Diffusion Mental Averages

Supplementary Material

A. Additional Implementation Details

A.1. Prompt Templates

Our method uses category-specific prompt templates as input to the Stable Diffusion model:

- Animal: *A photo of a* {*concept name*}
- Person: *Photo portrait of a* {*concept name*}
- Object: *A photo of a* {*concept name*}
- Abstract: *A conceptual photo representing* {*concept name*}

For Dreamshaper PixelArt, we append "pixel art style, detailed" to the end of the prompt.

A.2. Negative Prompt

For each variant, we use the negative prompt recommended by the respective author. All weighted negative prompts were encoded using **Compel**[1].

Realistic Vision v5.1's negative prompt: *(deformed iris, deformed pupils, semi-realistic, cgi, 3d, render, sketch, cartoon, drawing, anime:1.4), text, close up, cropped, out of frame, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, dehydrated, bad anatomy, bad proportions, extra limbs, cloned face, disfigured, gross proportions, malformed limbs, missing arms, missing legs, extra arms, extra legs, fused fingers, too many fingers, long neck"*

Dreamshaper PixelArt's negative prompt: *"worst quality"*

Flat 2D Animerge's negative prompt: *(worst quality:0.8), verybadimagenegative_v1.3, (surreal:0.8), (modernism:0.8), (art deco:0.8), (art nouveau:0.8)"*

A.3. BLIP-VQA for Grounded Clustering

We follow the grounded clustering method introduced in Stable Bias [56], where it is used as one of several tools for evaluating bias in diffusion models. This method clusters image embeddings produced by BLIP-VQA to group images that share similar attributes. In their work, these embeddings are conditioned on a question prompt, such as "What word best describes this person's ethnicity?", which ensures that the resulting embedding focuses on the specified attribute (in this example, ethnicity). We adapt this work for our task using the following prompt: "What word best describes this {concept name}'s {attribute}?".

[1] https://github.com/damian0815/compel

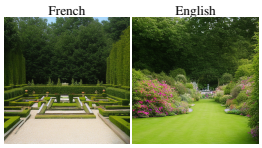
A.4. DiT Implementation Details

We use the DiT-XL/2-256 model [69], a large-scale Diffusion Transformer trained for class-conditional image generation. Unless otherwise noted, all DiT results are generated using a classifier-free guidance scale of 10.0, a learning rate of 5×10^{-4}, 20 inference steps, and a DMA stopping timestep of $t_{\text{stop}} = 20$. We use the output of the last transformer block in place of the Stable Diffusion's h-space to perform averaging in Algorithm 1. Experiment C.7 evaluates the use of different transformer blocks in DMA.

B. Additional Discussion

B.1. Practical Applications of Mental Averages

Mental averages offer a concrete visual summary of a learned concept, enabling analysis beyond individual samples and elucidating the model's internal representation.



Questions like "What distinguishes French from English gardens?" are hard to answer from samples alone, as people often lack clear visual expectations and do not know what to look for. Inspecting thousands of samples for statistical significance is inefficient and still not as conclusive, as it relies on the viewer's ability to detect both between- and within-class patterns. Averages can readily reveal French as more geometric and English as more organic—direct visual evidence that individual samples rarely make obvious (please see our website for more results). Running attribute classifiers for "geometricness" could offer summaries, but it is infeasible when relevant attributes are not known in advance or easily described by words.

This summary can assist in auditing model bias and diagnosing how different model variants alter a specific concept as shown in Figure 8. Beyond visualization, mental averages could serve as a tool to compress data into compact statistics for faster interpretation or privacy-preserving downstream training (e.g., *dataset distillation* [13, 29, 89]). Furthermore, they may act as regularizers and facilitate model debiasing. We leave these directions for future work.

B.2. Computational Costs

The current optimization process takes around 10 hours per concept on a single NVIDIA RTX 4080, with $K = 1000$, $t_{\text{stop}} = 10$, and $N = 300$. This remains our primary limitation. However, the three hyperparameters—the DMA number of optimization steps N, the stopping timestep t_{stop}, and

the number of sampled latents K—enable a trade off between consistency and efficiency. The computation cost of DMA scales linearly with these hyperparameters. As shown in Figure 10, reducing N from 300 to 5 speeds up DMA by approximately 60 times (taking only 10 minutes) and produces averages that already capture the same layout and composition. Increasing N to 10 improves color fidelity, and $N = 300$ captures fine details like leaf veins. Similarly, t_{stop} acts as an early stopping point for noise optimization; without it, all diffusion steps must be optimized. As shown in Figure 10, setting t_{stop} to 10 doubles the optimization speed with only minor degradation, while lowering K also helps accelerate DMA but may result in inconsistency across disjoint sets (Sec. C.6).

Ultimately, the choices for these hyperparameters depend on the specific task and the extent to which consistency can be traded for efficiency. For instance, a coarse-grained alignment of textures might be sufficient for general visualization (e.g., dog visualization can tolerate fur misalignment), whereas tasks like plant species recognition may require fine alignment of intricate features like leaf veins.

B.3. Limitations of Mode Discovery and Averages

In mode discovery (Section 4.1), our method focuses on computing the average of each cluster once they are established. This clustering is treated as a modular preprocessing step, currently using standard techniques like GMM on CLIP features, that can be independently replaced or improved. We observe that because different algorithms partition the space differently, the resulting mode averages are inherently dependent on the specific clustering method.

While applying LoRA on each cluster (Section 4.1) helps improve the model's conditioning of specific subconcepts, it modifies the original model being probed and may be less ideal as a diagnostic tool for the original model. Nonetheless, since our LoRA is trained exclusively on generated samples from the original model, it avoids introducing biases from external data.

B.4. Discussion on Representativeness Score

Formally, the average of a set of images is defined as the point that minimizes the mean squared distance to all images in the set. Based on this principle, we compute distances within semantic latent spaces such as CLIP, DreamSim, and LPIPS to quantify the optimality of the average, or the Representativeness Score. However, a good score in these spaces could only reflect the quality of the average within those specific spaces. We suggest that identifying the most suitable latent space for evaluating the perceptual and semantic fidelity of average images remains an interesting direction for future work.

C. Ablation Studies

C.1. Progressive Mean Alignment

In DMA, the mean h-space activation at each timestep is computed progressively using the current set of latents optimized from the previous timestep. As a baseline, we consider an alternative approach that precomputes the means for all timesteps using K *independent* full denoising processes: we completely denoise each of the K latents, extract their h-space activations at every timestep, and average them to obtain the per-timestep mean. DMA is then rerun from the start using the original K latents from $\mathcal{N}(0, \mathbf{I})$, but with the precomputed means used as alignment targets during noise optimization.

As shown in Figure 11, this baseline produces artifacts and increasingly corrupted results as optimization progresses toward later timesteps (i.e., closer to the final output). In contrast, our method yields high-quality images, highlighting its effectiveness.

C.2. Multi-Timestep Optimization

We consider a baseline that optimizes noise latents only at a single timestep, while all other timesteps follow standard inference. As shown in Fig. 12, single-timestep optimization produces inconsistent results regardless of the chosen timestep, and optimizing at later timesteps introduces artifacts and degrades output quality. In contrast, DMA yields more consistent and high-quality results.

C.3. Noise Optimization

We present a naive baseline, dubbed the *replacement baseline*, that directly substitutes the averaged h-space at each timestep during denoising instead of performing noise optimization. Here, the averaged h-space is computed in the same way as in DMA, and this substitution is applied across diffusion timesteps similarly until a cutoff $t_{\text{stop}} < T$. As shown in Figure 13, the baseline produces inconsistent results for all t_{stop} values. This inconsistency arises from skip connections, which can introduce stochastic information from individual noise latents, highlighting the need for noise optimization to constrain activations in other layers.

C.4. Effects of t_{stop}

We analyze the effects of varying the cutoff timestep t_{stop} in Figure 14. The results show that a low $t_{\text{stop}} = 5$ leads to inconsistent average images, such as architectural incoherence in the *castle*, ground-plane variations in the *dog*, and costume inconsistencies in the *firefighter*. Increasing t_{stop} beyond 10 provides only minor improvements in visual consistency for most concepts. We therefore set $t_{\text{stop}} = 10$ as a balance between computational efficiency and cross-sample consistency.

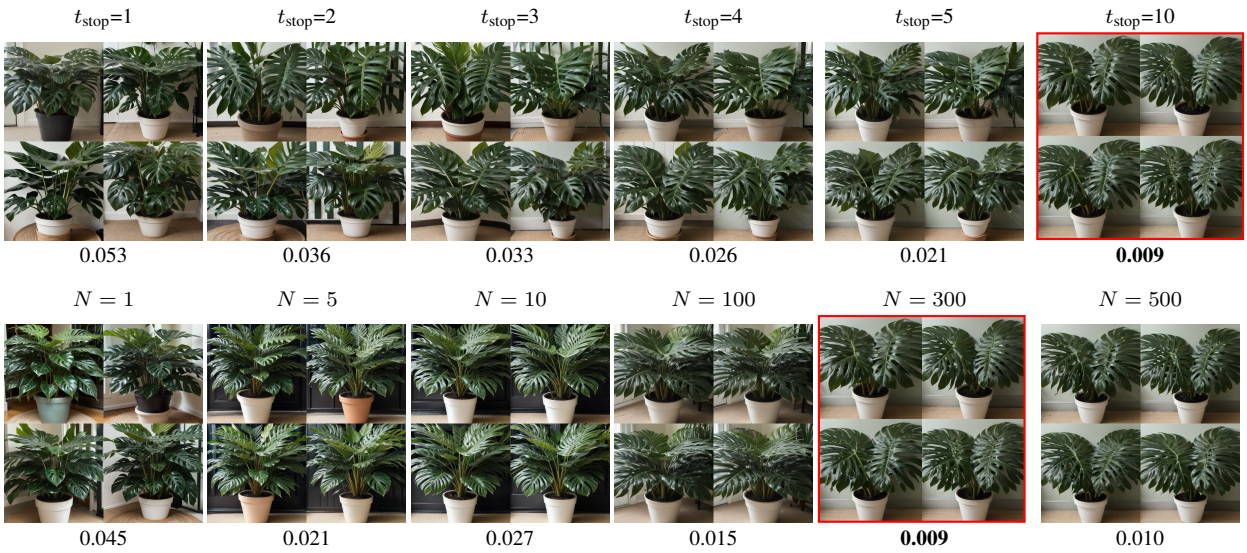


Figure 10. **Efficiency trade-off.** We vary t_{stop} and N, and show consistency scores (↓). Red is our default setting. Lower t_{stop} and N trade consistency for speed and are up to the user.

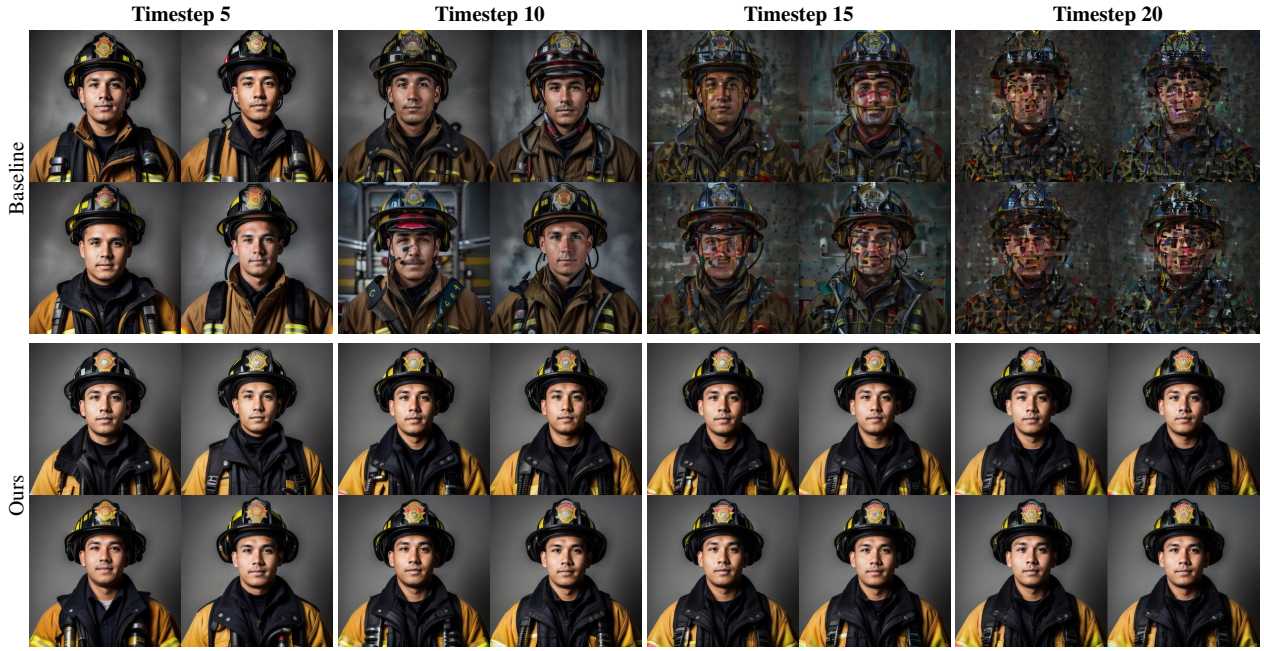


Figure 11. **Comparison with the pre-computed activation baseline.** The pre-computed baseline (top) degrades over timesteps, while DMA (bottom) remains stable and consistent.

C.5. Effects of Classifier-Free Guidance Scale

We examine the effects of different Classifier-Free Guidance (CFG) scales across various concepts in Figure 16. Low CFG values (1.0–5.0) produce structurally invalid results, such as distorted *monstera* leaves, while excessively high values (e.g., 12.0) lead to oversaturated images. The optimal CFG, however, varies across concepts: higher values benefit high-variation concepts like *monstera*, whereas lower values suffice for low-variation concepts, such as *zebra*. In this paper, we use a CFG scale of 7.0 for all concepts.

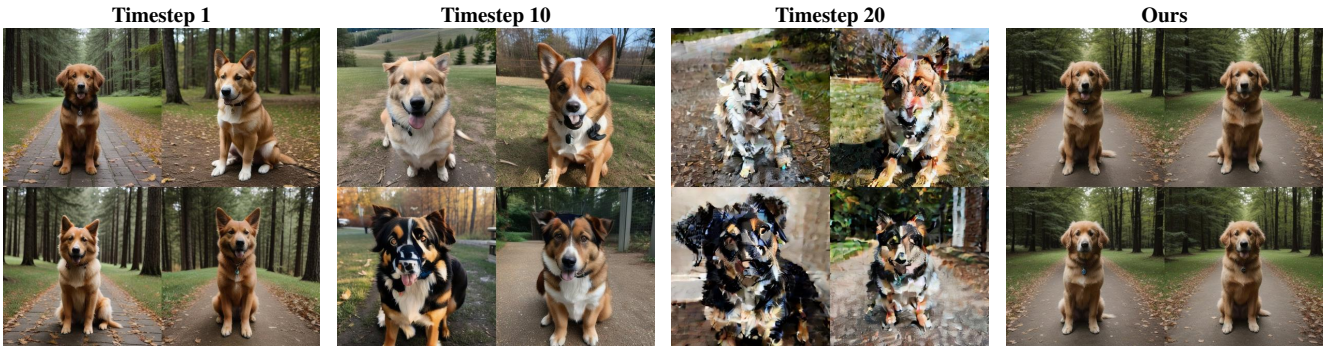


Figure 12. **Comparison with the single-timestep optimization baseline.** This baseline produces inconsistent results and introduces noticeable artifacts if optimization is done in later timesteps. In contrast, our method maintains both semantic consistency and visual quality.

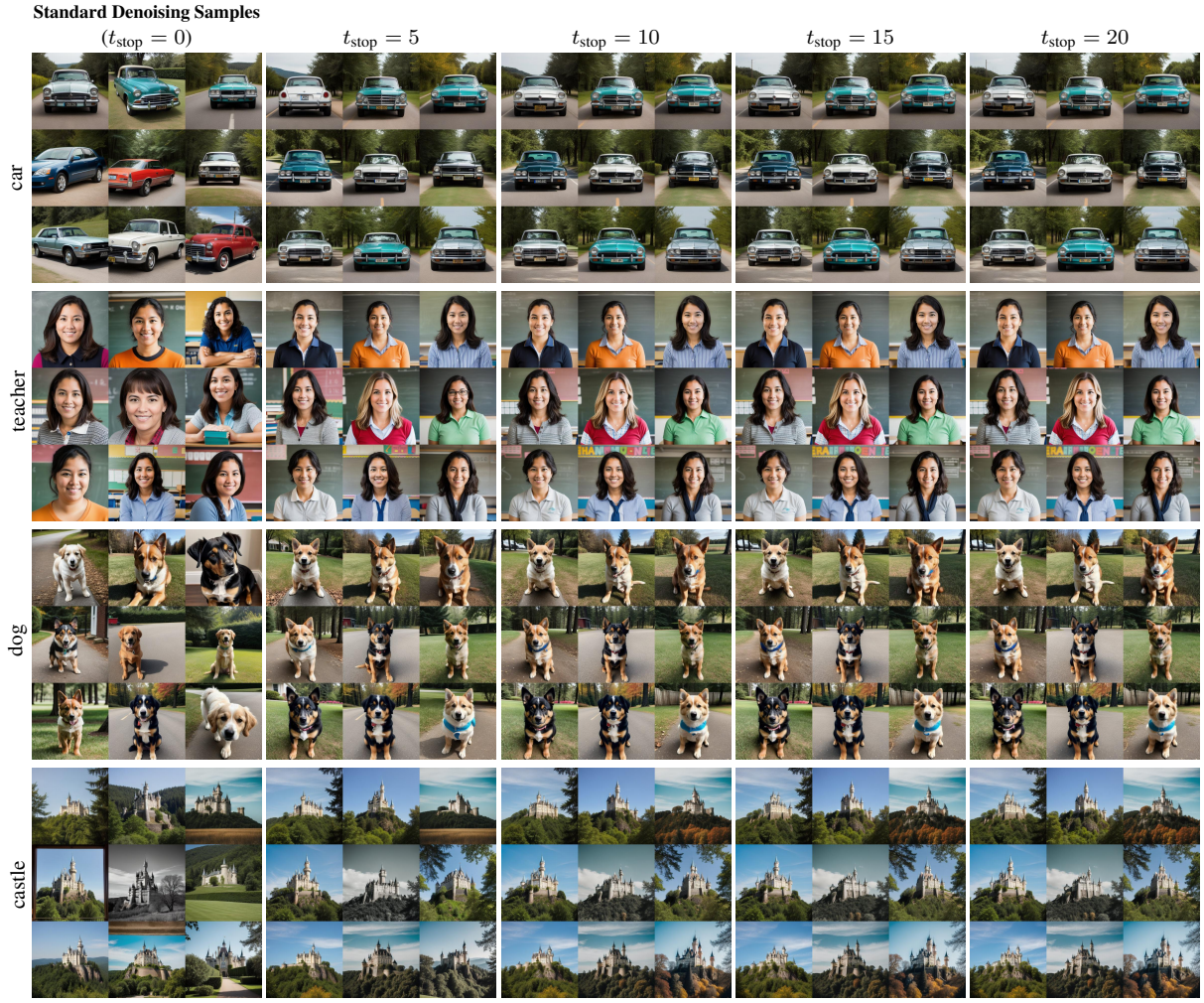


Figure 13. **Results of the *replacement baseline* with different t_{stop}.** This baseline yields inconsistent images, even with high t_{stop} values.

C.6. Effects of Number of Noisy Latents (K)

We analyze how the number of initial noisy latents, K, affects the variability of the resulting averages. For each value

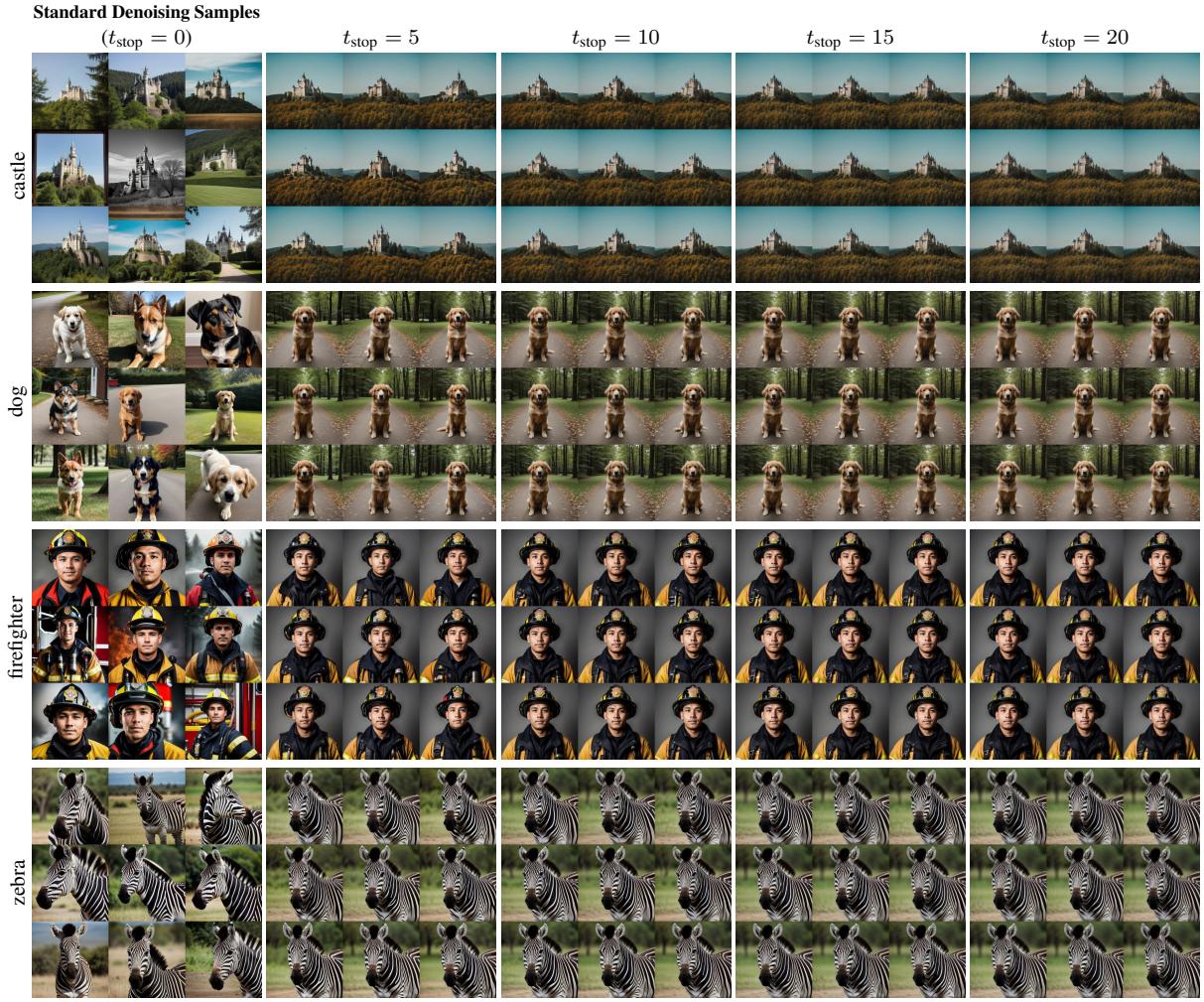


Figure 14. **Effects of different t_{stop} values on visual consistency.** $t_{\text{stop}} = 10$ yields consistent images with reduced computational cost.

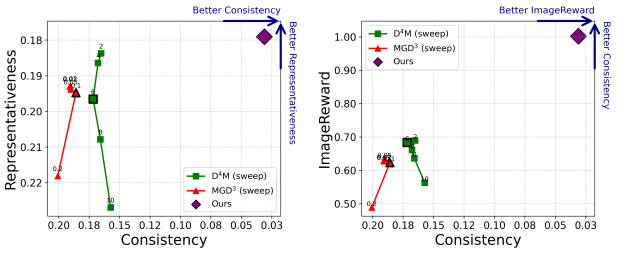


Figure 15. **Quality trade-off across baseline hyperparameters.** Across the full range of hyperparameter settings for each baseline, our method consistently achieves superior representativeness, consistency, and visual quality.

of K, we sample four disjoint sets of K noisy latents, and compute an average image for each set under the same concept prompt. As shown in Figure 17, the variation among the four averages decreases as K increases, consistent with expected statistical behavior. We find that $K = 1{,}000$ is sufficient for most concepts to appear visually consistent, whereas smaller K often yields noticeable differences across sets.

C.7. DiT Semantic Layer Selection

In Figure 18, we experiment with using the output of different transformer blocks for DMA. Preliminary results indicate that the final block produces the most consistent results. While these findings demonstrate the feasibility of extending DMA to DiT, this block-selection strategy is likely suboptimal, as DiT contains many other components that may provide more meaningful latent representations. A more comprehensive analysis is needed to understand how transformer-based models behave.

D. Implementation Details of D^{4}M and MGD^3

Both **D^{4}M** [89] and **MGD^3** [13] are dataset-distillation methods that compress a full dataset into a small set of

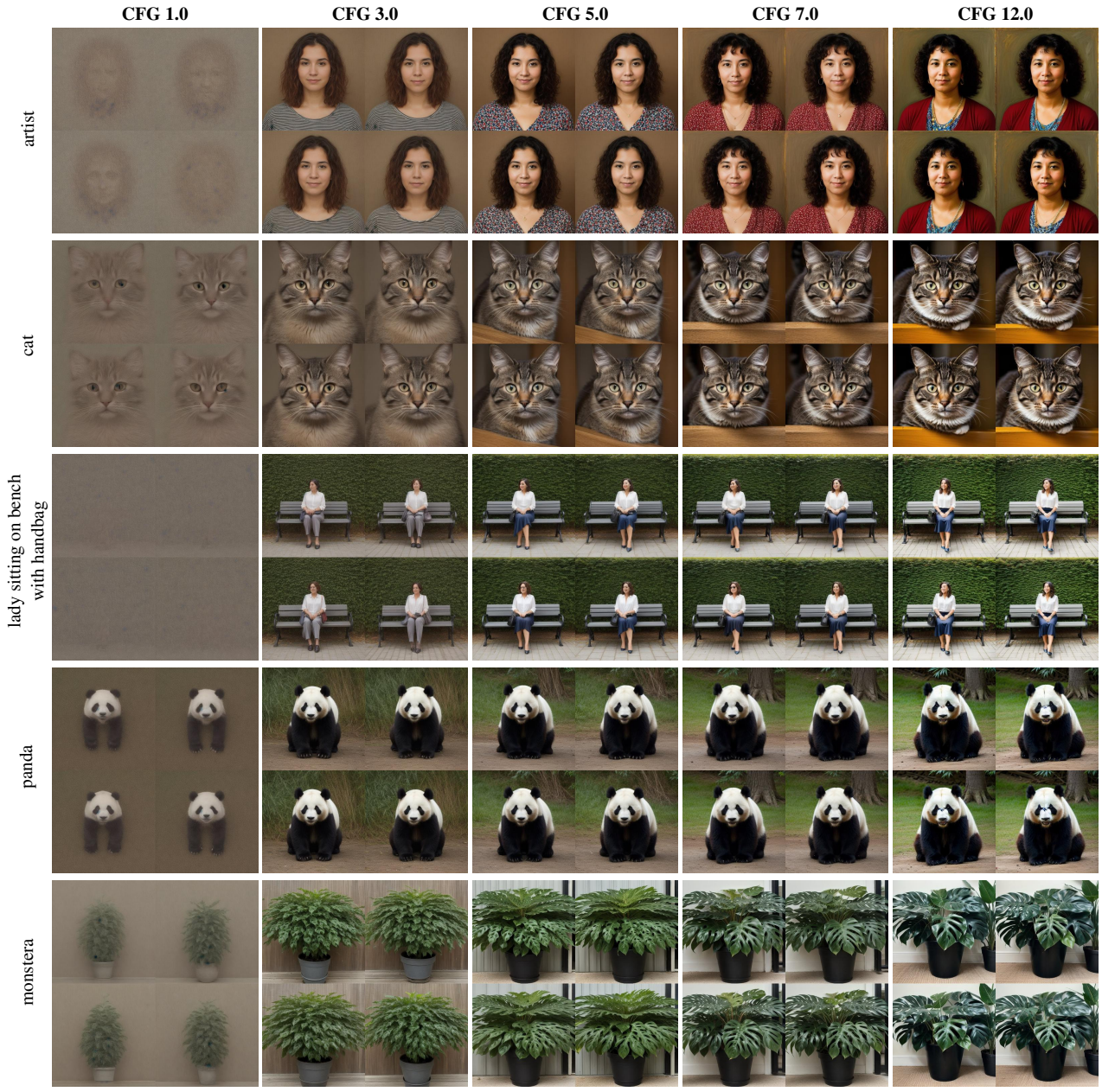


Figure 16. **Effects of CFG scale on visual quality.** Low CFG scales fail to generate faithful structures (as seen in *monstera*), while excessively high CFG scales lead to oversaturated colors and unnatural contrast.

synthetic prototypes. Unlike earlier approaches that train a classifier in the loop, they generate representatives directly from a pre-trained diffusion model, without extra optimization or task-specific losses, and benefit from the model's generative prior to produce higher-quality results. Both encode the dataset using a VAE, cluster the latents into w clusters (where w corresponds to the number of output prototypes per class (IPC)), and use the cluster centroids as prototype latents for guiding synthesis. **D^{4}M** adds noise to each centroid and denoises it (as in SDEdit [61]), whereas **MGD^3** introduces *Mode Guidance*, using the centroid as an attractor that steers the denoising trajectory at every timestep.

In our experiment in the main paper, we adapt **D^{4}M** and **MGD^3** to evaluate how well they can produce an *average image* of a diffusion model. Given generated samples from a single class, we treat them as the "dataset to distill" and distill them into a single prototype (IPC = 1). We re-

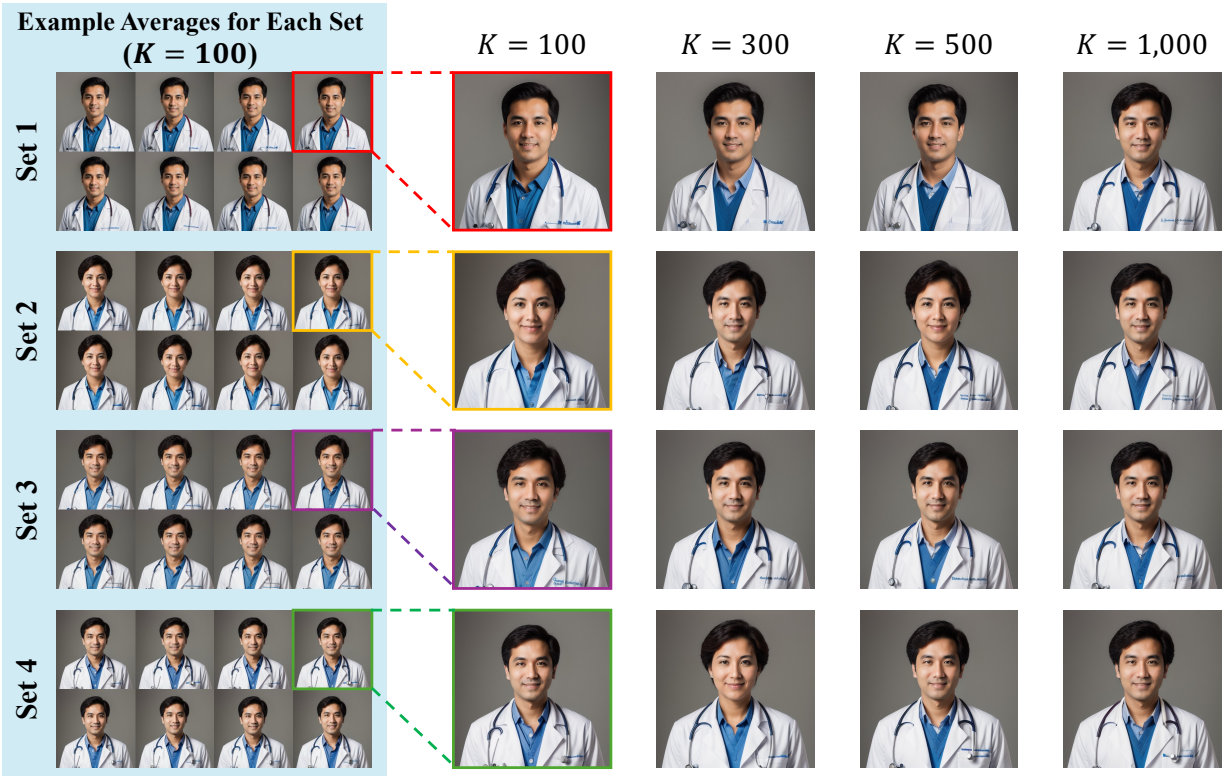


Figure 17. **Effect of the number of noisy latents K.** In the blue box, we sample four distinct sets of $K = 100$ noisy latents and compute their averages. For each set, we show eight example averages (in a 4x2 grid), each computed from a random latent in that set, which converge to similar-looking images *within each set*. However, at $K = 100$, noticeable variations are still observed *across the four sets* shown in the images with colored borders. As K increases, averages across distinct sets become more similar, consistent with the statistical behavior of larger sample sizes.

implemented both baselines on the same Stable Diffusion model as ours, using the same VAE encoder/decoder, and a 20-step DDIM sampler with classifier-free guidance scale 7.0.

D^{4}M [89]. We implement D^{4}M following Su *et al*. [89], adapting it to our single-class setting in which all samples belong to the same class. Since we aim to distill the class into one representative image (IPC = 1), the cluster centroid simplifies to the **average VAE latent** computed over all samples. D^{4}M injects noise into this prototype at diffusion timestep t using the forward diffusion step, followed by a denoising loop to synthesize the final representative image. For a fair comparison, we sweep the noise-injection timestep $t \in \{2, 4, 6, 8, 10\}$, where the main-paper configuration $t = 6$ corresponds to the recommended SDEdit strength of 0.7. Results are shown in Figure 19.

MGD^3 [13]. We implement MGD^3 following Algorithm 1 in Chan-Santiago *et al*. [13], adapting it to our single-class setting in which all samples belong to the same class. In this case, the estimate prototype m reduces to the **average VAE latent**. Starting from noisy latents, at each denoising step t, we compute the DDIM prediction of the clean latent $\hat{\mathbf{z}}_t^{(T)}$ and form the mode-guidance direction $\mathbf{g}_t = \mathbf{m} - \hat{\mathbf{z}}_t^{(T)}$. The predicted noise is then modified according to Algorithm 1:

$$\hat{\epsilon} \leftarrow \epsilon_\theta(\mathbf{z}, t, c) \; - \; \sqrt{1 - \bar{\alpha}_t}\, \lambda\, \mathbf{g}_t.$$

For a fair comparison, we sweep the guidance weight $\lambda \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$ using 10 guided steps, as recommended in the paper; the main-paper setting uses $\lambda = 0.1$. Results are shown in Figure 20.

Comparison. Figure 15 offers a more comprehensive comparison by sweeping hyperparameters, yielding curves that show how each baseline's performance varies across the tradeoff space. Our method outperforms both D^{4}M and MGD^3 in terms of consistency. While these baselines approach our performance in representativeness at certain hyperparameter settings, those settings result in significantly reduced consistency, demonstrating that our method achieves a superior overall tradeoff.

E. Additional Results

E.1. Quantitative Scores

We report a detailed breakdown by concept category of the scores from Table 1. We report representativeness scores in Table 4 and consistency scores in Table 3. As shown in the tables, our method surpasses the baselines across all categories.

DiT quantitative scores. We evaluate our method on 32 random classes from ImageNet with 1,000 samples per class. Since MGD^3 originally supports the DiT architecture, we use it as our baseline and adopt the default hyperparameters from their paper [13]. We report the results in Table 2, showing that our method consistently outperforms MGD^3 across all metrics.

Table 2. Quantitative comparison on DiT. (CLIP / LPIPS / DreamSim)

Method	Representativeness (↓)	Consistency (↓)	ImageReward (↑)
MGD^3	0.165 / 0.678 / 0.364	0.147 / 0.610 / 0.317	-0.8151
DMA (Ours)	0.151 / 0.626 / 0.318	0.050 / 0.192 / 0.093	-0.7085

E.2. Qualitative Results

Average Images. Additional results for average images of more concepts are shown in Figure 21-24.

Mode Discovery. Additional results for mode discovery using DMA with LoRA are shown in Figure 25-34.

F. Potential Negative Impacts

DMA is designed to reveal how a diffusion model internally represents a concept; any harmful biases present are inherent to the probe model and not a direct result of our method. Nonetheless, there are risks in how DMA representations are interpreted. For example, computing only a single or a few averages may marginalize minority modes. Using these averages as authoritative summaries or to stereotype groups or attributes can be misleading and harmful. To mitigate these concerns, we emphasize that DMA averages reflect only the biases of the specific probe model and should not be interpreted as ground-truth representation of any real population or concept.

Table 3. **Consistency (↓)** across concept categories. Lower values indicate higher consistency of prototypes within each seed. Each category includes representative concepts: *Animal* — Bird, Cat, Dog; *Person* — Astronaut, Doctor, Firefighter; *Object* — Bicycle, Car, TV Monitor; *Abstract* — Anger, Freedom, Poverty.

	Animal			**Person**			**Object**			**Abstract**		
Method	CLIP↓	DreamSim↓	LPIPS↓	CLIP↓	DreamSim↓	LPIPS↓	CLIP↓	DreamSim↓	LPIPS↓	CLIP↓	DreamSim↓	LPIPS↓
GANgealing [70]	0	0	0	0	0	0	0	0	0	–	–	–
Avg VAE	0	0	0	0	0	0	0	0	0	0	0	0
D^4M [89]	0.123	0.191	0.615	0.127	0.174	0.510	0.193	0.308	0.627	0.229	0.425	0.534
MGD^3 [13]	0.122	0.224	0.679	0.131	0.202	0.598	0.215	0.342	0.686	0.251	0.508	0.610
DMA (Ours)	0.015	0.021	0.138	0.022	0.019	0.118	0.040	0.039	0.158	0.047	0.049	0.101

Table 4. **Representativeness (↓)** across concept categories. Lower values indicate closer alignment between the prototype and the overall concept distribution.

	Animal			**Person**			**Object**			**Abstract**		
Method	CLIP↓	DreamSim↓	LPIPS↓	CLIP↓	DreamSim↓	LPIPS↓	CLIP↓	DreamSim↓	LPIPS↓	CLIP↓	DreamSim↓	LPIPS↓
GANgealing [70]	0.310	0.394	0.839	0.523	0.599	0.866	0.323	0.439	0.848	–	–	–
Avg VAE	0.505	0.867	0.831	0.504	0.780	0.796	0.464	0.832	0.882	0.420	0.746	0.711
D^4M [89]	0.158	0.320	0.708	0.144	0.230	0.613	0.224	0.379	**0.714**	0.260	0.522	0.652
MGD^3 [13]	0.155	0.310	0.718	**0.140**	0.230	0.636	0.223	0.375	0.724	0.262	0.543	0.671
DMA (Ours)	**0.143**	**0.307**	**0.669**	**0.142**	**0.208**	**0.595**	**0.202**	**0.368**	**0.718**	**0.229**	**0.479**	**0.638**

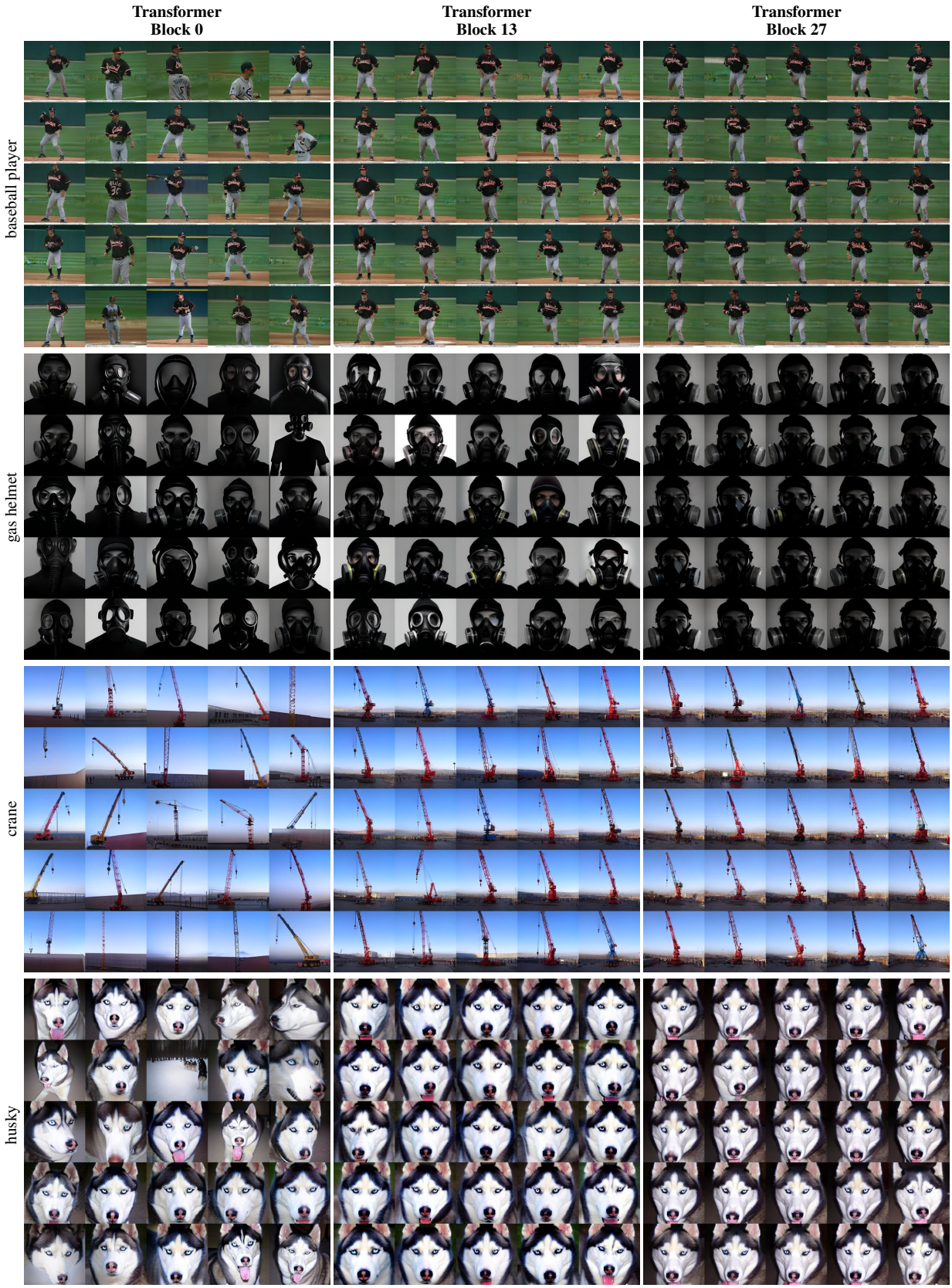


Figure 18. **Ablation on DiT transformer block selection for DMA.** See details in Section C.7.

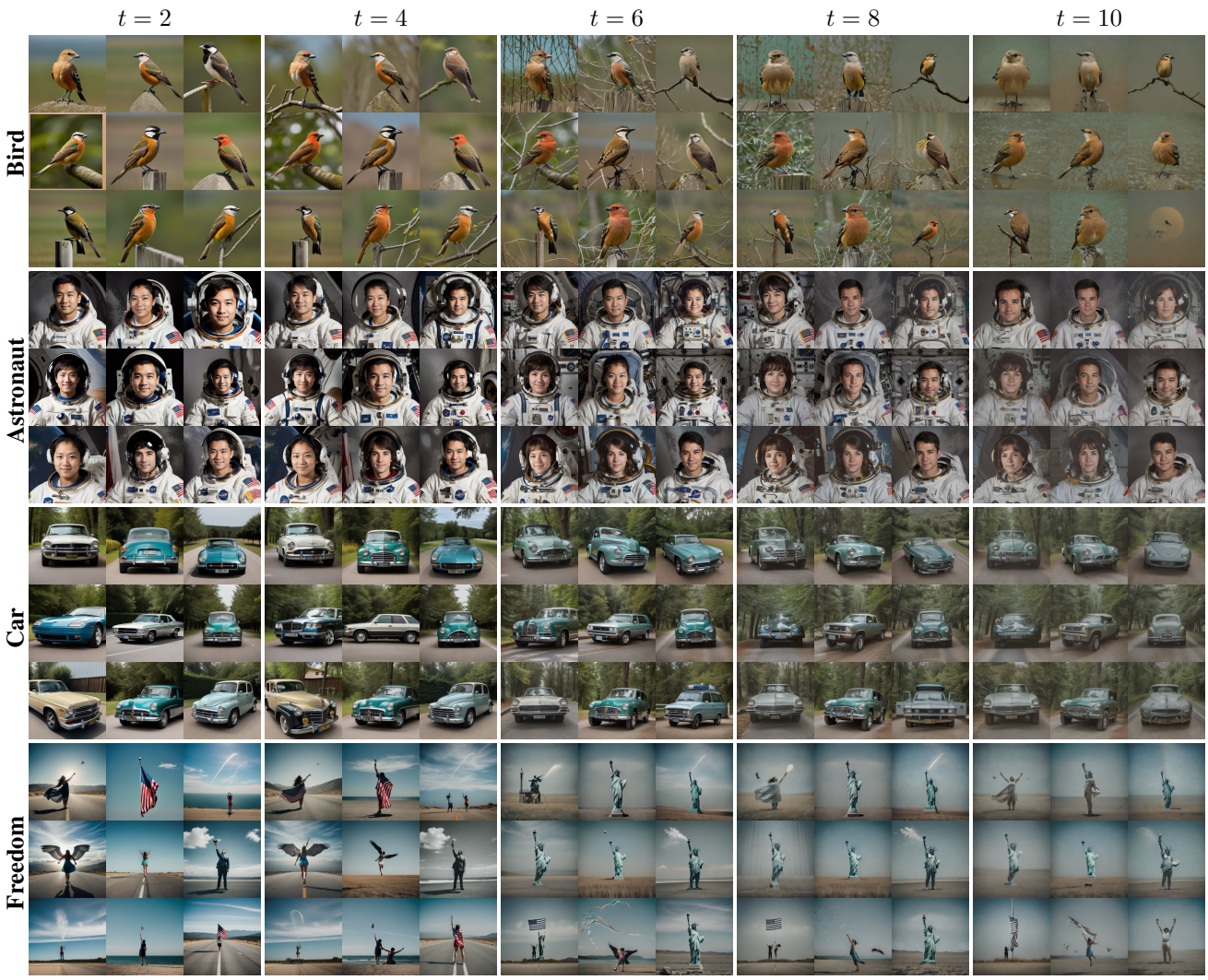


Figure 19. **Hyperparameter tuning of D^{4}M for fair comparison.** We sweep the noise-injection timestep $t \in \{2, 4, 6, 8, 10\}$ across four concepts: *bird*, *astronaut*, *car*, and *freedom*. Larger values of t preserve the averaged latent more strongly but lead to blurrier outputs, while smaller values make the result less consistent.

Figure 20. **Hyperparameter tuning of MGD^3 for fair comparison.** We sweep the mode-guidance weight $\lambda \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$ across four concepts: *bird*, *astronaut*, *car*, and *freedom*. Larger λ increases attraction toward the estimated mode, improving structure but sometimes introducing over-sharpening or hallucinations.

Figure 21. **Average images of *person*, *house*, and *cloth* from different ethnicities.**

Figure 22. **Average images of a *person* with different attributes.**

Figure 23. **Average images of additional concepts with two different attributes.**

Figure 24. **Average images of various concepts.**

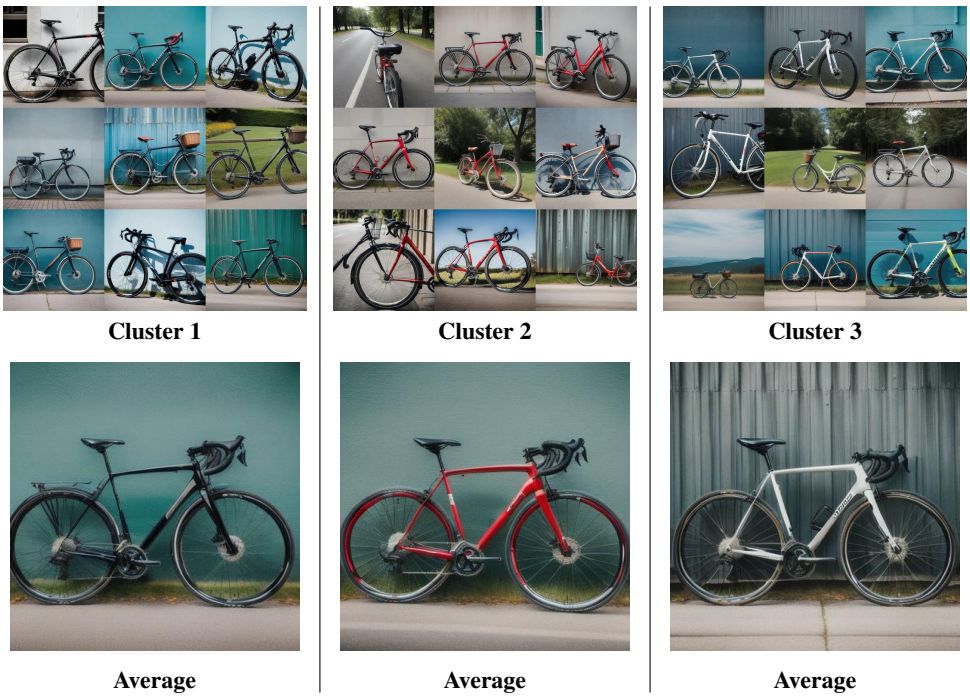


Figure 25. **Cluster averages for the *Bicycle* concept grounded by *color*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *color*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

Figure 26. **Cluster averages for the *Cake* concept grounded by *type*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *type*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

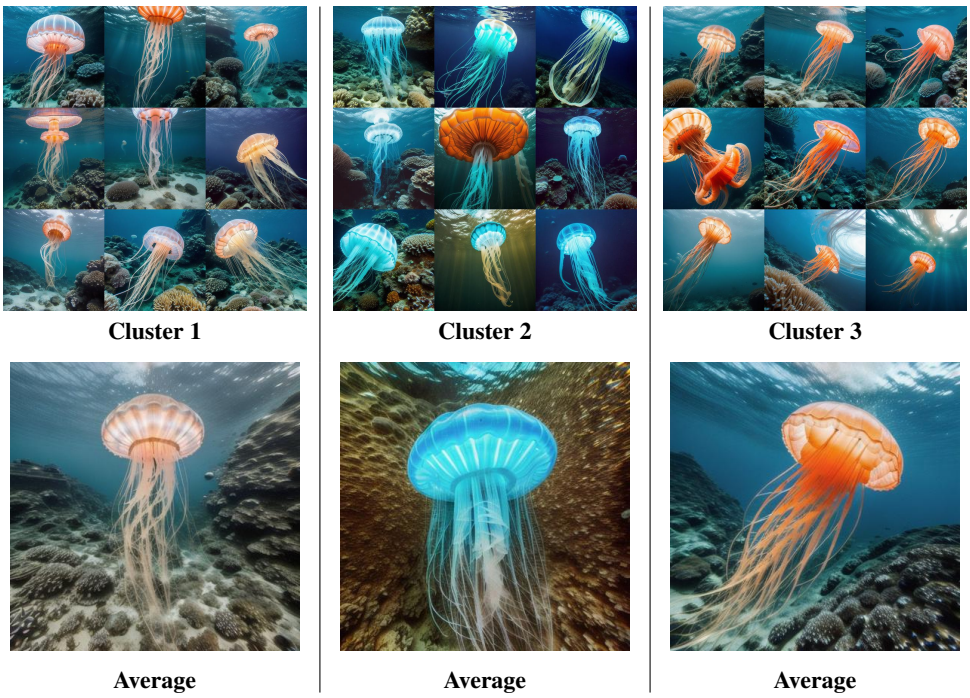


Figure 27. **Cluster averages for the *Jellyfish* concept grounded by *color*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *color*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

Figure 28. **Cluster averages for the *Toy* concept grounded by *shape*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *shape*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

Figure 29. **Cluster averages for the *Nurse* concept grounded by *ethnicity*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *ethnicity*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

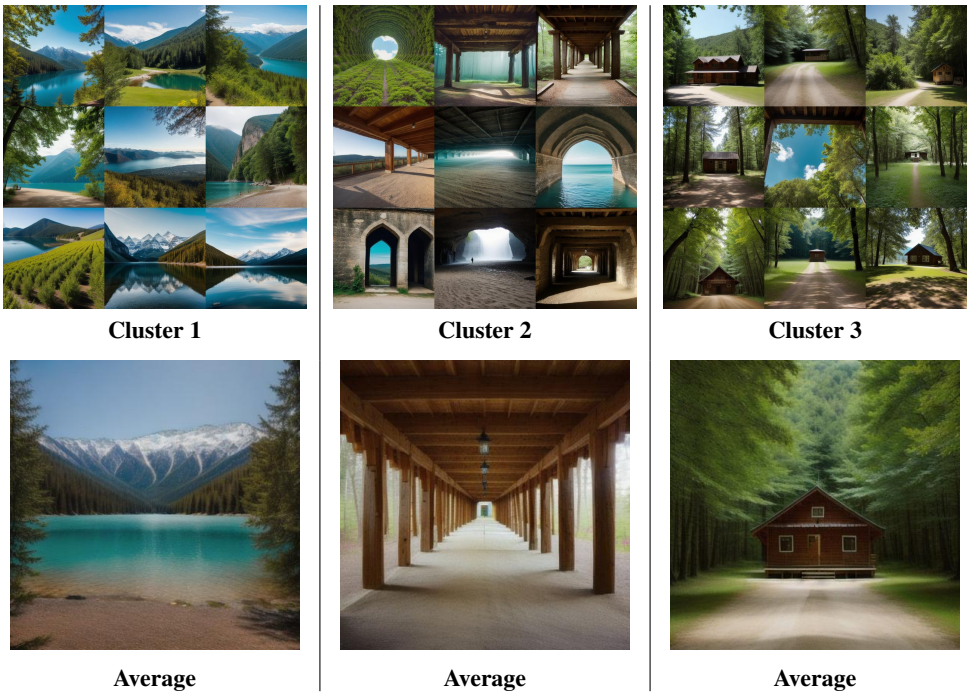


Figure 30. **Cluster averages for the *Place* concept grounded by *environment*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *environment*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

Figure 31. **Cluster averages for the *Object* concept grounded by *shape*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *shape*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

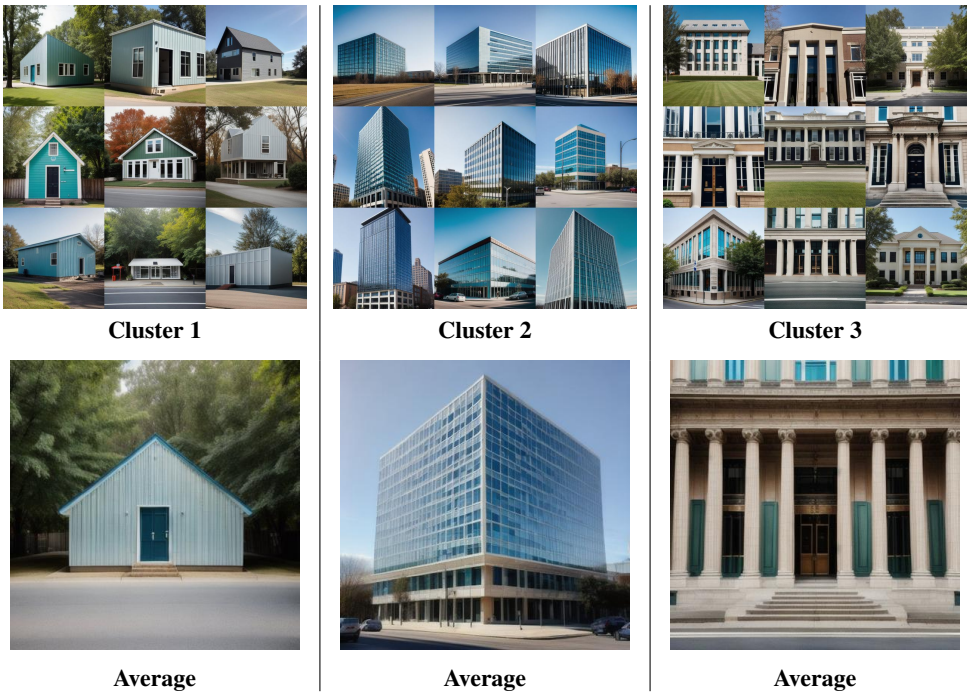


Figure 32. **Cluster averages for the *Building* concept grounded by *architecture*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *architecture*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

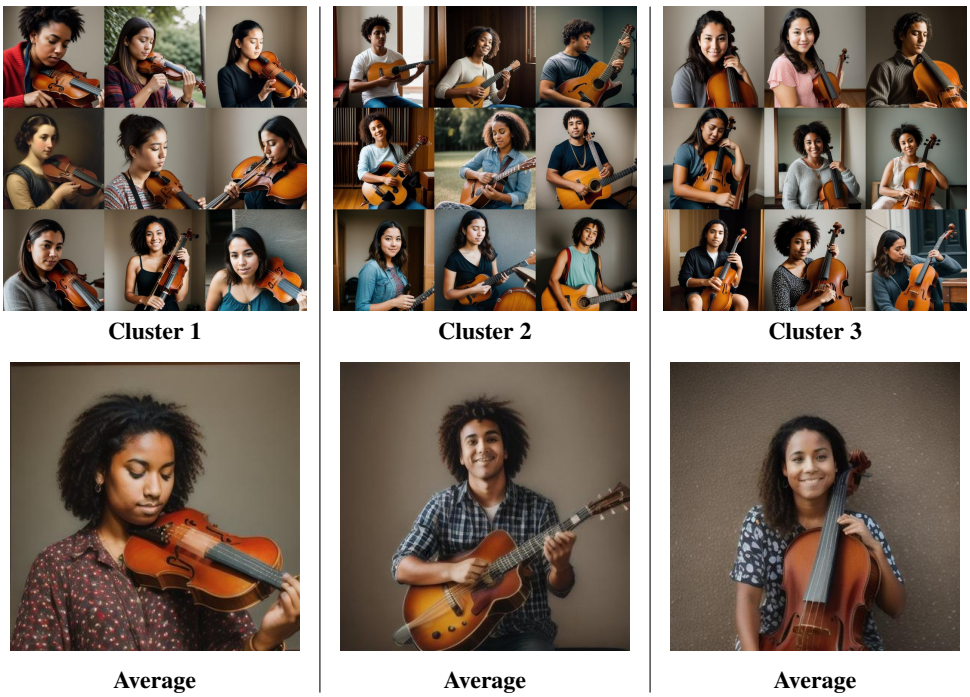


Figure 33. **Cluster averages for the *Person with musical instrument* concept grounded by *instrument*.** Top: Diffusion samples grouped into clusters according to the grounding attribute *instrument*. Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

Figure 34. **Cluster averages for the *Person with pet* concept grounded by *pet* .** Top: Diffusion samples grouped into clusters according to the grounding attribute *pet* . Bottom: Cluster averages computed by DMA with LoRA. These averages highlight how our method captures the dominant visual characteristics within each cluster.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 35. **Baseline comparison for the *astronaut* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 36. **Baseline comparison for the *firefighter* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 37. **Baseline comparison for the *doctor* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 38. **Baseline comparison for the *bird* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 39. **Baseline comparison for the *cat* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 40. **Baseline comparison for the *dog* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] | D^4M [89] | MGD^3 [13] | Ours

Figure 41. **Baseline comparison for the *bicycle* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^4M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 42. **Baseline comparison for the *car* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.

Generated Samples

GANgealing [70] D^{4}M [89] MGD^3 [13] Ours

Figure 43. **Baseline comparison for the *tvmonitor* concept.** Top: Generated samples from the diffusion model. Bottom: Average images produced by GANgealing [70], D^{4}M [89], MGD^3 [13], and our method.