InstaDA: Augmenting Instance Segmentation Data with Dual-Agent System

Supplementary Material

7. Implementation Details

7.1. Training details

All our experiments are conducted on a server equipped with eight NVIDIA L20 GPUs, using CUDA 11.6 and PyTorch 1.13.1. We trained models using the AdamW optimizer. To align our methodology directly with prior work, we replicated the setup from DiverGen [4] by setting the training seed to 42 and adopting their exact configurations for all data augmentations, including random horizontal flips, random crops, and the specific parameters of Copy-Paste [7]. Regarding the training time, the baseline model completes its training in approximately 26 hours. With our proposed data augmentation, the training duration is dependent on the data I/O performance of the training environment. On a local machine with high-speed data access, training typically completes within 46 to 51 hours. In our cloud computing environment, where data loading proved to be a bottleneck, the duration extended to 80 to 113 hours. This computational overhead remains comparable to that of DiverGen. The additional training time compared to the baseline is primarily due to the on-the-fly processing overhead of the Copy-Paste augmentation applied to each training batch.

7.2. Prompt template

T-Agent For the language model component of the T-Agent, we select the open-source DeepSeek [8, 19] models to substantially lower inference costs. As illustrated in Fig. 7, the entire process is fully automated, requiring only the list of category names from the LVIS [9] dataset to initiate the image generation process. For Prompt Rethink, detailed in Fig. 8, we design prompts to address issues at both the semantic and visual levels.

I-Agent The I-Agent utilizes Qwen2-VL-7B [31] to generate a text prompt that captures the semantic content of the source image. To achieve this, we employ a simple and direct template: "Based on this image, combine {*number* + *category*} into one sentence." {*number* + *category*} is dynamically filled by extracting the instance counts and their corresponding category names directly from the LVIS annotations of the given image.

8. Details of GDDE

Generative Data Diversity Enhancement (GDDE), proposed by DiverGen [4], is a strategy aimed at improving data diversity from three key perspectives: category, prompt, and generative model. (1) Category Diversity: This aspect incorporates additional categories from ImageNet based on semantic similarity. Although this approach helps reduce some misidentifications, it also introduces greater complexity to the generation process. (2) Prompt Diversity: For this, large language models are utilized, but all generations are constrained to a "blank background" in alignment with the SAM-bg [4] strategy. (3) Model Diversity: Finally, GDDE employs two diffusion models, Stable Diffusion [27] and DeepFloyd-IF, to diversify the visual styles of the generated instances. This multi-faceted approach enhances overall data diversity while addressing the challenges associated with each aspect.

Table 9. Comparison of the average generation time per image. Our method, InstaDA (comprising T-Agent and I-Agent), significantly reduces the image generation time compared to DiverGen, enhancing its practical applicability.

Method	Runtime (s)
DiverGen	60
T-Agent (FLUX)	**2.4**
T-Agent (SD3.5)	4
I-Agent (FLUX)	8.7

Table 10. Ablation study on I-Agent filtration strategies. Proportional filtration (with a 30% ratio) significantly outperforms the strategy using a text similarity threshold of 0.21.

Strategy	AP^{box}	AP^{mask}	AP_r^{box}	AP_r^{mask}
Text similarity threshold	50.87	45.01	46.61	41.61
Proportional filtration	**51.51**	**45.51**	**50.10**	**44.31**

9. Efficiency Analysis

9.1. Model Configuration

The Text-Agent (T-Agent) employs two diffusion models: (1) FLUX.1-dev [15], configured to generate images at a 512 $\times$ 512 resolution using the Euler sampler with a guidance scale of 9.5; and (2) Stable Diffusion 3.5-large [28], which generates images at a 1024 $\times$ 1024 resolution, also using the Euler sampler with its default settings. The Image-Agent (I-Agent) utilizes FLUX.1-dev with a ControlNet [39] strength of 0.7 for structural guidance. Crucially, the integration of LoRA [11] allows us to reduce the sampling steps for all generative processes to just 8, significantly improving efficiency. All generative workflows are implemented using the ComfyUI framework.

For instance processing, we utilize BiRefNet [41]

Table 11. Ablation study on the thresholds for the CLIP dual-similarity metric. The results demonstrate that a text similarity threshold of 0.21 and an image similarity threshold of 0.6 achieve the best performance across all evaluation metrics.

Gen Data	Text Similarity Threshold	Image Similarity Threshold	AP^{box}	AP^{mask}	AP_r^{box}	AP_r^{mask}
480k	0.25	0.6	50.41	44.59	45.64	40.48
480k	0.23	0.6	50.52	44.80	47.41	42.41
480k	0.22	0.6	50.73	44.73	48.33	42.51
480k	0.21	0.6	**50.79**	**45.03**	**48.47**	**43.26**
480k	0.20	0.6	50.49	44.86	47.17	42.24
480k	0	0.6	50.33	44.53	46.44	41.36
480k	0.21	0	50.64	44.92	47.36	42.60

Table 12. Ablation on the proportional filtration ratio k. The results show that a 20% data expansion ($k = 20$) achieves the best performance across all evaluation metrics.

k (%)	AP^{box}	AP^{mask}	AP_r^{box}	AP_r^{mask}
30	51.18	45.37	47.73	43.07
20	**51.33**	**45.43**	**48.82**	**43.59**
10	50.07	45.21	48.03	42.56

Table 13. Efficiency comparison of the total generation runtime. The reported time for our T-Agent (FLUX) includes the computational cost of all Prompt Rethink iterations. Here, h stands for hours and m for minutes.

Method	Runtime
DiverGen	50h10m
T-Agent (FLUX)	4h20m

(BiRefNet-general-resolution_512x512-fp16-epoch_216), the Segment Anything Model (SAM) [14] ViT-H, and the CLIP[24] ViT-L/14 model for filtration. The UMAP [21] visualizations are generated with n_neighbors=15, min_dist=0.1, n_components=2. To ensure reproducibility, all experiments use a fixed random seed of 42.

9.2. Generation Efficiency and Comparison

Our generation process stands in stark contrast to the prior work DiverGen [4], which relies on a computationally intensive pipeline involving two separate stages. The process of DiverGen involves a 100-step base generation followed by a 50-step refinement stage, totaling 150 sampling steps. While this approach may enhance image quality, it incurs a prohibitive temporal cost, especially for large-scale data generation. In contrast, our methodology, accelerated by LoRA [26], accepts a marginal reduction in image quality in exchange for a dramatic acceleration. By reducing the process to a single stage with just 8 sampling steps, the optimal number proposed in [26], we eliminate this bottleneck, thereby making large-scale generative augmentation feasible. As evidenced in Table 9, this approach results in a substantial speedup, accelerating the data generation process (excluding rethink) by a factor of **6.8** to **25** compared to DiverGen.

9.3. Efficiency Analysis of Prompt Rethink

Each iteration of our Prompt Rethink mechanism runs the instance generation, foreground segmentation, and instance filtration pipeline again for failed prompts, incurring computational costs. We evaluate the mechanism on a set of 480k images, where 192,119 instances initially pass the filtration. The first rethink boosts this count by 62% to 311,174, while the second brings the cumulative gain to 83% (351,114 instances). The clear diminishing returns validate our two-step mechanism, as further iterations would be computationally inefficient. Crucially, by sequentially targeting semantic and then visual failures, the tiered process ensures that computational resources are utilized effectively to correct specific types of errors.

The overall runtime is overwhelmingly dominated by the instance generation, as the subsequent segmentation and filtration are computationally inexpensive. Consequently, we exclude the minor overhead from these post-processing stages and compare the total generation time of InstaDA, including the rethink mechanism, directly with DiverGen to evaluate practical performance. Since DiverGen does not provide details about its prompt generation time, we exclude this component from our comparison. As shown in Tab. 13, when generating a total of 24k images (20 images per category), InstaDA is approximately **11** times faster than DiverGen in terms of total generation time, utilizing a DeepSeek [8, 19] application programming interface (API) with a concurrency of 1,203.

10. Ablation Studies

Effect of the I-Agent proportional filtration We compare our proportional filtration against a method based on a text similarity threshold. As shown in Table 10, the threshold-based approach degrades performance. In contrast, our proportional filtration method yields a notable performance

Table 14. Ablation on the effectiveness of Copy-Paste. The hybrid method outperforms the full dynamic sampling method, highlighting a performance bottleneck in the standard Copy-Paste.

Method	AP^{box}	AP^{mask}	AP_r^{box}	AP_r^{mask}
Full	51.5	45.6	50.1	**44.6**
Hybrid	**51.7**	**45.7**	**50.8**	**44.6**

improvement. This degradation occurs because threshold-based, per-instance filtration is incompatible with per-image Copy-Paste [7], which can lead to pasting incomplete instance sets from a source image. Conversely, our proportional filtration operates at the image level, aligning the filtration unit with the Copy-Paste unit. This alignment preserves the completeness of instance groups, which directly translates to the performance improvement.

Ablation on thresholds for CLIP dual-similarity Table 11 presents our ablation study to determine the optimal parameters for our CLIP dual-similarity metric: a text similarity threshold of 0.21 and an image similarity threshold of 0.6. This combination yields the best performance because it strikes an optimal balance between preserving data diversity and maintaining high domain fidelity.

Ablation on proportional filtration ratio We conduct an ablation study to find the optimal proportional filtration ratio, denoted by the parameter k. This parameter specifies the data expansion percentage, where a value of $k = 20$ corresponds to a 20% expansion. As illustrated in Table 12, the performance peaks at $k = 20$. This suggests that a 20% data expansion achieves the most effective overall data distribution by introducing adequate diversity. Accordingly, we set $k = 20$ for our method in all experiments.

Ablation on the effectiveness of Copy-Paste As shown in Tab. 14, we compare two methods under identical settings: (1) a full method, which dynamically samples from the entire generated data pool, and (2) a hybrid approach. The second method involves two stages. First, one synthetic instance per category is permanently added to the training set. Second, the subsequent Copy-Paste augmentation samples only from the remaining pool (excluding the added instances). This result highlights a limitation in the standard Copy-Paste approach, indicating a performance bottleneck that the hybrid method helps to overcome.

Ablation of I-Agent components We ablate two key components of the I-Agent: the Qwen2-VL-7B [31] text guidance and the adaptive denoising strength. The fused edge map is kept fixed, as it is essential for image generation in our pipeline. As shown in Tab. 15, removing either the text prompt or the adaptive strength degrades performance. This indicates that stronger semantic alignment and a tailored denoising strength are both crucial for fully exploiting the I-Agent and improving overall model performance.

Table 15. Ablation study on the I-Agent components. The results demonstrate that both semantic alignment (Qwen) and a tailored denoising strength (adaptive strength) are critical for improving performance.

Qwen	adaptive strength	AP^{box}	AP^{mask}	AP_r^{box}	AP_r^{mask}
✗	✓	**49.5**	43.7	42.0	37.5
✓	✗	49.3	43.7	41.8	37.3
✓	✓	**49.5**	**43.9**	**42.1**	**37.6**

11. Visualization

Prompt Rethink Fig. 9 shows examples of failed generations for the air conditioner that trigger our Prompt Rethink mechanism. The primary causes of failure are either semantic or visual discrepancies.

Visualization of the limitations of CLIP As shown in Fig. 10, the limitations of CLIP cause some visually promising samples to be filtered out, despite their potential to enhance model performance. The observation suggests that employing a more powerful multimodal model is a promising avenue for improving the effectiveness of our generative data augmentation.

More examples We provide additional visual examples for T-Agent (Fig. 11) and I-Agent (Fig. 12) to further demonstrate their distinct functionalities.

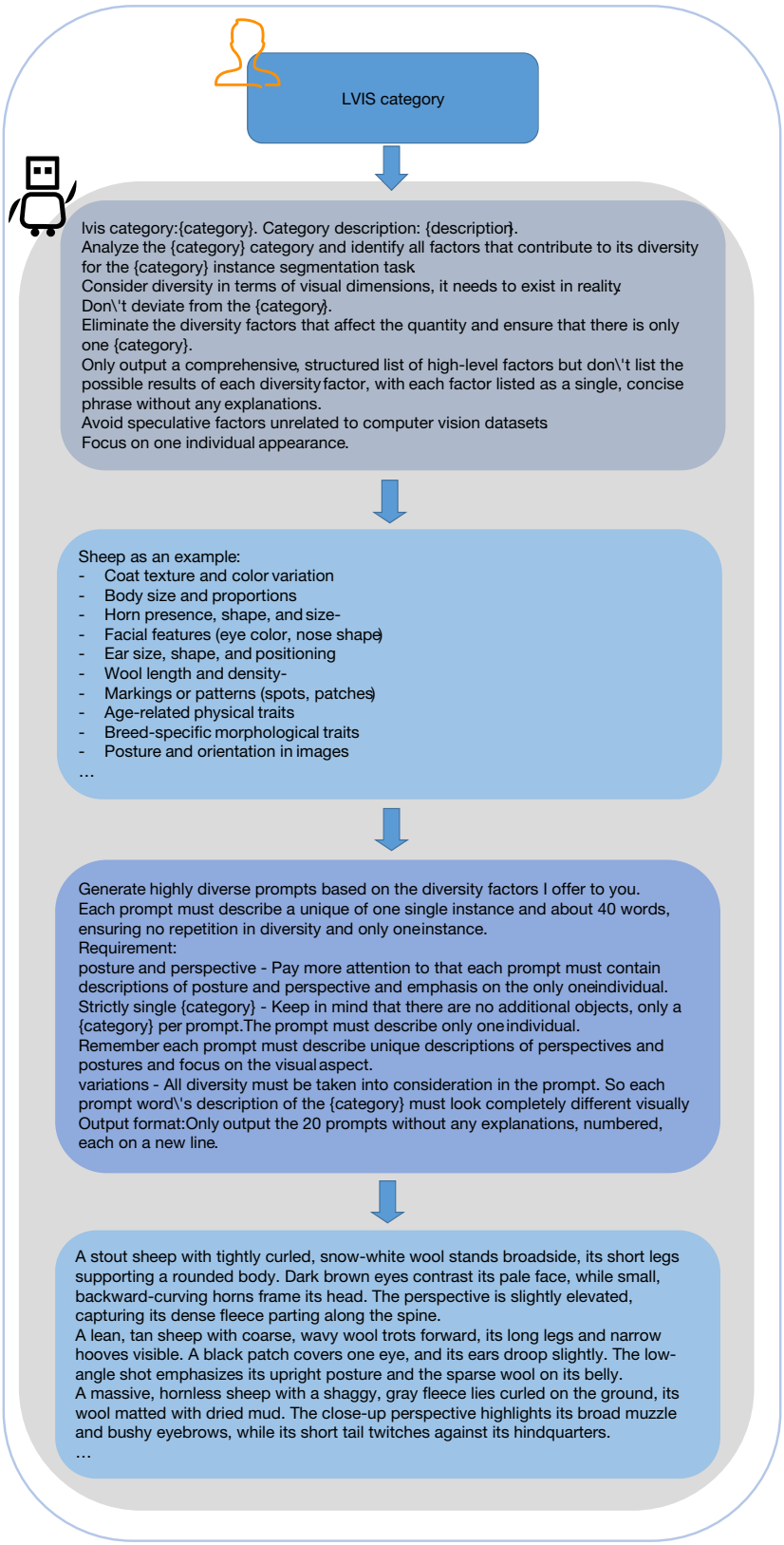


Figure 7. Example of the automated prompt generation pipeline. The workflow takes an LVIS category name, represented by {category}, and its corresponding metadata description: {description}, to systematically generate a diverse set of prompts.

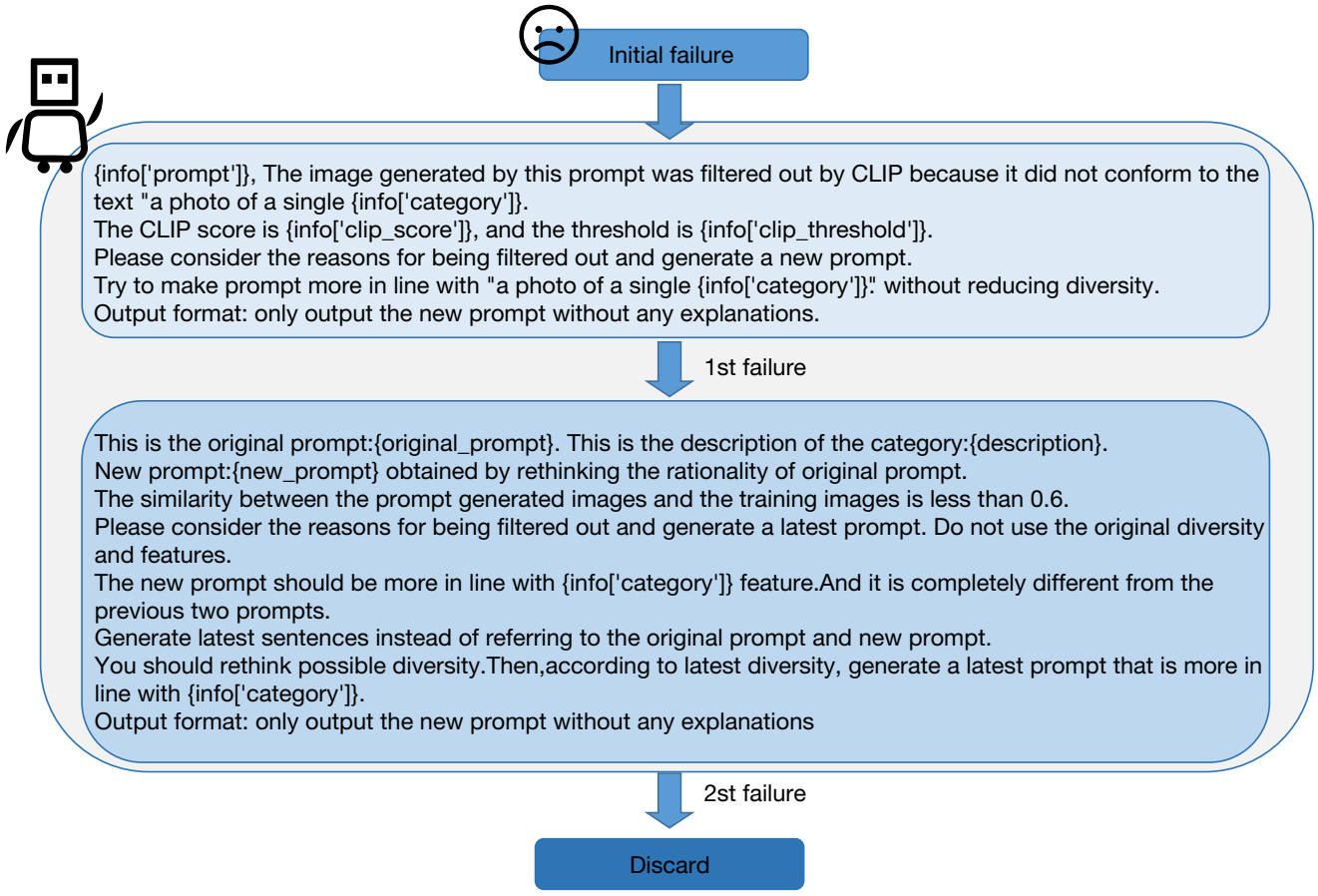
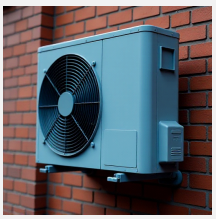


Figure 8. Template for the Prompt Rethink feedback loop. The process analyzes the historical generation data ({info}) from the T-Agent.

A split outdoor unit with a cylindrical shape, matte blue finish, and spiral vent pattern. Seen from a diagonal angle, it's mounted on a brick wall, its unique shape and color making it a focal point.

A photo of a single air_conditioner with a cylindrical shape, matte blue finish, and spiral vent pattern, mounted on a brick wall, seen from a diagonal angle, showcasing its unique design as a focal point.



A compact split outdoor unit with a camouflage-patterned coating. Seen from a side view, its low-profile design and recessed vents make it nearly blend into its surroundings.

A photo of a single air_conditioner with a compact split outdoor unit featuring a camouflage-patterned coating, low-profile design, and recessed vents, blending into its surroundings from a side view.

A sleek, modern split-type air conditioner with a textured matte finish, showcasing its streamlined design and subtle vent placement from a side angle, emphasizing its functional yet discreet appearance in an urban setting.

Figure 9. Visualization of the complex prompts and the corresponding low-quality images, which trigger the rethink process. The top illustrates the first rethink stage, where the prompt is simplified to ensure semantic alignment. The bottom depicts the second stage, which is activated if the first attempt also fails, revising the visual elements of the prompt to improve visual fidelity.

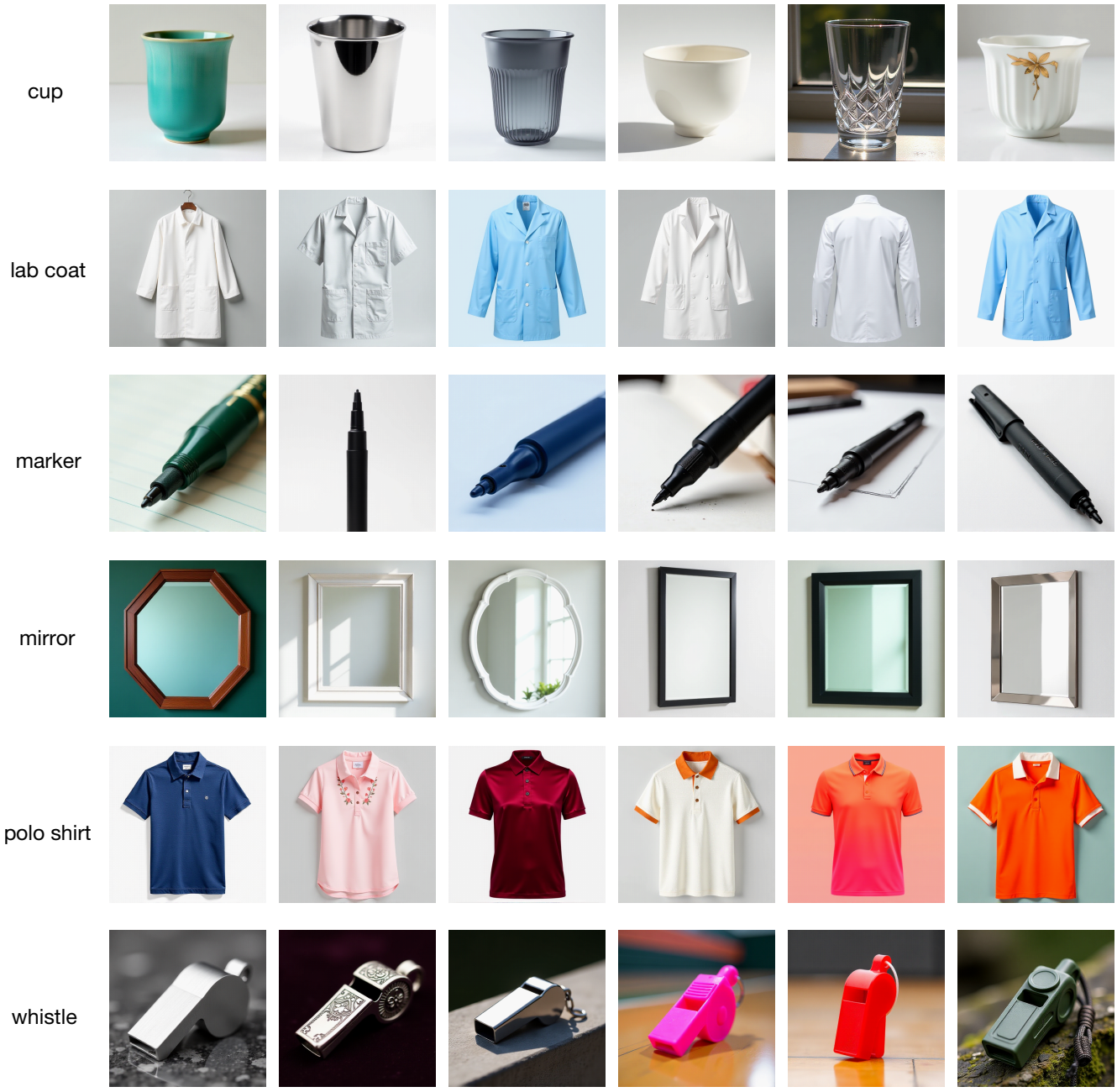


Figure 10. Visualization of visually appealing images filtered out by CLIP. This indicates that leveraging superior multimodal models for filtration is a promising avenue for further enhancing dataset quality.

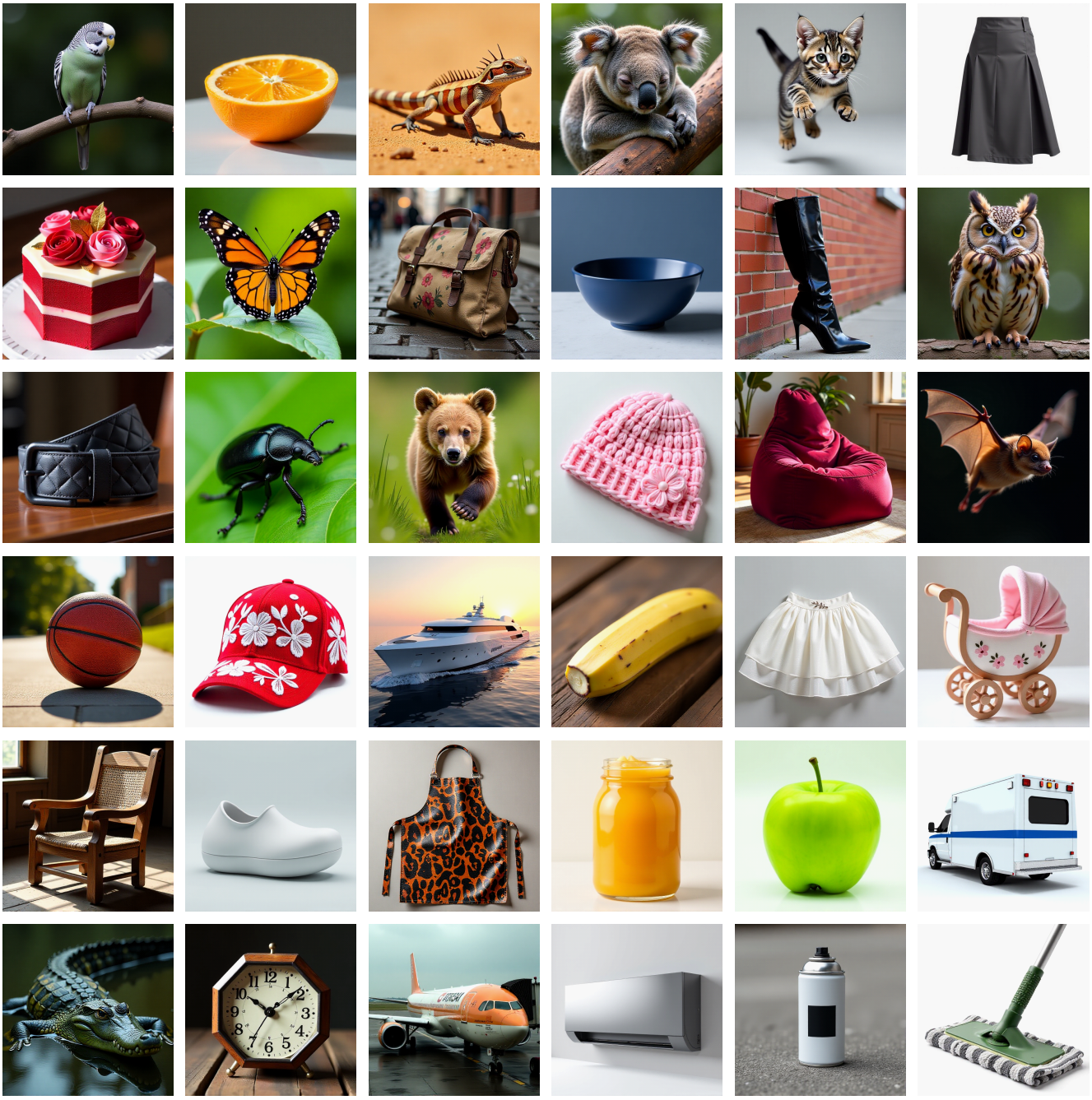


Figure 11. Examples generated by the T-Agent. The remarkable visual diversity is crucial for mitigating overfitting.

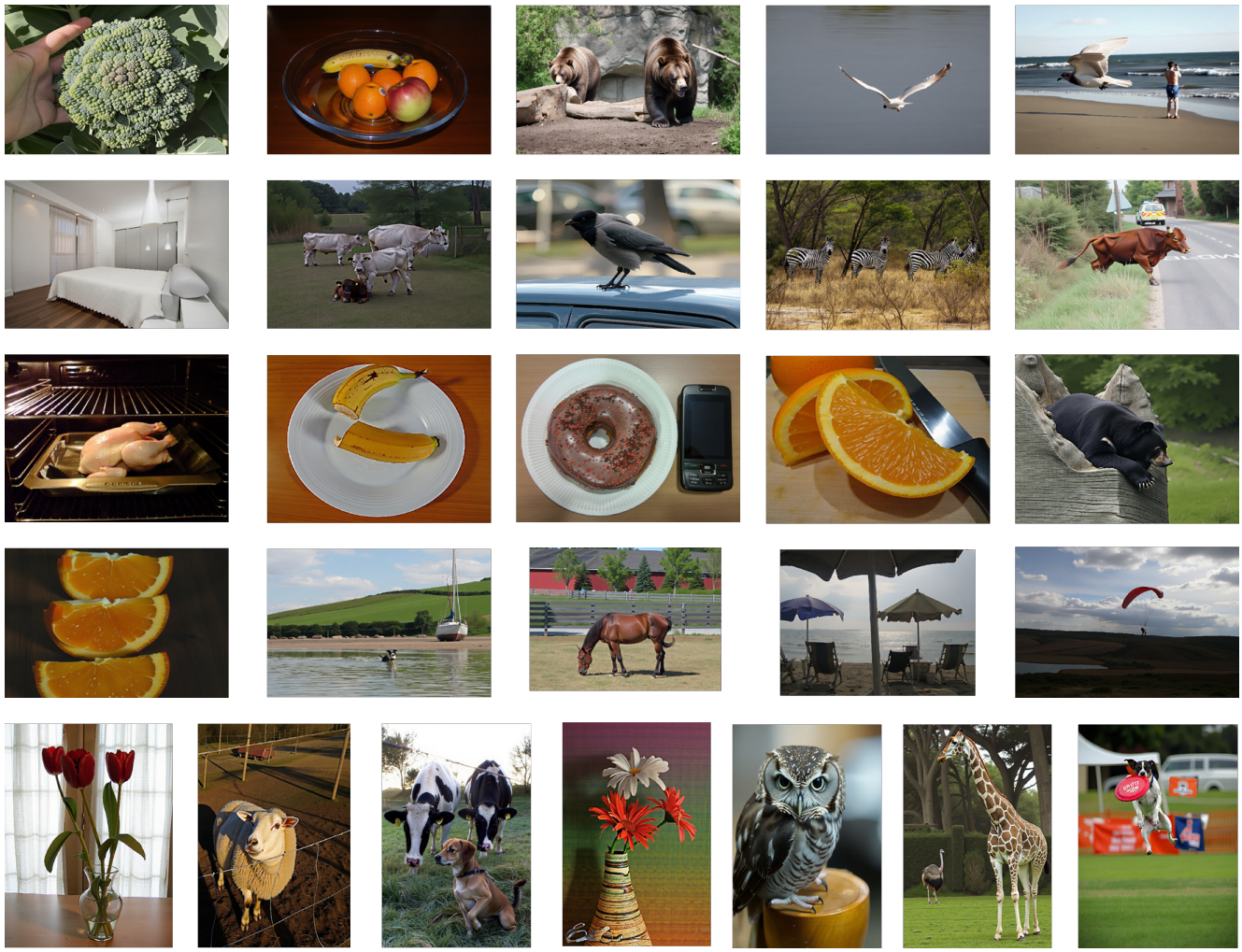


Figure 12. Examples generated by the I-Agent. These samples play a vital role in enriching the overall data distribution.