UltraGen: Extremely Fine-grained Controllable Generation via Attribute Reconstruction and Global Preference Optimization

Anonymous ACL submission

Abstract

Fine granularity is an essential requirement for controllable text generation, which has seen rapid growth with the ability of LLMs. However, existing methods focus mainly on a small set of attributes like 3 to 5, and their performance degrades significantly when the number of attributes increases to the next order of magnitude. To address this challenge, we propose a novel zero-shot approach for extremely fine-grained controllable generation (EFCG), proposing *auto-reconstruction (AR)* and *global preference optimization (GPO)*. In the AR phase, we leverage LLMs to extract soft attributes (e.g., *Emphasis on simplicity and minimalism in design*) from raw texts, and combine them with programmatically derived hard attributes (e.g., *The text should be between 300 and 400 words*) to construct massive (around 45) multi-attribute requirements, which guide the fine-grained text reconstruction process under weak supervision. In the GPO phase, we apply direct preference optimization (DPO) to refine text generation under diverse attribute combinations, enabling efficient exploration of the global combination space. Additionally, we introduce an efficient attribute sampling strategy to identify and correct potentially erroneous attributes, further improving global optimization. Our framework significantly improves the constraint satisfaction rate (CSR) and text quality for EFCG by mitigating position bias and alleviating attention dilution.

1 Introduction

While large language models (LLMs) have shown promising performance in various tasks, processing massive input information remains a challenging setup (Liu et al., 2024). For controlled text generation (CTG) (Wang et al., 2023; Song et al., 2024), models are often required to satisfy a certain number of constraints simultaneously. In previous works, the typical number of constraints ranges from 3 to 5. However, when the number of constraints scales to the extreme (e.g. 30 or more), performance degrades significantly (Figure 1).

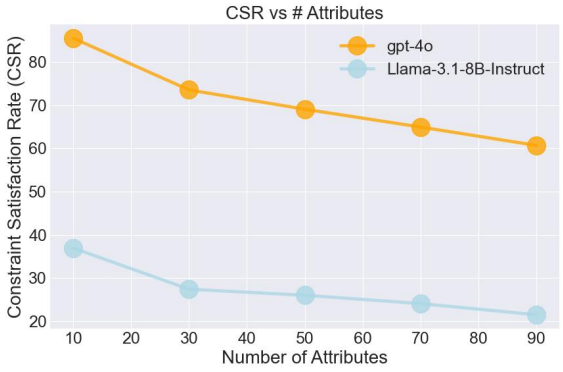

Figure 1: Constraint Satisfaction Rate (CSR) across different numbers of attributes for GPT-4o and LLaMA-3.1-8B-Instruct.

One representative challenge arises in travel itinerary planning (Appendix F). Consider a prompt requiring a detailed 5-day travel plan that satisfies over 30 constraints, covering timing, budget, transportation, meal preferences, and specific landmark visits. Despite LLMs' impressive fluency, they frequently violate crucial attributes like *each activity must be under 2 hours* or *avoid scheduling during 1 PM to 2 PM due to lunch break*. Why do LLMs perform well in general text generation but struggle under EFCG? We hypothesize that this limitation stems from two fundamental issues.

First, When the number of constraints increases, later-specified conditions are more likely to be neglected, as the model's performance exhibits position-dependent degradation (Liu et al., 2024), meaning that attributes appearing later in the prompt are less likely to be satisfied (Figure 3). Second, the pretraining and instruction-tuning corpus of LLMs involve simple prompts with a few loosely defined requirements. Models rarely encounter instances with 30+ precise constraints in a single prompt. As a result, when faced with ex-

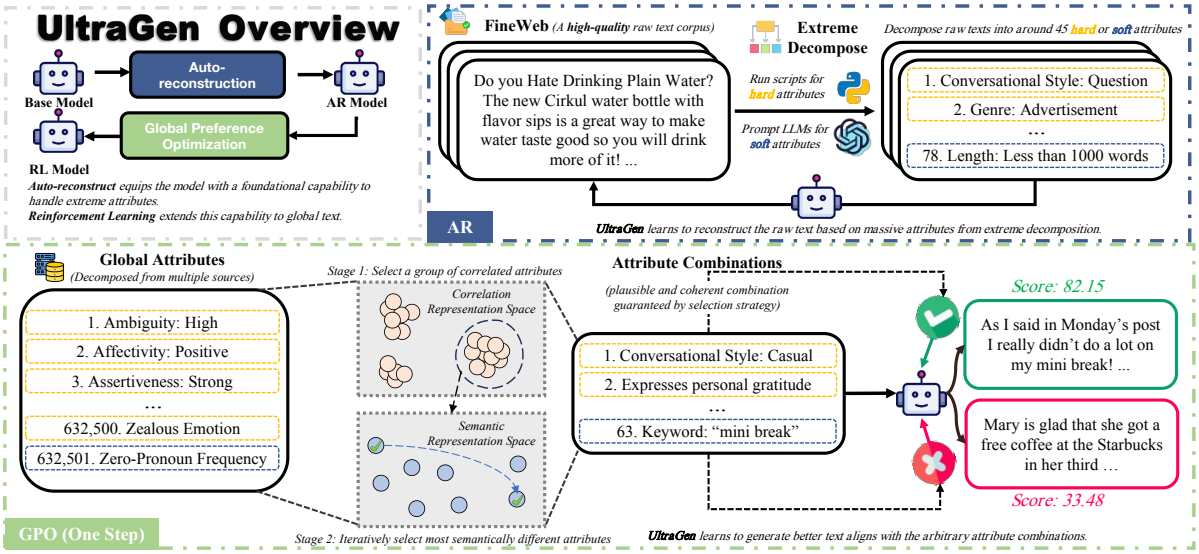

Figure 2: The whole pipeline of our two-stage **UltraGen** framework. The auto-reconstruction stage constructs a large-scale dataset by extracting soft and hard attributes from web corpora and then reconstructing the raw text. The global preference optimization stage applies DPO with attribute correlation modeling and diversity selection to enhance multi-attribute generalization over a global corpus.

treme constraint setups, models fail to maintain attention across all conditions, leading to attention dilution (Figure 6) during generation.

To tackle the challenges, we propose a framework designed to enhance LLMs' ability to handle a massive number of constraints effectively. We hypothesize that position bias arises partly due to the lack of exposure to diverse attribute positions during training. To mitigate this, we construct a large-scale automated dataset pipeline that extracts soft attributes (e.g., style, content) and hard attributes (e.g., keywords, structure) from natural texts without human annotation. A quality check is conducted to ensure that the attributes align with the raw texts (Section 3.3.1). By training LLMs on these realistic multi-attribute inputs, we expose the model to variable attributes across different positions, enabling it to better internalize the relationship between attributes and the text regardless of position.

Besides, as the number of constraints increases, the prompt space shifts towards an underrepresented distribution in the pre-training corpus, exacerbating attention dilution. To address this, we introduce a global preference optimization strategy in the second stage. First, we fine-tune an embedding model via contrastive learning to capture attribute correlations, encouraging the model to prioritize plausible and coherent attribute combinations. This helps steer generation away from implausible combinations rarely seen during pre-training. For example, consider an attribute set that requires the inclusion of an international politics term such as *impose 25% tariffs*, alongside a medical term like *myocardial infarction*. Such a combination is highly unlikely to appear together in real-world cases. Second, we promote diversity by selecting the least similar candidate from a pool of generations. This prevents the model from collapsing to a small set of frequent patterns and encourages exploration of less common yet valid combinations. Together, correlation modeling narrows the search space towards possible regions, while diversity selection expands coverage within that space, enabling the model to retain and balance a large set of attributes during generation.

Our contributions are threefold. First, we design an automated pipeline for dataset construction tailored to extreme constraints, enabling high-quality training and evaluation. Second, we develop a training strategy that integrates reconstruction and RL (Rafailov et al., 2024) to address the fundamental challenges in EFCG. Finally, we conduct extensive experiments to validate the proposed framework, providing insights into its efficacy and limitations.

2 Related Work

Controllable Text Generation CTG tasks involve hard constraints (e.g., text length, keyword inclusion)(Takase and Okazaki, 2019; Carlsson et al., 2022) and soft constraints (e.g., sentiment,

topic)(Gu et al., 2022; Lu et al., 2022). Fine-tuning LLMs with instructional data improves their constraint-following ability (Weller et al., 2020; Sanh et al., 2021; Mishra et al., 2022; Jiang et al., 2024), but evaluations show LLMs often fail to meet all constraints (Jiang et al., 2023; Qin et al., 2024; Ren et al., 2025). Despite this, these works primarily focus on a relatively small number of attributes or conditions, typically from 3 to 5, leaving a gap in understanding LLM's performance under more extreme requirements.

Evaluation of CTG Evaluating LLM's adherence to constraints is challenging and typically involves automatic and programmatic assessments using various metrics (Yao et al., 2023; Zhou et al., 2023b; Chen et al., 2022). Zhou et al. (2023a) centers on assessing 25 verifiable instructions. Jiang et al. (2023) progressively integrates fine-grained constraints to develop multi-level instructions, thereby enhancing complexity across six distinct types. Wen et al. (2024) constructs a novel benchmark by synthesizing and refining data from the aforementioned benchmarks, with an emphasis on the combinatorial types of constraints. Zhang et al. (2024) proposes a comprehensive constraint-following benchmark over 50 NLP tasks. However, none of them investigate the effects of extreme fine-grained attributes.

Multi-objective Alignment Recent work (Mudgal et al., 2023) focuses on balancing multiple objectives in text generation while maintaining linguistic quality. MORLHF (Zhou et al., 2023c; Rame et al., 2024) optimizes human preferences via reinforcement learning but is costly and unstable. RiC (Yang et al., 2024) reduces complexity by using supervised fine-tuning with multi-reward control and dynamic inference adjustment. DeAL (Huang et al., 2024) introduces a decoding-time alignment framework for large language models, enabling flexible customization of alignment objectives, such as keyword constraints and abstract goals like harmlessness, without requiring retraining.

3 Method

3.1 Preliminaries

EFCG: An Overview Our extremely EFCG is an extension to controllable text generation CTG, whose goal is to generate an output Y based on a given input X and a set of control conditions c.

Figure 3: Score degradation as the position of hard attributes shifts in Llama-3.1-8B-Instruct and Qwen2-7B-Instruct, showing a consistent performance drop.

Formally, this can be expressed as:

$$P(Y \mid X, c) = \prod_{i=1}^{n} P_\theta \left(Y_i \mid Y_{<i}, X, c\right)$$

where n denotes the length of Y, and θ represents the parameters of a language model, and $Y_{<i}$ refers to generated tokens before the i-th one. In conventional CTG models, X typically serves as a prompt, representing an incomplete text, while Y constitutes its continuation.

Despite advancements made in CTG, existing models often struggle to effectively handle fine-grained control conditions, particularly when the position of attributes within the input context varies. As illustrated in Figure 3, this limitation manifests as a significant performance degradation when attributes are positioned away from the beginning of the input context. Such trends highlight that modern language models may not robustly utilize information across the entire input sequence, with biases toward primacy regions of the context window. Addressing this challenge is crucial for improving the robustness and accuracy of EFCG task.

Semantic Similarity E5-large (Wang et al., 2022) is a powerful pre-trained encoder optimized for embedding sentences. In EFCG, we utilize E5-large to encode document-grounded attributes and measure their relationships using cosine similarity. Cosine similarity is a widely used metric to quantify the similarity between two high-dimensional vectors, defined as:

$$\mathrm{Sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$

where $\mathbf{u}$ and $\mathbf{v}$ represent the vector embeddings of two attributes. The cosine similarity yields val-

ues between -1 (completely dissimilar) and 1 (identical). Using cosine similarity, we identify semantically related attributes and filter out redundant or weakly correlated pairs.

3.2 UltraGen

As shown in Figure 2, our approach, UltraGen, addresses the challenge of EFCG through a two-stage framework. First, we introduce auto-reconstruction training to align text with a rich set of soft and hard attributes. Second, we apply global preference optimization to enhance the model's adaptability to diverse, globally complex attribute compositions.

3.2.1 Auto-reconstruction Stage

The auto-reconstruction stage trains on naturally aligned attribute-text pairs, where attributes are directly extracted from real texts of FineWeb (Penedo et al., 2024), a high-quality web corpus, to ensure intrinsic constraint compatibility (i.e., no conflicting attributes exist by construction). Then we train the model to re-generate the text based on the decomposed attributes. Formally, given a training example $(Y, c) \in D_{\text{UltraBench}}$, the model learns to reconstruct Y by minimizing the negative log-likelihood.

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(Y,c)} \log P_\theta(Y \mid c)$$

This process achieves dual objectives: **constraint grounding**, which forces the model to internalize the relationships between atomic attributes and their textual realizations, and **fluency preservation**, which maintains the base model's generative quality by leveraging the natural language distribution of the original corpus. The resulting reconstruct model serves as a coherent and constraint-aware initial policy for RL, providing essential prior knowledge for subsequent exploration of complicated constraint combinations.

3.2.2 Global Preference Optimization Stage

To extend the foundation capability to global text generation, we first collect a massive pool of attributes from multiple sources. The attributes pool integrates diverse sources spanning multiple domains, styles, and formats. Unlike reconstruction, which applies mainly web data, our RL phase leverages data from (1) Books, (2) Academic Papers (arXiv), (3) Social Media (Reddit), (4) Technical Forums (StackExchange), (5) News (CC-News), and (6) Encyclopedic Sources (Wikipedia). This ensures broad coverage of textual variations, enabling the model to generalize across different contexts and constraint types. In each iteration, a valid subset of attributes is selected from this pool. Using the auto-reconstruction model, we generate K candidate responses conditioned on the selected attributes. We then apply the CSR metric to identify the preferred and less favorable responses, which are subsequently used for DPO training.

A key challenge in selecting a valid subset of attributes lies in balancing **topic coherence** and **anti-redundancy**. Topic coherence requires a high correlation among attributes to ensure interdependent constraints are holistically satisfied. For example, keywords *chain of thought* and *use formal tone* jointly imply technical writing. In contrast, diversity prevents overfitting to frequent patterns and enhances fluency. For example, phrases like *the dreariest place, a dreary day* are redundant and make the text uninformative. Therefore, our pipeline comprises three key steps:

Attribute Correlation Modeling We fine-tune the E5-large encoder using triplet contrastive learning (Gao et al., 2021) on document-grounded attributes. For each anchor attribute $\mathcal{A}_i$, a positive pair $\mathcal{A}_j$ shares context from the same document, while a negative pair $\mathcal{A}_k$ is sampled from unrelated contexts. The encoder minimizes the triplet loss, yielding 81.6% validation accuracy in distinguishing correlated attributes.

Attribute Set Expansion The process begins by randomly sampling 2000 seed soft attributes from the attributes pool as the initial attribute set. For each seed attribute A_i, we retrieve its top 1024 most correlated candidates using the fine-tuned E5 encoder, where correlation is quantified by the cosine similarity in the correlation representation space using the fine-tuned model. To enforce diversity and minimize semantic redundancy, candidates are iteratively added to the set based on a redundancy score $\text{Sim}\left(A_{\text{candidate}}, A_i\right)$, which is defined as the cosine similarity in their original E5 semantic representation space. Expansion terminates when each set contains a randomly determined number from 10 to 110.

DPO Pair Generation For each attribute set, DPO training pairs are constructed by generating K responses using the auto-reconstruction model. The soft and hard attribute scores are obtained by using the Python scripts and GPT-4o (Section 3.3.3), respectively. The total score is computed by

averaging the two scores.

Responses are ranked from highest to lowest based on their scores. The highest-scoring response is chosen, while the lowest-scoring response is rejected. This automated scoring and ranking ensure the selection of thematically coherent and high-quality responses, refining the model's ability to distinguish and generate optimal outputs.

3.3 UltraBench

3.3.1 Dataset Construction

To support our training framework, we construct two specialized dataset splits named **UltraBench**, derived from FineWeb (Penedo et al., 2024) and multiple sources. The UltraBench dataset is designed to evaluate and train models on extremely fine-grained controllable text generation. Its construction involves Two stages, as detailed below.

Attribute Extraction Attributes were categorized into two types:

1. **Soft attributes:** (e.g., style, tone, content) were inferred using GPT-4o (Achiam et al., 2023) to capture semantic properties. For example, a soft attribute might describe a passage as a *vivid personal narrative focused on childbirth experience*.

2. **Hard attributes:** consist of programmatically verifiable constraints extracted directly from the text. These included keyword requirements (e.g., *include sustainability*), structural rules (e.g., *generate exactly three paragraphs*), and syntactic directives (e.g., *use all lowercase letters*).

For the FineWeb split, we use each attribute set along with its corresponding raw text to perform the auto-reconstruction stage. For the multi-sources split, we aggregate and de-duplicate all decomposed attributes to form a global attribute pool.

Consistency Verification To ensure the reliability of soft attribute extraction, we conducted a human evaluation on a randomly selected subset of 100 documents. Human experts assessed whether the extracted attributes accurately reflected the underlying text. We computed the Agreement Rate (AR), defined as the proportion of samples where automated extractions matched the original raw text. This process achieved an AR of 96.5%, indicating a strong alignment between attributes and original text.

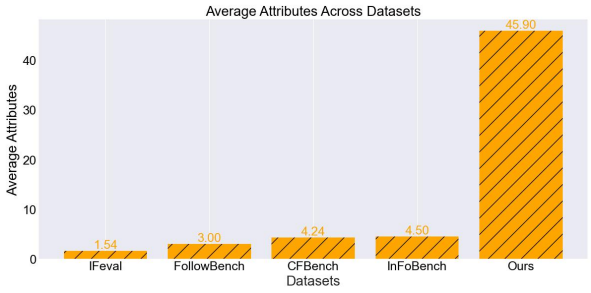

Figure 4: Comparison of average attributes across datasets.

3.3.2 Dataset Statistics

Overall Statistics In Appendix C, we summarize the dataset details. Table 4 details the composition of UltraBench, with separate configurations for reconstruction and multi-sources subsets. Table 6 further analyzes the multi-sources subset's domain distribution, while Table 7 quantifies quality control metrics.

Compared with Other Benchmarks Table 5 provides a detailed comparison of our dataset with other relevant works. While IFeval and FollowBench include synthesized data (Synt.), they fall short in capturing the diversity and complexity required for evaluating real-world applications. Another key strength of our dataset lies in the average number of attributes per sample, where we achieve a remarkable value of 45.9 and 29.9 on two splits, far exceeding the benchmarks' maximum of 4.5. This demonstrates the ability of our dataset for evaluating tasks requiring fine-grained attribute understanding.

3.3.3 Evaluation Protocol

To rigorously evaluate EFCG capabilities, we use two evaluation metrics:

Constraint Satisfaction Rate (CSR) For a given instruction with both soft and hard constraints, we compute the CSR as follows:

$$\text{CSR} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n^{(i)}}\sum_{j=1}^{n^{(i)}}s_j^{(i)}$$

where $s_j^{(i)} = 1$ if the j-th constraint for the i-th instruction is satisfied, and 0 otherwise. Here, $n^{(i)}$ is the number of constraints (hard or soft) for instruction i, and m is the total number of evaluated instructions.

1. **Hard Constraint Verification**: For programmatically verifiable constraints, we perform

deterministic checks via Python scripts. Due to the significant imbalance in hard attributes, we adopt macro accuracy to ensure fair evaluation. Macro accuracy computes the average CSR across different types, giving equal weight to each type regardless of its frequency.

2. **Soft Constraint Evaluation**: For semantic constraints, we employ an LLM-based judge (GPT-4o), assigning a binary score (0 or 1) to each constraint. We validate the quality of the LLM-based judges on a randomly selected set of 100 samples. By calculating the Cohen's Kappa coefficient between the scores of LLM-based judge and human experts, we found a strong agreement (84.55%) between the automatic evaluation and human experts' assessment.

BERTScore In the auto-reconstruction phase, we also use BERTScore (Zhang et al., 2020) to measure the quality of the reconstructed text. BERTScore leverages the contextual embeddings from pre-trained language models to capture semantic similarity. BERTScore is widely used in text generation tasks, as it aligns better with human judgments of semantic quality compared to traditional n-gram overlap-based metrics.

4 Experiments

4.1 Experiment Setup

Models. Our experiments evaluate the EFCG task using one mainstream instruction-tuned base model: Llama-3.2-3B-Instruct (Dubey et al., 2024), chosen for its demonstrated proficiency in instruction-following tasks within the 3B parameter range. To systematically assess the impact of our methodology, we compare three training paradigms: (1) **BASE**, which directly employs the unmodified base models to establish a performance baseline; (2) **AR**, where models undergo the auto-reconstruction stage on our meticulously constructed FineWeb dataset (§3.2), enriched with fine-grained attributes to enhance multi-constraint adherence; and (3) **AR+GPO**, a hybrid optimization approach combining direct preference optimization with global embedding space adaption.

4.2 Evaluation Results on UltraBench

Our experimental findings, summarized in Table 1, demonstrate the substantial advancements achieved by applying the UltraGen paradigm to EFCG. The evaluation leverages the validation set of FineWeb and Global splits to assess model performance under both local and global constraints.

The application of AR yielded significant improvements over the base model. On the FineWeb split, the AR model attained an overall score of 56.05, representing a relative improvement of 11.4%. The soft score rose to 81.44, indicating enhanced adherence to semantic and stylistic attributes, while the hard score increased to 30.65, reflecting better performance on programmatically verifiable constraints. On the Global split, the AR model demonstrated its ability to generalize, achieving an overall score of 50.15.

Further optimization through GPO demonstrated remarkable performance on the Global split, where the model achieved an overall score of 57.23 and an impressive hard score of 45.44. This highlights the model's robust generalization and optimization capabilities when dealing with diverse and challenging global constraints. Notably, despite being trained on the Global split, the AR+GPO model exhibited strong performance on the FineWeb split as well, achieving an overall score of 59.61, a soft score of 84.33, and a hard score of 34.89. This result underscores the model's ability to transfer its learned capabilities from the broader and more diverse Global split to the more localized FineWeb split.

Ablation To evaluate the contribution of key components in our UltraGen framework, we conducted ablation studies by systematically modifying the training process. We tested the impact of reducing the number of attributes during AR, removing the AR stage, replacing curated attributes with random sampling, and eliminating the high-correlation or low-redundancy selection steps. The results demonstrate that both AR and GPO stages are crucial for achieving strong performance, as reducing constraints, removing correlation modeling, or neglecting redundancy minimization leads to performance degradation.

4.3 Data Synthesis Improvement

To demonstrate the improvement in the usage of texts synthesized by UltraGen, we utilize several diverse well-established text classification benchmarks to test the data synthesis capability, such as sentiment analysis **(1) Emotion** (Saravia et al., 2018), attitude classification towards a particular public figure **(2) Hillary** (Barbieri et al., 2020),

	Model	FineWeb Split				Multi-source Split		
		Overall Score	Soft Score	Hard Score	BERTScore F1	Overall Score	Soft Score	Hard Score
Main	Base Model	50.30	67.08	33.51	59.92	37.45	36.10	38.79
	UltraGen (AR)	56.05	81.44	30.65	62.00	50.15	62.41	37.89
	UltraGen (AR+GPO)	59.61	84.33	34.89	61.22	57.23	69.01	45.44
Ablation	AR (Few Constraints)	48.25	74.09	22.41	60.10	38.38	46.00	30.76
	GPO	55.57	74.50	36.63	60.59	42.44	51.00	33.86
	AR+GPO (Random Sampling)	59.77	85.42	34.11	60.56	55.24	68.01	42.47
	AR+GPO (High Similarity)	59.44	83.22	35.65	60.85	55.45	66.05	44.85
	AR+GPO (Low Correlation)	58.91	83.59	34.23	60.00	54.47	65.22	43.71

Table 1: Performance scores for Llama-3.2-3B-Instruct models on the validation set under different evaluation conditions across FineWeb and Global splits.

Dataset (Domain)	Base	AR	AR+GPO
Emotion (Tweet Emotion)	28.25	**42.30**	38.65
Hillary (Tweet Stance)	55.93	45.76	**58.31**
AG-News (News Topic)	80.03	79.96	**83.28**
TREC (Question Type)	38.00	51.20	**51.40**
Average	50.55	54.81	**57.91**

Table 2: Performance comparison for data synthesis.

topic classification **(3) AG News** (Zhang et al., 2015), question type classification **(4) TREC** (Li and Roth, 2002).

For each dataset, we analyze the unique properties and paraphrase these properties as hard and soft attributes. Then using a uniform prompt tailored for each dataset, we generate 2,000 synthetic samples per dataset. These generated samples are then used to train a classifier, which is subsequently evaluated on the original test set of the dataset. This procedure allows for a fair comparison of model performance on synthetic data.

The results, summarized in Table 2, demonstrate the superior generalization ability of the AR+GPO model trained on the Global split. Notably, the AR+GPO model achieved the highest average score of 57.91 across the benchmarks, significantly outperforming both the base model and the AR models. While the AR model's performance stagnated (45.76, lower than the original one) on the Hillary benchmark, reflecting a focus on localized attributes, the AR+GPO model excelled with a score of 58.31, indicating its generalization and adaptability beyond localized training objectives.

4.4 Trade-Offs in EFCG

Figure 5 illustrates the interplay between BERTScore and CSR across different numbers of attributes from 10 to 50 for each model. As the figure shows, increasing the number of attributes presents a clear double-edged effect: while more attributes can enhance fine-grained control (e.g., higher F1 score) over the generated text, the added complexity makes it more difficult for the model to maintain high constraint adherence.

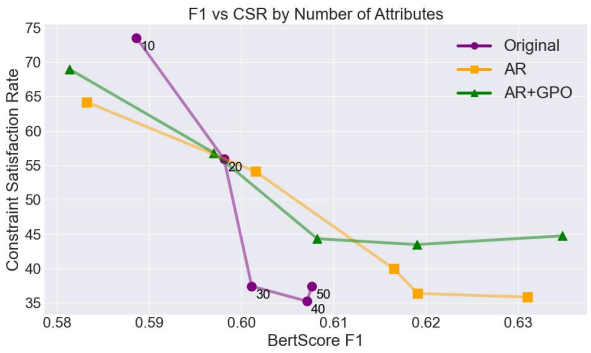

Figure 5: The Trade-off between F1 score and CSR. While BERTScore tends to improve with more attributes, CSR declines

Better Multi-Objective Alignment Under EFCG. When looking at the 30, 40, and 50 attribute conditions: AR+GPO consistently attains CSR values 5–10 points higher than the other two models without sacrificing F1. For example, at 50 attributes, AR+GPO's CSR (44.76%) is considerably above AR's (35.86%) and Original's (37.40%), while also delivering the highest F1 (0.6348 vs. 0.6310 for AR and 0.6076 for Original).

This pattern illustrates a more favorable trade-off for AR+GPO: it does not simply chase high BERTScore by ignoring constraints, nor does it force all constraints at the expense of overall text quality. Instead, AR+GPO's global optimization helps coordinate multiple constraints while retaining strong semantic alignment. In contrast, AR appears effective at moderate attribute counts but loses ground on CSR once the load goes beyond 30 attributes, and the Original model experiences an even steeper decline.

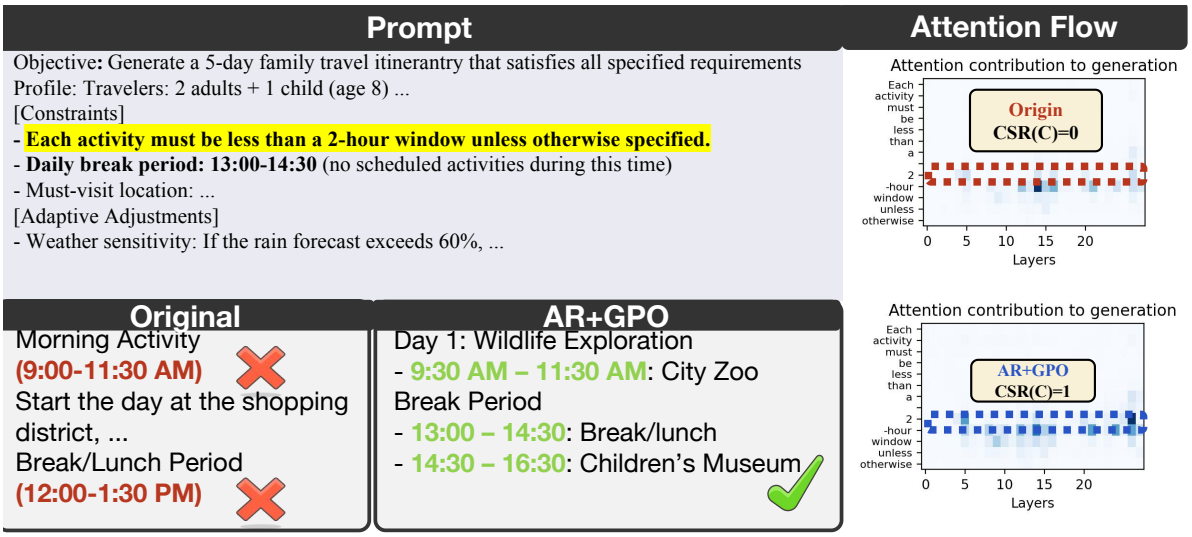

Figure 6: In a case study on travel itinerary generation, the attention flow illustrates improved constraint awareness in AR+GPO.

5 Analysis

5.1 UltraBench Mitigates Performance Degradation Across Different Positions

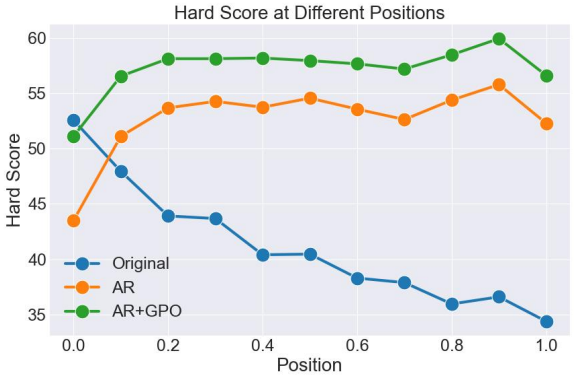

Figure 7: Hard score across different positions, showing that our approach (AR+GPO) effectively mitigates performance degradation.

Our method effectively mitigates the sensitivity to positional changes in hard attributes. As shown in Figure 3, baseline models such as Llama-3.1-8B-Instruct and Qwen2-7B-Instruct exhibit a significant drop in hard scores as the position increases, indicating a degradation in performance when hard attributes appear later in the input. In contrast, our approach significantly stabilizes performance across all positions (Figure 7). The introduction of AR already improves robustness compared to the original model, and the addition of GPO further enhances consistency, maintaining high hard scores even at later positions. This demonstrates that our approach effectively addresses the position sensitivity issue, ensuring more reliable model performance regardless of attribute placement.

5.2 A Real-world Travel Case

To further evaluate our approach, we analyze a real-world travel planning scenario where the itinerary must satisfy over thirty attributes. One crucial constraint is that *each activity should be less than two hours long*. The authors examined the response generated by three models. We observe that only the AR+GPO model consistently generates activities that adhere to this constraint, whereas the original model and AR model occasionally violate it. To gain deeper insights, we provide the user prompt along with a partially generated response (e.g., "14:00 PM - 1") and examine the attention flow distribution at this intermediate step. As illustrated in Figure 6, the AR+GPO model exhibits significantly higher attention weights on constraint-related tokens (e.g., "2"), suggesting that it effectively retains and incorporates constraint-relevant information during generation. In contrast, the original model's attention weights are relatively weak, indicating a lower degree of constraint awareness.

6 Conclusion

We proposed UltraGen, a two-stage framework for extremely fine-grained controllable generation. The Auto-Reconstruction stage trains LLMs to align with both soft and hard attributes, while Global Preference Optimization further enhances constraint satisfaction under diverse attribute combinations. Experiments on UltraBench demonstrate that UltraGen significantly improves both constraint adherence and text quality.

Limitations

While UltraGen demonstrates strong performance in handling extremely fine-grained controllable generation, several limitations remain. First, the set of hard attributes used in this work, though diverse and practical, primarily focuses on structural and keyword constraints; future work could explore more complex and domain-specific hard constraints to further stress-test model capabilities. Second, although our attribute correlation and diversity strategies reduce implausible combinations, ensuring absolute coherence across a large number of constraints remains an open challenge.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.

Fredrik Carlsson, Joey Öhman, Fangyu Liu, Severine Verlinden, Joakim Nivre, and Magnus Sahlgren. 2022. Fine-grained controllable text generation using non-residual prompting. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6837–6857.

Howard Chen, Huihan Li, Danqi Chen, and Karthik Narasimhan. 2022. Controllable text generation with language constraints. *arXiv preprint arXiv:2212.10466*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022. A distributional lens for multi-aspect controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1043, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

James Y Huang, Sailik Sengupta, Daniele Bonadiman, Yi-an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. 2024. Deal: Decoding-time alignment for large language models. *arXiv preprint arXiv:2402.06147*.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, Qun Liu, and Wei Wang. 2024. Learning to edit: Aligning llms with knowledge editing. *CoRR*, abs/2402.11905.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2023. Followbench: A multi-level fine-grained constraints following benchmark for large language models. *arXiv preprint arXiv:2310.20410*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*, volume 35, pages 27591–27609. Curran Associates, Inc.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, et al. 2023. Controlled decoding from language models. *arXiv preprint arXiv:2310.17022*.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2024. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36.

Qingyu Ren, Jie Zeng, Qianyu He, Jiaqing Liang, Yanghua Xiao, Weikang Zhou, Zeye Sun, and Fei Yu. 2025. Step-by-step mastery: Enhancing soft constraint following ability of large language models. *arXiv preprint arXiv:2501.04945*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. In *Proceedings of NAACL-HLT*, pages 3999–4004.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1361–1375, Online. Association for Computational Linguistics.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, et al. 2024. Benchmarking complex instruction-following with multiple constraints composition. *arXiv preprint arXiv:2407.03978*.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*.

Shunyu Yao, Howard Chen, Austin W Hanjie, Runzhe Yang, and Karthik Narasimhan. 2023. Collie: Systematic construction of constrained text generation tasks. *arXiv preprint arXiv:2307.08689*.

Tao Zhang, Yanjun Shen, Wenjing Luo, Yan Zhang, Hao Liang, Fan Yang, Mingan Lin, Yujing Qiao, Weipeng Chen, Bin Cui, et al. 2024. Cfbench: A comprehensive constraints-following benchmark for llms. *arXiv preprint arXiv:2408.01122*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023a. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023b. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.

Zhanhui Zhou, Jie Liu, Chao Yang, Jing Shao, Yu Liu, Xiangyu Yue, Wanli Ouyang, and Yu Qiao. 2023c. Beyond one-preference-for-all: Multi-objective direct preference optimization. *arXiv preprint arXiv:2310.03708*.

A Hard Attributes

The hard attributes employed in this study, as detailed in Table 3, comprise a set of verifiable instructions designed to enforce precise, programmatically assessable constraints on text generation. These attributes are categorized into four primary groups: (1) Keywords, which mandate the inclusion or frequency of specific terms (e.g., "Include {keyword1}" or "appear {N} times"); (2) Length Constraints, governing structural requirements such as paragraph count, word limits, or sentence boundaries; (3) Change Cases, enforcing syntactic rules like all-uppercase or all-lowercase formatting; and (4) Positional Directives, such as starting responses with predefined phrases. Each attribute is selected for its objective verifiability through rule-based checks while also reflecting common real-world application scenarios, such as compliance with stylistic guidelines or technical specifications. By anchoring the evaluation in these deterministic constraints, the framework guarantees rigorous assessment of model adherence to fine-grained requirements, aligning with the dataset's emphasis on combinatorial complexity and practical utility.

B Generalization to Unseen Attributes

We evaluate the models' ability to generalize to unseen, more challenging attributes, focusing on two types:

1. **Absolute Position of a Word**: The k-th ($k \leq 5$) word in the text must be A.
2. **Relative Position Between Two Words**: Word A must appear before word B.

We use text from the FineWeb validation set and extract 50 attributes per document, focusing on these two types of harder attributes. We then evaluate the three models on this benchmark. The results show that the original model achieves a score of 21.56, while auto-reconstruct slightly reduces performance to 20.79. However, incorporating GPO alongside AR improves generalization, yielding a score of 24.05, suggesting that GPO enhances the model's ability to handle these harder constraints.

C Dataset Statistics

In this section, we present detailed statistics of our dataset, including a comparison with existing datasets, quality control evaluation, and the composition of our multi-sources subset.

Comparison with Existing Datasets. Table 5 provides a comparison between our dataset and several representative constraint-based datasets, including IFeval (Zhou et al., 2023a), FollowBench (Jiang et al., 2023), CFBench (Zhang et al., 2024), and InFoBench (Qin et al., 2024). Our dataset distinguishes itself with a significantly larger number of samples (6,159) and a notably higher average number of attributes per instance (45.9). Unlike prior datasets, which primarily rely on either human annotations or simple constraints, our data features a rich combination of both hard and soft constraints, offering a more challenging and comprehensive benchmark. Importantly, our data is not synthesized, ensuring its alignment with real-world use cases.

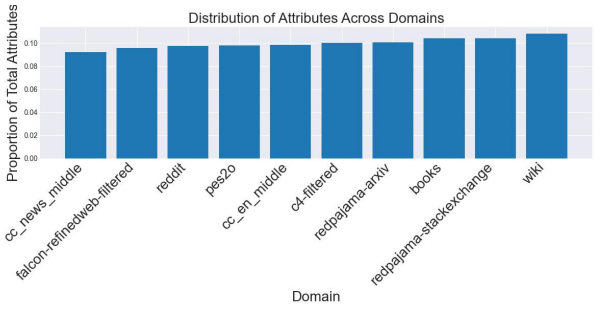

Figure 8: Proportion of Attributes Across Different Data Domains. The bar chart visualizes the relative contribution of each domain to the multi-sources subset, highlighting a balanced distribution across various sources such as web data, forums, papers, books, and Wikipedia.

Domain Composition in Multi-sources Subset Our multi-sources subset is constructed from a diverse range of data sources, encompassing web data, forums, academic papers, books, and Wikipedia. Figure 8 illustrates the proportion of attributes contributed by each domain, highlighting a balanced distribution across these categories. Table 6 further details the exact composition, showing that no single source overwhelmingly dominates, ensuring robustness and variety in downstream tasks.

Quality Control Metrics Maintaining data quality is critical for ensuring reliable evaluations. We assess the agreement rate (AR) between human annotators and the final dataset as a key metric. As summarized in Table 7, the FineWeb subset achieves an AR of 92.3%, while the multi-sources subset attains 88.7%. These high agreement rates reflect the robustness of our data curation process, confirming that both subsets align closely with hu-

Instruction Group	Instruction	Description
Keywords	Include Keywords	Include keywords {keyword1} in your response
Keywords	Keyword Frequency	In your response, the word word should appear {N} times.
Length Constraints	Number Paragraphs	Your response should contain {N} paragraphs. You separate paragraphs using \n \n
Length Constraints	Number Words	Answer with at least / around / at most {N} words.
Length Constraints	Number Sentences	Answer with at least / around / at most {N} sentences.
Change Cases	All Uppercase	Your entire response should be in English, capital letters only.
Change Cases	All Lowercase	Your entire response should be in English, and in all lowercase letters. No capital letters are allowed.
Start with	Start With	Finish your response with this exact phrase {end_phrase}. No other words should follow this phrase.

Table 3: The list of 8 verifiable instructions, with brief descriptions. We use these instructions because we think they are either easy to verify or common in real-world applications.

Subset	Train Size	Val Size	Avg Length (words)	Soft Attrs (per sample)	Hard Attrs (per sample)	Total Attrs (per sample)
FineWeb (Local)	6,159	200	361.6	7.80	38.10	45.90
Multi-sources (Global)	1600	400	-	5.1	24.8	29.9

Table 4: UltraBench Dataset Composition

Method	Data Quality			
	Nums.	Cons.	Avg Attr.	Synt.
IFeval (Zhou et al., 2023a)	541	H	1.54	✓
FollowBench (Jiang et al., 2023)	820	H/S	3.0	✓
CFBench (Zhang et al., 2024)	1000	H/S	4.24	✗
InFoBench (Qin et al., 2024)	500	H/S	4.5	✗
UltraGen (FineWeb Split)	6159	H/S	**45.9**	✗
UltraGen (Multi-source Split)	1600	H/S	**29.9**	✗

Table 5: Detailed comparison of relevant works. Ours represents our dataset construction approach. 'Nums.', 'Cons.', 'Avg Attr.', and 'Synt.' denote the number of samples, constraint types, average number of attributes, and whether the data is synthesized.

Category	Domain	Percentage
Web Data	CC News (Middle)	9.24%
	Falcon RefinedWeb Filtered	9.59%
	CC EN (Middle)	9.86%
	C4 Filtered	10.03%
Forum	Reddit	9.75%
	StackExchange (RedPajama)	10.42%
Papers	arXiv (RedPajama)	10.05%
	Pile-Extracted Scientific Open (PES2O)	9.79%
Books	Books	10.42%
Wikipedia	Wikipedia	10.84%

Table 6: Distribution of data sources in the multi-sources subset by category. The data sources are grouped into major categories: Web Data, Forums, Papers, Books, and Wikipedia. Percentages represent the proportion of each domain within the subset.

man judgment.

Metric	FineWeb Subset	Multi-sources Subset
Agreement Rate (AR)	97%	96%

Table 7: Quality Control Metrics

D DPO data quality

High Correlation:
- Thought-provoking narrative with a call to action
- Author's Name and Location Identifier: The text begins with the name S. TEITELBAUM followed by a location ST. JOHNS, FL.
- Engaging Headline: The title captures the reader's attention by listing 5 Reasons for a specific action.

Low Correlation:
- AI Leadership: The partnership aims to position Singapore as a leader in AI within healthcare
- Focus on Competitive Standards: The passage stresses the competitiveness of FAU's admissions process

Table 8: Correlation Examples

In this section, we showcase some examples sampled by our global selection strategy.

High Correlation Our attribute correlation modeling step aims to select semantically coherent and mutually reinforcing attributes during GPO training. This process effectively groups attributes that frequently co-occur in natural text, leading to the selection of high-quality attribute combinations.

Low Similarity While high correlation ensures that attributes are semantically aligned, it is equally important to maintain attribute diversity to prevent redundancy and overfitting. Our global selection strategy aims to minimize the presence of highly

Low Similarity:
- Focus on Natural Ingredients: Emphasizes the importance of natural ingredients
- Protein-rich for Satiety and Muscle Growth: The high protein content in buffalo milk helps increase satiety

High Similarity:
- Health focus: The text emphasizes overall health benefits
- Detailed explanation for each benefit: Each health benefit mentioned is followed by an explanation or reasoning

Table 9: Similarity Examples

similar attributes within the same prompt. For instance, attributes like *"Engaging Headline"* and *"Attention-Grabbing Title"* convey nearly identical meanings and offer little additional training value when paired together. By prioritizing low-similarity combinations, we encourage the model to generalize across a broader range of attribute expressions, improving its adaptability to diverse prompts.

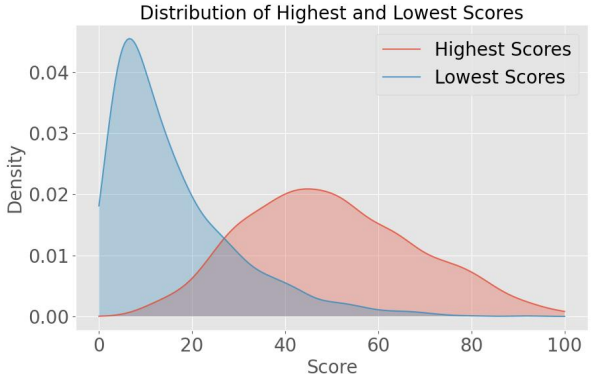

Figure 9: Score distributions of chosen and rejected data.

To further illustrate the effectiveness of our sampling strategy in the GPO stage, we present several representative cases selected by our attribute-based sampling approach. These examples demonstrate the diversity and coverage achieved through our strategy, highlighting both common and edge-case attribute combinations.

E Prompts Used in This Study

We employ three distinct prompts to support different stages of our EFCG pipeline: Decomposition, Judging, and Generation.

Decomposition Instruction. This prompt is used to extract a set of soft attributes from a given text. The goal is to decompose the paragraph into its most defining characteristics, capturing both stylistic and semantic elements. Models are instructed to focus on identifying specific, explicit features of the text rather than relying on generic descriptions or subjective interpretations. Attributes must reflect the unique aspects of the text and be grounded in the content.

Requirements
For the following paragraph, propose attributes that capture its overall characteristics. Focus on what makes this text unique and distinctive, rather than using predefined categories. Your analysis should:
- Identify the most prominent and defining features of the text
- Use clear, specific descriptions rather than vague terms
- Base attributes solely on what is explicitly present in the text
- Describe each attribute with enough detail to be meaningful

Avoid:
- Overly broad or generic attributes
- Speculative interpretations
- Attributes not clearly supported by the text
- Complex or academic jargon

Output each attribute on a separate line, separated by a single newline, with no line breaks within each attribute.
Now, analyze the following paragraph and summarize its key attributes:
Text
{text}
Attributes

Table 10: Decompose Prompt

You are a binary evaluator. Given a text and several attributes, determine if the text fulfills each attribute.
Your task is simple:
- Score 0 if the text does NOT fulfill the attribute or the attribute is not directly mentioned
- Score 1 if and only if the text directly fulfills the attribute

Text to evaluate:
{text}
Attributes to evaluate:
{attributes}
Provide exactly {num_attributes} scores, one per line, using this format:
Score: 0 or 1
- Scores should correspond to attributes in order
- Only provide the scores, no additional explanation

Table 11: Judge Prompt

Judge Instruction. This prompt serves as a binary evaluation guideline to determine whether a generated text satisfies a given set of attributes. Evaluators are asked to assess each attribute independently, assigning a score of 1 if the text explicitly fulfills the attribute and 0 otherwise. The evaluation is strict, requiring the text to directly align with the specified attribute for a positive score.

You are an expert at generating text that matches given attributes. Your task is to generate a text that satisfies as many of the provided attributes as possible. ### Hard Attributes: {hard_attributes} ### Soft Attributes: {soft_attributes}

Table 12: Generation Prompt

Generation Instruction. This prompt is used to instruct the language model to generate a piece of text that aligns with a provided set of hard constraints and soft attributes. Hard attributes typically represent structural or factual constraints (e.g., budget, schedule), while soft attributes reflect stylistic or semantic preferences (e.g., tone, vividness). The model is guided to generate text that adheres to as many of these attributes as possible, balancing the satisfaction of both hard and soft constraints.

F The Complete Case Study

The travel planner case study exemplifies the practical usefulness of EFCG in handling complex, multi-faceted requirements. As shown in Table 13, generating a 5-day travel itinerary involves satisfying a diverse set of hard attributes (e.g., budget limits, time scheduling, location constraints) alongside soft attributes (e.g., tone, emotion, visual details), while also adapting to real-time factors like weather and physical endurance. Such a task necessitates precise control over both hard and soft constraints, making it a natural testbed for evaluating EFCG systems.

Objective:
Generate a 5-day family travel itinerantry that satisfies all specified requirements while adhering to highly fine-grained constraints. The generated itinerary should balance real-time adaptability, strict hard attributes, and semantic soft attributes.
User Profile:
- Travelers: 2 adults + 1 child (age 8)
- Budget: <= $300/day (total $1,500 for the trip)
- Activity Balance: 70% educational/cultural experiences, 20% relaxation, 10% family-friendly shopping.
Hard Attributes:
- Activity Scheduling:
  - Each activity must have a defined start and end time, ensuring there is no overlap between activities.
  - A break period from 13:00-14:30 is mandatory daily.
  - Each activity must fit within a 2-hour window unless otherwise specified.
- Budget Requirements:
  - Each day's total cost (including transportation, food, and activities) must not exceed $300.
  - Transportation is limited to metro and walking only, with a maximum of 3 metro rides per day.
- Location Constraints:
  - Must-visit locations: City Zoo (Day 1) and Science Museum (Day 3).
  - Activities must occur in geographically adjacent areas to minimize walking distance.
- Keyword Requirements:
  - Each day's description must include specific keywords. For example:
  - Day 1: "wildlife," "exploration," and "interactive learning."
  - Day 3: "science," "innovation," and "hands-on exhibits."
- Structure Constraints:
  - Each day's itinerary must consist of 4 sections:
    - Morning activity
    - Break/lunch period
    - Afternoon activity
    - Evening summary (limited to 50 words)
Soft Attributes
- Tone and Emotion:
  - Day 1: Use a tone that conveys "excitement and discovery."
  - Day 3: Use a tone that conveys "curiosity and wonder."
- Language Style:
  - Use descriptive, vivid, and family-friendly language throughout.
  - Include at least one metaphor or simile per day (e.g., "The Science Museum felt like stepping into the future!").
- Visual Details:
  - Each activity must include specific sensory details (e.g., "the bright colors of the parrots at the zoo" or "the tinkling sound of water fountains at the park").
- Adaptive Adjustments (Real-time Constraints):
  - Weather Sensitivity:
    - If the rain forecast exceeds 60%, replace outdoor activities with indoor alternatives while keeping the overall tone and keywords intact.
  - Physical Endurance:
    - If a day's total walking distance exceeds 10 kilometers, the next day's activities must reduce walking by 30%.
  - Health Responsiveness:
    - If a health-related issue arises (e.g., fatigue or illness), adjust the itinerary dynamically to:
    - Reduce activity duration to half.
    - Substitute the activity with a more relaxing or passive option.

Table 13: The complete travel planner case study.