MAGICGEN: A UNIVERSAL MULTIMODAL DATA SYNTHESIS AGENT FOR DOMAIN-SPECIFIC VISION-LANGUAGE MODEL TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision–language models (VLMs) need large, domain-aligned multimodal data, yet high-quality data collection is costly and slow, especially in specialized domains with privacy, expertise, and distribution-shift constraints. Current synthesis methods are narrow, labor-intensive, or lack rigorous quality control, yielding brittle pipelines and noisy supervision. We introduce MagicGen, a universal agent that composes end-to-end, domain-specific data pipelines from natural-language prompts. Using unified interfaces, MagicGen selects and chains tools for image synthesis, text generation, augmentation, and modality transformation, enabling modular and scalable composition. The agent is trained with hybrid supervision: expert-authored reference pipelines plus LLM-generated candidates iteratively verified by humans for robust cross-domain generalization. We also propose an automated hierarchical evaluation pipeline: Image Validation (aesthetic + technical metrics) and Annotation Validation (multi-model discriminator with iterative decisions) for reliable quality control. Across diverse VLM tuning scenarios, MagicGen boosts data quality, reduces manual effort, and accelerates scalable dataset construction. It outperforms strong baselines on downstream tasks with less human oversight, and ablations confirm the importance of curated tool modularity, hierarchical evaluation, and hybrid training.

1 INTRODUCTION

Recent years have witnessed the rapid advancement of **vision-language models (VLMs)**, which have demonstrated remarkable capabilities across a wide range of multimodal tasks, including image captioning Stefanini et al. (2022), visual question answering de Faria et al. (2023), and visual grounding Rasheed et al. (2024). In particular, recent advanced large-scale VLMs, characterized by billions of parameters and trained on massive, diverse datasets, have achieved impressive results across nearly all standard multimodal benchmarks Wang et al. (2024b); Chen et al. (2024); OpenAI (2023). While state-of-the-art large-scale VLMs exhibit strong performance and generalization capabilities on standard benchmarks, their effectiveness in many specialized, real-world vertical domains remains limited and often fails to meet practical requirements Choi et al. (2024). For example, in medical image analysis Nath et al. (2025); Li et al. (2023a); Yang et al. (2024), remote sensing interpretation Kuckreja et al. (2024); Li et al. (2024); Zhang et al. (2024) and industrial applications Bhatia et al. (2024), general-purpose VLMs often fail to capture fine-grained domain knowledge, semantic nuances, or rare concepts that are crucial for downstream applications.

To bridge this gap, two prevalent strategies have been explored: (1) integrating domain-specific data during pretraining to tailor model representations toward the target domain Nath et al. (2025); Zhong et al. (2025); and (2) **domain specifc post-training** of VLMs on curated datasets for specific vertical tasks Cheng et al. (2024); Li et al. (2025). Nevertheless, both strategies fundamentally rely on the availability of large-scale, high-quality, domain-specific multimodal datasets. However, in practice, the acquisition of such data is often prohibitively challenging. While several efforts have been made to construct domain-specific datasets aimed at training expert models Bossard et al. (2014); Min et al. (2020; 2023), these datasets typically require substantial investment in domain experts for annotation, resulting in relatively small dataset sizes compared to widely used unsupervised datasets Oquab et al. (2023); Schuhmann et al. (2022). Recently, generative models—such

as diffusion models Ho et al. (2020); Nichol & Dhariwal (2021); Lugmayr et al. (2022) and large language models Guo et al. (2025); Yang et al. (2025)—have emerged as powerful tools for synthesizing large-scale, diverse multimodal data. These models have demonstrated impressive capabilities in generating high-quality images or texts, offering a promising solution to the data scarcity problem in vertical domains Cheng et al. (2024); Wu et al. (2023); Rombach et al. (2022). However, ensuring the relevance, accuracy, and domain-specific fidelity of the synthesized data raise another substantial challenge. In particular, the process of generating domain-specific data that faithfully captures the intricate characteristics, rare concepts, and semantic nuances unique to specialized fields often necessitates meticulously engineered data generation pipelines. Such pipelines typically require deep domain expertise and extensive manual intervention to define task-specific prompts, quality filters, and validation strategies, which significantly increases development costs and suffer from limited generalization, scalability, and flexibility. This highlights an urgent need for automated, scalable, and generalizable frameworks that can synthesize high-quality domain-specific multimodal data with minimal human intervention.

LLM as automated multimodal data synthesis agent. Large language models (LLMs) have recently demonstrated remarkable capabilities in a wide range of tasks, including in-context learning, complex planning, and adaptive tool-use Schick et al. (2023). These advancements have empowered LLMs to function as autonomous agents, orchestrating intricate workflows and facilitating intelligent decision-making across diverse domains. Building upon these breakthroughs in LLM-based tool-learning and automated agent frameworks Shi et al. (2025); Wang et al. (2024a); Xi et al. (2025), we introduce a novel and unified framework, MagicGen, which addresses the critical challenge of domain-specific multimodal data generation for VLMs in a general, efficient, and automated manner. MagicGen is designed as a general-purpose agent that can, given a textual description of a domain-specific data synthesis task, **automatically compose a multimodal data generation pipeline** by selecting, configuring, and chaining together appropriate tools and models from a multimodal curated toolkit. Specifically, MagicGen incorporates the following core components:

- **Curated Multimodal Toolset:** We collect and encapsulate a diverse set of state-of-the-art tools and models for image synthesis, text generation, data augmentation, and modality transformation, representing each as callable modules with unified interfaces.

- **Agent Training via Synthetic and Human-Validated Pipelines:** We first manually construct a set of reference pipelines for several representative domains. Leveraging the reasoning and composition abilities of LLMs, we then generate additional candidate pipelines for new domains, which are subsequently verified and refined by human experts. The resulting high-quality dataset is leveraged to facilitate the systematic training of our agent, empowering it to generalize pipeline composition strategies across a broader range of domains.

- **Automated Hierarchical Evaluation Pipeline:** We propose a systematic and scientific automated data evaluation pipeline, which consists of two core components: Image Validation and Annotation Validation. Image validation integrates both aesthetic and technical quality scores to achieve a comprehensive assessment of generated images. Annotation validation employs a multi-VLMs discriminator approach with an iterative decision-making mechanism, effectively enhancing the accuracy and robustness of annotated data. This pipeline significantly reduces manual intervention, improves data filtering efficiency, and provides a reliable foundation for large-scale synthetic data evaluation.

To validate the effectiveness and generality of MagicGen, we conduct extensive experiments on three distinct domain-specific tasks. In each scenario, MagicGen is able to autonomously generate specialized multimodal data synthesis pipelines, which are then used to produce high-quality synthetic datasets. Models trained on these datasets achieve **significant performance improvements** over strong baselines, demonstrating the utility and scalability of our approach. We believe MagicGen opens new avenues for automated, scalable, and domain-adaptive data generation, serving as a valuable foundation for future vision-language research and applications.

2 Related Works

2.1 Data Synthesis

With the rapid advancements of large language models (LLMs), the use of LLMs for textual data synthesis has become a highly promising area of research. Recent studies demonstrate that LLMs are capable of generating fluent and human-like text, positioning LLM-generated synthetic text as a viable substitute or supplement to human-annotated data Hartvigsen et al. (2022); Sahu et al. (2022); Ye et al. (2022); Tang et al. (2023); Gao et al. (2022). The extensive pretraining of LLMs allows them to acquire broad linguistic and factual knowledge Kim et al. (2022); Ding et al. (2022), resulting in high-fidelity and contextually rich data generation. Moreover, the instruction-following capabilities of LLMs provide fine-grained control and adaptability in the generation process, enabling the creation of task-specific and customizable text dataset Zhao et al. (2024).

In addition to textual data, image data synthesis has witnessed remarkable progress, particularly with the advent of diffusion-based generative models such as Stable Diffusion (SD) Ho et al. (2020); Nichol & Dhariwal (2021); Lugmayr et al. (2022). Diffusion models employ a forward process that incrementally adds noise to the data and a learned reverse process that reconstructs images from noise, enabling the generation of high-resolution, diverse, and photorealistic images. Compared to traditional generative approaches such as Generative Adversarial Networks (GANs) Goodfellow et al. (2014), diffusion models demonstrate superior performance in terms of sample diversity and fidelity. To enhance the controllability of image generation, techniques such as ControlNet Zhang et al. (2023) and IPAdapter Ye et al. (2023) enable the incorporation of additional conditioning signals (e.g., edge maps, pose, style) to guide the synthesis process, thereby allowing for more precise and semantically meaningful control over the generated content. Despite these advances, current methods are still challenged by the need for fine-grained and flexible control, especially when generating complex scenes or aligning with detailed textual descriptions.

2.2 LLM-based Agent

Recent advances in large language models (LLMs) have paved the way for the development of highly autonomous agents, capable of sophisticated reasoning, planning, and interactive decision-making. Driven by carefully designed prompts, LLM-based agents are able to decompose complex tasks into manageable subgoals Khot et al. (2022), systematically reason through each component, and explore alternative solution paths to optimize task performance. Beyond direct problem-solving, these agents exhibit the ability to dynamically select, sequence, and utilize appropriate tools to address multifaceted challenges in diverse, real-world environments Li et al. (2023b); Ruan et al. (2023); Gao et al. (2023). Schick et al. (2023) first formalized the decision criterion for when to invoke tools, suggesting that tool use is only necessary if the LLM cannot directly answer correctly. Du et al. further extended this view by noting that agents should first evaluate whether available tools are suitable and cost-effective for a given task. In simple tasks, the overhead of tool invocation may outweigh its benefits, while in complex scenarios, effective tool orchestration becomes essential. Tool retrieval involves decomposing tasks into subtasks and selecting the most appropriate tools. Qin et al.Qin et al. (2023) describe this as structured scheduling within the toolset, while others highlight the need for comparative selection across and within tool categories, especially as toolsets evolve Gao et al. (2024); Gou et al. (2023).

3 Automated Data Synthesis Pipeline Agent

We introduce MagicGen, an automated data synthesis pipeline agent designed to systematically construct domain-specific multimodal data generation workflows with minimal human intervention. MagicGen is built upon Qwen3-32B (Yang et al., 2025), augmented with reasoning capabilities and access to a curated set of multimodal processing tools. As shown in Figure 1, given a vertical domain specification—comprising textual descriptions, MagicGen performs a multi-stage reasoning process to:

- Analyze the domain-specific data requirements and identify potential modalities and data generation strategies.

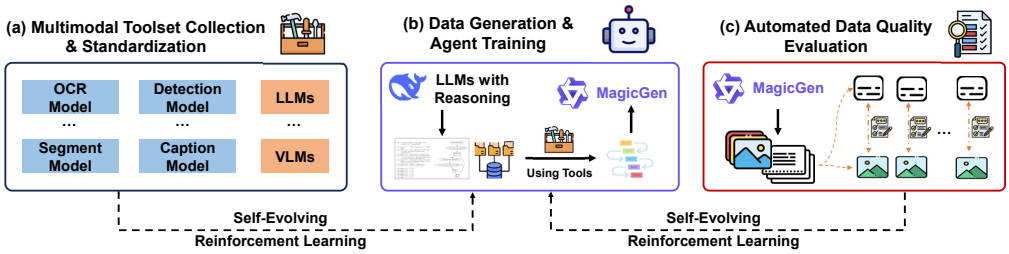


Figure 1: Overview of MagicGen: (a) collection and standardization of multimodal toolsets, (b) data generation and agent training with reasoning and tool use, and (c) automated data quality evaluation; all stages are linked via self-evolving reinforcement learning.

- Formulate structured pipeline blueprints that outline the sequence of operations required to synthesize the target data.
- Select and configure appropriate multimodal tools from the provided toolset, chaining them into executable workflows that can directly produce high-quality multimodal datasets.

Unlike conventional rule-based pipeline design, MagicGen autonomously explores multiple synthesis strategies, evaluates their feasibility based on tool capabilities, and determines the optimal pipeline structure to maximize data diversity, fidelity, and scalability. The resulting pipelines can be instantiated into fully operational systems that generate large-scale multimodal datasets tailored to the target domain, significantly reducing the need for manual design and integration efforts.

In this section, we begin by examining the key components that form the basis of MagicGen, providing the groundwork for the developments discussed in the following sections.

3.1 Multimodal Toolset Collection and Standardization

Our automated data synthesis pipeline agent begins with the systematic collection and categorization of state-of-the-art (SOTA) models spanning diverse modalities. To ensure comprehensive coverage and robustness, we leveraged multiple sources, such as open-source repositories, recent benchmark leaderboards, and peer-reviewed publications. Each candidate model was evaluated based on criteria such as modality coverage, architectural novelty, performance metrics, and compatibility with downstream tasks.

To facilitate seamless integration within our multimodal data synthesis pipeline, we standardized the input and output interfaces of each selected model. This involved designing a unified schema that abstracts away modality-specific intricacies, enabling each model to be encapsulated as a modular node. Each node adheres to a well-defined API specification, supporting batch processing, error handling, and extensibility for future updates.

To ensure reliability and maintainability, we also established a versioning and metadata management protocol for all nodes, recording provenance, performance benchmarks, and supported modalities. This metadata facilitates efficient node discovery, dependency tracking, and future node replacement or upgrades.

Given a list of tools $\{t_i\}_{i=1}^n$, we manually encapsulate the tools into a dictionary

$$\{t_i : (I_i, O_i, D_i)\}_{i=1}^n,$$

with each tool t_i assigned with standardized input I_i, standardized output O_i and a general description about the tool node function D_i, forming a comprehensive and extensible toolkit. This toolkit serves as the foundation for subsequent data synthesis pipeline generation workflows, supporting flexible composition and dynamic orchestration of data synthesis pipelines.

3.2 Pipeline Data Generation

To acquire a sufficiently diverse vertical domain data for agent training, we adopt a hierarchical domain expansion framework inspired by Chen et al. (2025). Specifically, we systematically construct

100,000 synthetic vertical domains, each accompanied by a comprehensive and semantically rich domain description.

In our early exploration, we experimented with multiple strategies for constructing vertical-domain-specific data synthesis pipelines. One naive approach was to instruct a large-scale LLM to directly output purely textual descriptions of data synthesis workflows given the domain description. However, it lacked the structural precision required for downstream automated execution, and the descriptions were often ambiguous or underspecified with respect to tool invocation sequences.

We also attempted a single-step generation approach, where GPT-4o OpenAI. (2024) was prompted to directly produce executable pipeline specifications in one pass. However, due to the necessity of embedding a large amount of multimodal toolset information $\{t_i\}_{i=1}^{n}$ into a single prompt, we observed a noticeable degradation in both the logical coherence and creativity of the generated pipelines. The overload of tool-specific constraints appeared to bias the model towards syntactic conformity at the expense of innovative and domain-appropriate synthesis strategies. Consequently, the overall quality and domain alignment of the resulting pipelines were suboptimal.

Based on these observations, we ultimately adopted a two-stage generation strategy leveraging GPT-4o. In the first stage, GPT-4o is provided with the detailed domain description alongside a carefully designed few-shot prompt. Conditioned on this input, GPT-4o produces a structured textual analysis of potential data generation strategies and outlines candidate data synthesis workflows specific to the target domain. This stage serves to decompose the problem space and identify feasible synthesis approaches in a human-interpretable manner. In the second stage, the set of multimodal tools $\{t_i\}_{i=1}^{n}$, curated in the previous section, is provided to GPT-4o. Conditioned on the Stage 1 textual analysis, GPT-4o selects the most optimal tool nodes and assembles them into several coherent synthesis pipelines. To standardize the representation of the generated pipelines data, we formalize the selected multimodal tools as graph nodes, with directed edges denoting the data flow between tools. GPT-4o is instructed to produce Python-based *Graphviz* code that encodes each pipeline as a directed acyclic graph (DAG). The graph is anchored with a single input node, `Domain Description`, and terminates with two designated output nodes, `Image` and `Annotation`. This standardized graph representation not only facilitates automated visualization but also ensures consistency of our pipelines data.

As part of the post-generation quality assurance, we implement a rigorous data curation procedure to ensure the validity and consistency of the synthesized pipeline data. We first perform graph validity verification. Specifically, each generated pipeline's *Graphviz* code is executed within a controlled Python environment to verify its syntactic correctness and its ability to produce valid DAGs visualization. Pipelines whose *Graphviz* code fails to compile or render valid DAGs are automatically discarded. In addition, we conduct toolset compliance checking to enforce semantic consistency between the generated pipelines and the predefined multimodal toolset $\{t_i\}_{i=1}^{n}$. A rule-based verification module systematically inspects each pipeline to ensure that every tool node instantiated in the graph is drawn exclusively from the authorized toolset. Pipelines containing out-of-scope or undefined tool nodes are eliminated from the dataset. This two-tiered curation process ensures that the retained pipeline data is both structurally executable and semantically aligned with the intended multimodal capabilities.

3.3 Agent Training

Following the aforementioned two-tiered curation process, we obtain a final corpus of approximately 30,000 high-quality pipeline instances. This curated dataset is then used to fine-tune our LLM-based agent. The base model of our agent is Qwen3-32B, a large-scale language model equipped with an explicit thinking mode that separates internal reasoning from the final user-facing output. To fully exploit this architectural feature, we design our training data to explicitly align the pipeline's textual analytical reasoning with the model's thinking channel, while placing the corresponding *Graphviz* code in the response channel. As shown in Listing 1, for each curated pipeline, the structured textual analysis generated in Stage 1—capturing the rationale behind tool selection, pipeline topology, and expected data flow—is encapsulated within the model's thinking segment. The Stage 2 output, namely the Python-based *Graphviz* code representing the final DAGs, is placed in the response segment.

Listing 1: Training data structure

```
<|im_start|>user
Training prompt
<|im_end|>
<|im_start|>assistant
<think>
Textual Analysis of Data Synthesis Strategy
</think>
Graphviz Code for Finalized DAG Pipeline
<|im_end|>
```

We adopt a supervised fine-tuning (SFT) paradigm, minimizing the autoregressive loss over both reasoning and response sequences. Training is conducted on 16 NVIDIA A800 GPUs (80GB), using a global batch size of 512 and a cosine learning rate schedule with a peak learning rate of 1×10^{-5}. This SFT process enables the model to acquire accurate reasoning patterns, standardized pipeline structural formats, and domain-specific data synthesis knowledge. The resulting SFT model is denoted as **MagicGen-sft**.

To rigorously verify the effectiveness of **MagicGen-sft**, we construct a proprietary evaluation benchmark derived from the training corpus. Specifically, from the 10,000 domains in the corpus, we select 100 domains that are both highly representative and exhibit substantial diversity. For each selected domain, an initial pipeline is generated by the agent, followed by careful refinement and augmentation by expert human annotators to establish high-quality ground-truth annotations. For evaluation, we focus exclusively on the structural fidelity of the generated pipelines, as accurately reproducing the topological structure is critical for ensuring correct execution in downstream applications. We employ the *Graph Edit Distance* (GED) Cheng et al. (2025), which quantifies the minimum number of node and edge operations (insertion or deletion) required to transform one graph into another. GED is normalized to the range $[0, 1]$, and the structural similarity is defined as:

$$S_{\text{GED}}(G_1, G_2) = 1 - \frac{\text{GED}(G_1, G_2)}{\max(|V_1| + |E_1|,\ |V_2| + |E_2|)},$$

where $|V|$ and $|E|$ denote the number of nodes and edges, respectively. A value closer to 1 indicates higher structural alignment. In practice, for each domain, all possible pairings between predicted and ground-truth pipelines are enumerated, and the pairing with the highest S_{GED} is taken as the final evaluation score. This structure-oriented evaluation ensures that the agent is rigorously assessed on its ability to produce pipelines with accurate and executable topologies, thereby directly guaranteeing the correctness of the generated pipelines.

3.4 Automated Data Quality Evaluation

Given the substantial volume of data produced by synthesis pipelines, manual inspection becomes infeasible. Our evaluation system consists of two complementary components: **Image Validation** and **Annotation Validation**.

Image Validation We jointly assess both model-based **aesthetic scores** and **image quality scores**. The aesthetic score primarily addresses subjective dimensions such as composition, color, and creativity, reflecting the visual appeal and artistic value of an image. In contrast, the image quality score focuses on objective technical indicators, including clarity, noise, distortion, and resolution, to evaluate the technical merit of the image. Domain-specific thresholds are defined for both metrics; images exceeding both thresholds are labeled as high-quality. For domains where aesthetic evaluation is irrelevant or quality metrics may be misleading (e.g., vertical domains that requires blurred or mosaic content), the corresponding score is disabled. The *overall pipeline quality* is defined as the proportion of high-quality images to the total generated, and is only computed for pipelines involving image generation nodes. For open-source datasets with directly collected images, we assume all are high-quality as a baseline.

Annotation Validation To address hallucinations and textual inaccuracies in VLM-generated annotations, a common approach is to use a single VLM to judge annotation quality. However, this method is often unreliable due to model biases and domain-specific blind spots. To overcome these

limitations, we introduce a **multi-VLMs discrimination** mechanism as a core evaluation component applied throughout the annotation validation process.

We select several top-ranked VLMs from the OpenCompass leaderboard Duan et al. (2024) and group them in pairs according to ascending average score. Each group acts as a discriminator unit in a weak-to-strong evaluation cascade: if both models in a group agree on the judgment, the decision is accepted; otherwise, the next stronger group is invoked. Lack of consensus after the final group results in rejection. This ensemble-based approach mitigates individual model biases, increases robustness, and is invoked in every validation step.

With our multi-VLMs discriminator in place, we employ a hierarchical evaluation pipeline to validate the correctness of annotations. We begin our annotation validation by locating and extracting content enclosed within domain-specific special tokens (e.g., `<box>...</box>`, `<ref>...</ref>`) using a pre-defined token inventory. Extracted segments are normalized into a unified format and evaluated for semantic alignment. Processed tokens are then removed to avoid interference in subsequent checks, and outputs with format inconsistencies or mismatched token counts are filtered out. For the remaining annotations, structured outputs (e.g., dictionaries, lists, JSON) are decomposed into independent items. Each item is classified as descriptive or non-descriptive: non-descriptive pairs (question–answer) are directly judged by the multi-VLM mechanism, whereas descriptive responses are segmented into smaller units for independent evaluation. An annotation is considered high-quality only if all corresponding units pass the multi-VLM discrimination. In our practical deployment, the results of this evaluation system are used as a strict filter: only pipelines whose generated data meet the defined quality thresholds are retained, while those producing low-quality data are discarded to ensure that all synthesized data entering our corpus are of high fidelity and reliability.

3.5 Self-Evolving Reinforcement Learning

We initialize our agent (**MagicGen-sft**) via supervised fine-tuning on Qwen3-32B, which equips the agent with the basic ability of generating feasible data synthesis workflows. To further enhance its capability to produce rational and creative pipelines, we propose a self-evolve framework where we apply Direct Preference Optimization (DPO) to continuously improve the rationality and creativity of our automated pipeline agent.

The self-evolution process proceeds in iterative cycles. At iteration t, we first collect hard vertical domain samples that perform poorly in real-world deployment of our agent obtained from the previous iteration M_{t-1} (initialized using **MagicGen-sft** at the first iteration), mainly based on the results of our automated evaluation system. Subsequently, we use these collected hard vertical domains as seed and employ GPT-4o to expand these vertical domains to a larger test dataset $\{x_i\}_{i=1}^{n}$. Then for each test data x_i, we sample multiple responses $\{y_j^i\}_{j=1}^{m}$ from M_{t-1}:

$$y_j^i \sim M_{t-1}(\cdot \mid x_i)$$

Each candidate is evaluated along two dimensions: **Rationality**—assessing logical soundness, coherence, and task relevance via GPT-4o and **Diversity**—measuring structural variability of the generated pipelines. The two scores are averaged to determine the preferred and less-preferred responses (y_w^i, y_l^i). With these preference pairs, DPO is applied to update the model from M_{t-1} to M_t. By repeating this loop, the agent incrementally learns from its most challenging cases, achieving sustained gains in functional correctness, diversity, and overall data synthesis quality.

4 Experiments

In this section, we present comprehensive experiments to evaluate the effectiveness of our automated data synthesis pipeline agent. Specifically, our auto-pipeline agent is deployed across three distinct domains, enabling the generation of diverse and domain-specific data synthesis pipelines. To ensure a fair and robust evaluation, we perform unified fine-tuning on two representative models, InternVL3-2B and InternVL3-8B (Zhu et al., 2025), using the synthesized data generated by the pipeline. The experimental results provide insights into the generalizability and effectiveness of our approach, as well as its impact on downstream multimodal domain-specific tasks.

4.1 Data Synthesis Pipelines Generation

To evaluate the versatility of our automated data synthesis pipeline agent, we select three diverse and challenging domains: food composition recognition, small object detection, and OCR-based information extraction. These tasks differ substantially in visual patterns, annotation demands, and downstream objectives, covering a broad spectrum of multimodal data synthesis challenges.

4.1.1 Food Composition Recognition

We first evaluate our approach on the challenging visual question answering (VQA) task of *food composition recognition*, which aims to determine the constituent ingredients or food items present in a given image. To address the scarcity and annotation cost of fine-grained food composition datasets, we leverage our proposed data synthesis pipelines agents to automatically generate large-scale, high-quality training data.

The data synthesis pipeline generated by our MagicGen agent is illustrated in Figure 2. Specifically, it employs Qwen3-32B (Yang et al., 2025) to randomly generate plausible food composition annotations. These synthetic annotations are then used as prompts for the state-of-the-art text-to-image generation model FLUX (Batifol et al., 2025), which synthesizes corresponding food images.

```
dot = Digraph(comment='Food Composition
    Recognition', format="png")

dot.node('1-1', 'Domain Description', shape='
    box')
dot.node('1-2', 'Text Generation Model-LLM',
    tooltip='Generate possible food
    ingredients')
dot.node('1-3', 'Annotation', shape='box')
dot.node('1-4', 'Text Generation Model-LLM',
    tooltip='Generate a detailed description
    of food with given ingredients')
dot.node('1-5', 'Text-based Image Generation
    Model', tooltip='Generate image of the
    given food description')
dot.node('1-6', 'Image', shape='box')
dot.node('1-7', 'Data', shape='box')

dot.edge('1-1', '1-2')
dot.edge('1-2', '1-3')
dot.edge('1-3', '1-4')
dot.edge('1-4', '1-5')
dot.edge('1-5', '1-6')
dot.edge('1-6', '1-7')
dot.edge('1-3', '1-7')
```

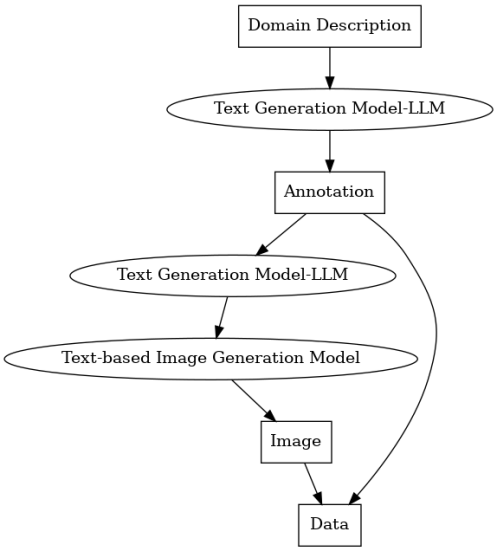


Figure 2: The Graphviz script (left) and its corresponding visualization (right) of the data synthesis pipeline for food composition recognition.

4.1.2 Small Object Detection

We further validate our auto-pipeline agent in the domain of *small object detection*—a challenging scenario where the target objects occupy less than 10% of the image area and are often embedded in complex backgrounds. The data synthesis pipeline generated by our auto-pipeline agent are presented in Figure 3. First, images with complex backgrounds and multiple objects are identified via similarity search. A pre-trained small object detection model is then used to localize small objects within these images. The resulting object regions are further annotated by a vision-language model (VLM) Bai et al. (2025), producing new image-text pairs for the small object detection task. Subsequently, an LLM (Yang et al., 2025) is employed to filter and confirm whether each pair pertains specifically to small object detection, thereby constructing a high-quality, domain-specific dataset. Owing to the high quality of the source datasets, resulting in a collection of about 200,000 high-quality data samples.

4.1.3 Receipts Information Extraction

```
dot = Digraph(comment='Small Object Detection',
      format="png")
dot.node('1-1', 'Domain Description', shape='
     box')
dot.node('1-2', 'Existing Databases', shape='
     box')
dot.node('1-3', 'Image-Text Similarity Search',
      tooltip='Search images from existing
     databases')
dot.node('1-4', 'Image', shape='box')
dot.node('1-5', 'Small Object Detection',
     tooltip='Detect small object from images
     to get annotation')
dot.node('1-6', 'Annotation', shape='box')
dot.node('1-7', 'Data', shape='box')

dot.edge('1-1', '1-3')
dot.edge('1-2', '1-3')
dot.edge('1-3', '1-4')
dot.edge('1-4', '1-5')
dot.edge('1-5', '1-6')
dot.edge('1-6', '1-7')
dot.edge('1-4', '1-7')
```

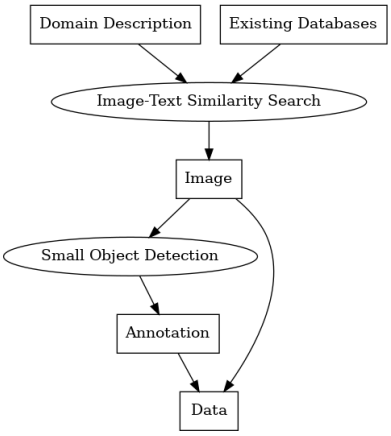


Figure 3: The Graphviz script (left) and its corresponding visualization (right) of the data synthesis pipeline for small object detection.

```
dot = Digraph(comment='Receipts Information
     Extraction', format="png")

dot.node('1-1', 'Domain Description', shape='
     box')
dot.node('1-2', 'Image-Text Similarity Search',
      tooltip='Search images based on the
     receipt scene')
dot.node('1-3', 'Image', shape='box')
dot.node('1-4', "OCR Text Recognition",
     tooltip="Detect and recognize OCR results
     ")
dot.node('1-5', 'General VLM Model', tooltip="
     Use VLM to extract key information and
     then generate new key information")
dot.node('1-6', 'Annotation', shape='box')
dot.node('1-7', 'Image Text Editing Model',
     tooltip="Change key information words on
     the image")
dot.node('1-8', 'Image', shape='box')
dot.node('1-9', 'Data', shape='box')

dot.edge('1-1', '1-2')
dot.edge('1-2', '1-3')
dot.edge('1-3', '1-4')
dot.edge('1-4', '1-5')
dot.edge('1-5', '1-6')
dot.edge('1-6', '1-7')
dot.edge('1-7', '1-8')
dot.edge('1-8', '1-9')
dot.edge('1-6', '1-9')
```

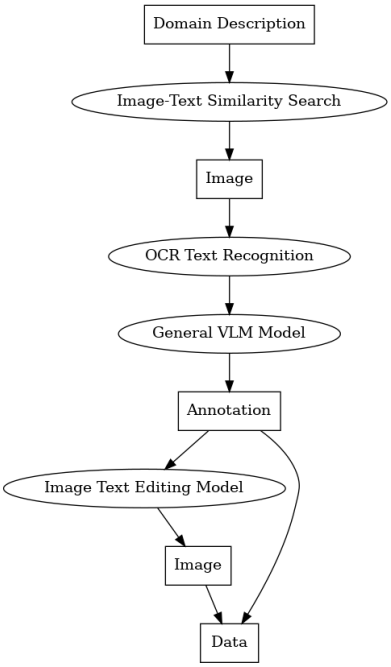


Figure 4: The Graphviz script (left) and its corresponding visualization (right) of the data synthesis pipeline for receipts information extraction.

We further evaluate our approach on the challenging OCR task of *receipts information extraction*, which aims to accurately recognize and extract key textual elements from receipt images. This task is representative of real-world document understanding scenarios, yet the scarcity and high annotation cost of diverse, high-quality receipt datasets significantly hinder the development of robust OCR models.

To address this limitation, we employ our proposed auto-pipeline agent to automatically construct specialized data synthesis pipelines tailored for the OCR domain, as illustrated in Figure 4. Specifically, the pipeline first retrieves relevant receipt images from an existing database via similarity-based matching. Next, a state-of-the-art OCR system, PaddleOCR (Cui et al., 2025), is applied to extract text boxes along with their recognition results. These recognized texts are then passed to

a large language model, Qwen2.5-14B (Bai et al., 2025), which generates multiple semantically equivalent yet content-varied alternatives. Finally, an image-text editing model, QwenImage (Wu et al., 2025), replaces the original key information in the images with the synthesized texts. The synthesized texts, together with the corresponding edited images, constitute the final multimodal dataset obtained through our pipeline.

4.2 SFT WITH SYNTHETIC DATA

To enhance model performance in domain-specific scenarios, we construct synthetic multi-modal datasets for each target vertical, augmenting the original training data with varying amounts of generated samples. For each domain, we systematically control the scale of augmentation to create multiple dataset versions, enabling a fine-grained study of the relationship between synthetic data volume and model adaptation. These datasets are then used to perform Supervised Fine-Tuning (SFT) on two representative vision-language models, InternVL3-2B and InternVL3-8B.

For food composition recognition, we use the train set of *Nutrition5k* dataset (Thames et al., 2021) as the original training data and the test set as our test benchmark, which provides a challenging and diverse set of food images with detailed composition labels. To ensure a fair and consistent evaluation, we employ Qwen3-Embedding to semantically match predicted ingredient lists against ground-truth annotations, thereby accounting for potential variations in ingredient naming and linguistic expression. For small object detection, we split *SORCE-1K* dataset (Liu et al., 2025) equally into training and test sets since the official training set has not been open-sourced yet. For receipts information extraction, we employ the *SROIE* benchmark Huang et al. (2019), which comprises 347 invoice images annotated with key information fields.

As shown in Table 1, for the InternVL-2B model in the food composition domain, augmenting the original 2.7k training samples with synthetic data up to a total of 6k yields the highest performance (F1 = 67.8). Further increases to 9k and 12k result in only minor fluctuations, suggesting early saturation where additional synthetic samples no longer provide consistent gains. In contrast, the InternVL-8B model continues to benefit from larger-scale augmentation, peaking at 9k samples (F1 = 72.1) before showing a slight decline at 12k.

For the small object detection task (Table 2), both InternVL-2B and InternVL-8B reach their highest accuracy when the original 0.5k training set is scaled up to 200k samples, yielding mAP scores of 55.2 and 65.6, respectively. Notably, within the intermediate range of 1k–5k samples, the performance trends differ: the 2B model improves steadily with more data, whereas the 8B model initially drops before recovering as the dataset grows. This discrepancy may stem from differences in the models' learning capacities and their ability to exploit synthetic data at varying scales.

In the receipt information extraction domain (Table 3), peak performance is observed at a relatively small augmentation scale—expanding the original 0.6k samples to 1k yields F1 scores of 82.5 for 2B and 86.5 for 8B. Adding more synthetic data beyond this point fails to produce further gains, likely due to the high stylistic homogeneity of receipts, where excessive augmentation risks introducing redundancy and overfitting.

Train Data	**Data Size**	**InternVL-2B**			**InternVL-8B**		
		Precision	Recall	F1	Precision	Recall	F1
Zero-shot	0	29.7	23.6	26.3	47.3	39.3	42.9
SFT	3k	65.6	58.0	61.6	69.0	65.6	67.3
	6k	**69.5**	66.1	**67.8**	71.9	67.8	69.8
SFT+MG†	9k	68.4	65.1	66.7	**73.6**	**70.6**	**72.1**
	12k	68.2	**66.2**	67.2	73.1	69.6	71.3

Table 1: Performance comparison of InternVL models on the Nutrition5k dataset. †*SFT+MG* denotes Supervised Fine-Tuning with synthetic data generated by MagicGen.

Train Data	Data Size	InternVL-2B		InternVL-8B	
		mAP	m-IoU	mAP	m-IoU
Zero-shot	0	0.6	1.2	6.0	13.2
SFT	0.5k	14.8	19.6	59.6	50.9
	1k	17.4	24.8	56.2	50.1
SFT+MG	5k	22.0	27.2	58.8	50.8
	200k	**55.2**	**48.2**	**65.6**	**55.6**

Table 2: Performance comparison of InternVL models on the SORCE-1K dataset.

Train Data	Data Size	InternVL-2B		InternVL-8B	
		Accuracy	F1	Accuracy	F1
Zero-shot	0	72.9	59.6	86.6	71.7
SFT	0.6k	88.5	80.0	90.9	84.3
	1k	**90.9**	**82.5**	**93.7**	**86.5**
SFT+MG	3k	89.7	82.1	93.4	84.8
	6k	90.6	82.0	93.6	85.1

Table 3: Performance comparison of InternVL models on the SROIE dataset.

4.3 Visual Analysis of Synthetic Data Diversity

To further evaluate the effectiveness of our domain-specific data synthesis pipelines, we conduct a visual analysis focusing on the diversity and richness of the generated datasets. We first present the visualization of CLIP image features of the small object detection datasets using t-SNE in Figure 5(a), where the original dataset is represented in blue and the synthetic dataset in red. The KDE curves delineate the distribution of features, clearly showing that the synthetic data (red points) is more widely dispersed across the feature space, thereby indicating an enhancement in visual diversity.

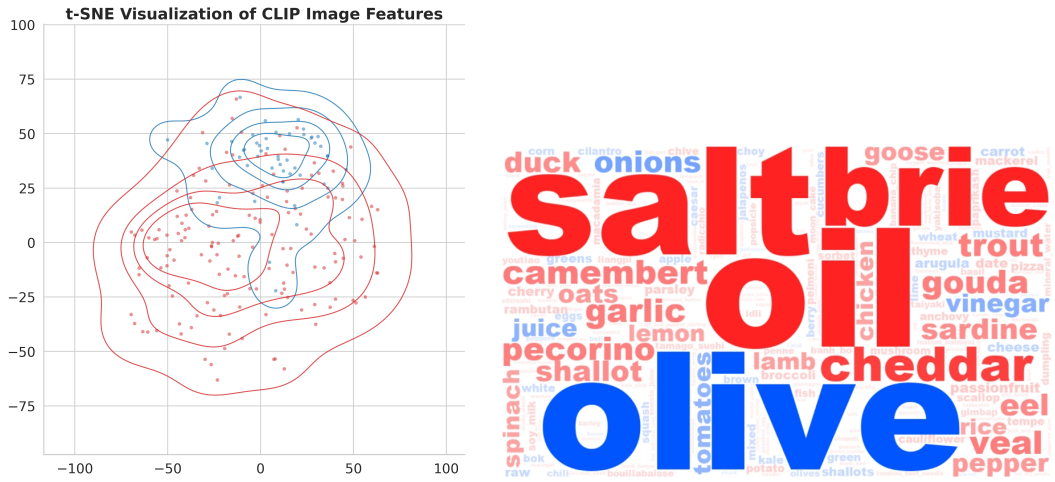
The second visualization examines the textual annotations associated with the food composition datasets through a word cloud representation, as shown in Figure 5(b). The synthetic dataset demonstrates an increased richness in vocabulary, with a more even distribution of terms and broader coverage of semantic concepts compared to the original data. This suggests that the synthesis process not only introduces additional descriptive variety but also balances the representation of different concepts within the dataset.

Together, these visualizations provide complementary perspectives on the synthetic data's characteristics, demonstrating its potential to not only expand the coverage of the original dataset in the visual feature space but also to enrich its semantic content.

5 Conclusion

In this work, we presented MagicGen, a universal, automated data synthesis pipeline agent capable of generating high-quality, domain-specific multimodal datasets with minimal human intervention. By leveraging a curated and standardized multimodal toolset, hybrid supervision that combines expert-authored references with iteratively validated LLM-generated candidates, and a hierarchical evaluation framework for automated quality control, MagicGen addresses the key challenges of scalability, data fidelity, and cross-domain adaptability in vision–language model (VLM) training.

Our experimental results across diverse vertical domains—food composition recognition, small object detection, and OCR-based information extraction—demonstrate that MagicGen not only enhances dataset diversity and annotation accuracy but also significantly reduces manual design overhead. The modular tool selection and chaining mechanism enables flexible adaptation to new do-



(a) (b)

Figure 5: t-SNE visualization of CLIP image features with KDE contour lines (left) and word cloud for annotations (right). In both figures, red indicates synthetic data, while blue represents the original data.

mains, while the self-evolving reinforcement learning loop continuously refines the agent's reasoning and planning capabilities, ensuring long-term robustness and scalability.

Beyond the immediate performance gains in downstream VLM fine-tuning tasks, MagicGen establishes a generalizable paradigm for automated, domain-aware data synthesis. The proposed pipeline design methodology and evaluation strategies can be readily extended to other specialized application scenarios, paving the way for fully autonomous, high-quality dataset construction systems.

Future work will explore dynamic tool discovery and integration to further expand the agent's capabilities, as well as multi-agent cooperative synthesis for large-scale, distributed data generation. We also plan to investigate tighter integration between synthetic data generation and adaptive model training, enabling closed-loop, data–model co-evolution for even greater efficiency and generalization.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pp. arXiv–2506, 2025.

Gagan Bhatia, El Moatez Billah Nagoudi, Hasan Cavusoglu, and Muhammad Abdul-Mageed. Fintral: A family of gpt-4 level multimodal financial large language models. *arXiv preprint arXiv:2402.10986*, 2024.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pp. 446–461. Springer, 2014.

Jiankang Chen, Tianke Zhang, Changyi Liu, Haojie Ding, Yaya Shi, Feng Cheng, Huihui Xiao, Bin Wen, Fan Yang, and Tingting Gao. Taskgalaxy: Scaling multi-modal instruction fine-tuning with tens of thousands vision task types. 2025.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source

multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Daixuan Cheng, Shaohan Huang, Ziyu Zhu, Xintong Zhang, Wayne Xin Zhao, Zhongzhi Luan, Bo Dai, and Zhenliang Zhang. On domain-specific post-training for multimodal large language models. *arXiv preprint arXiv:2411.19930*, 2024.

Qihao Cheng, Da Yan, Tianhao Wu, Zhongyi Huang, and Qin Zhang. Computing approximate graph edit distance via optimal transport. *Proceedings of the ACM on Management of Data*, 3(1):1–26, 2025.

Juhwan Choi, Junehyoung Kwon, JungMin Yun, Seunguk Yu, and YoungBin Kim. Voldoger: Llm-assisted datasets for domain generalization in vision-language tasks. *arXiv preprint arXiv:2407.19795*, 2024.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.

Ana Cláudia Akemi Matsuki de Faria, Felype de Castro Bastos, José Victor Nogueira Alves da Silva, Vitor Lopes Fabris, Valeska de Sousa Uchoa, Décio Gonçalves de Aguiar Neto, and Claudio Filipi Goncalves dos Santos. Visual question answering: A survey on techniques and common trends in recent literature. *arXiv preprint arXiv:2305.11033*, 2023.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Shafiq Joty, Boyang Li, and Lidong Bing. Is gpt-3 a good data annotator? *arXiv preprint arXiv:2212.10450*, 2022.

Yu Du, Fangyun Wei, and Hongyang Zhang. Anytool: Self-reflective, hierarchical agents for large-scale api calls, 2024. *URL https://arxiv. org/abs/2402.04253*.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 11198–11201, 2024.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. Self-guided noise-free data generation for efficient zero-shot learning. *arXiv preprint arXiv:2205.12679*, 2022.

Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang, Xin Xin, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 18030–18038, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL `https://arxiv.org/abs/1406.2661`.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1516–1520. IEEE, 2019.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*, 2022.

Su Young Kim, Hyeonjin Park, Kyuyong Shin, and Kyung-Min Kim. Ask me what you need: Product retrieval using knowledge from gpt-3. *arXiv preprint arXiv:2207.02516*, 2022.

Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27831–27840, 2024.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36: 28541–28564, 2023a.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. *arXiv preprint arXiv:2304.08244*, 2023b.

Xiang Li, Congcong Wen, Yuan Hu, Zhenghang Yuan, and Xiao Xiang Zhu. Vision-language models in remote sensing: Current progress and future trends. *IEEE Geoscience and Remote Sensing Magazine*, 2024.

Xinyao Li, Jingjing Li, Fengling Li, Lei Zhu, Yang Yang, and Heng Tao Shen. Generalizing vision-language models to novel domains: A comprehensive survey. *arXiv preprint arXiv:2506.18504*, 2025.

Chunxu Liu, Chi Xie, Xiaxu Chen, Wei Li, Feng Zhu, Rui Zhao, and Limin Wang. Sorce: Small object retrieval in complex environments, 2025. URL https://arxiv.org/abs/2505.24441.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Weiqing Min, Linhu Liu, Zhiling Wang, Zhengdong Luo, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 393–401, 2020.

Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. Large scale visual food recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):9932–9949, 2023.

Vishwesh Nath, Wenqi Li, Dong Yang, Andriy Myronenko, Mingxin Zheng, Yao Lu, Zhijian Liu, Hongxu Yin, Yee Man Law, Yucheng Tang, et al. Vila-m3: Enhancing vision-language models with medical expert knowledge. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14788–14798, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

OpenAI. Gpt-4 technical report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, volume abs/2303.08774, 2023.

OpenAI. Hello GPT-4o, 2024. URL https://openai.com/index/hello-gpt-4o/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13009–13018, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Hangyu Mao, Ziyue Li, Xingyu Zeng, Rui Zhao, et al. Tptu: Task planning and tool usage of large language model-based ai agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

Gaurav Sahu, Pau Rodriguez, Issam H Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. Data augmentation for intent classification with off-the-shelf large language models. *arXiv preprint arXiv:2204.01959*, 2022.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. Tool learning in the wild: Empowering language models as automatic tool agents. In *Proceedings of the ACM on Web Conference 2025*, pp. 2222–2237, 2025.

Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From show to tell: A survey on deep learning-based image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):539–559, 2022.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360*, 2023.

Quin Thames, Arjun Karpur, Wade Norris, Fangting Xia, Liviu Panait, Tobias Weyand, and Jack Sim. Nutrition5k: Towards automatic nutritional understanding of generic food. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8903–8911, 2021.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024a.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024b.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng Ming Yin, Shuai Bai, Xiao Xu, and Yilei Chen. Qwen-image technical report. 2025.

Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. *Advances in Neural Information Processing Systems*, 36:54683–54695, 2023.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lin Yang, Shawn Xu, Andrew Sellergren, Timo Kohlberger, Yuchen Zhou, Ira Ktena, Atilla Kiraly, Faruk Ahmed, Farhad Hormozdiari, Tiam Jaroensri, et al. Advancing multimodal medical capabilities of gemini. *arXiv preprint arXiv:2405.03162*, 2024.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*, 2022.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

Wei Zhang, Miaoxin Cai, Tong Zhang, Yin Zhuang, and Xuerui Mao. Earthgpt: A universal multimodal large language model for multi-sensor image comprehension in remote sensing domain. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

Chenyang Zhao, Xueying Jia, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Self-guide: Better task-specific instruction following via self-synthetic finetuning. *arXiv preprint arXiv:2407.12874*, 2024.

Yuan Zhong, Ruinan Jin, Xiaoxiao Li, and Qi Dou. Can common vlms rival medical vlms? evaluation and strategic insights. *arXiv preprint arXiv:2506.17337*, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.