies a single framework to introduce 6 modal data.