Supplement
Unified Control for Inference-Time Guidance of Denoising Diffusion Models

Maurya Goyal[1], Anuj Singh[1,2], Hadi Jamali-Rad[1,2]

[1]Delft University of Technology, The Netherlands
[2]Shell Global Solutions International B.V., Amsterdam, The Netherlands

A. Related Works

A.1. Image Generation via Diffusion Models.

Early efforts on image generation relied on GANs [1] and VAEs [2, 3]. However, these approaches suffer from problems such as training instabilities, limited representational power, and mode collapse. This resulted in the generation of highly smooth images for VAEs [4–6] or a lack of sample diversity and control for GANs [7, 8]. Diffusion models tackle this by gradually transforming noise into clean samples through a learned denoising process iteratively, preserving both the fidelity and diversity of the target distribution [9–11]. This makes them particularly well-suited for high-quality and diverse image-generation tasks.

A.2. Conditional Image Generation.

Image generation models are typically trained on large datasets which enable generating diverse and high-quality images. To provide users with greater control over the generated content, conditional image generation is used. Rather than sampling from the marginal distribution $p(x)$ conditional methods sample from $p(x|c)$, where c represents the conditioning signal such as class labels, textual prompts, images, or other forms of guidance. Diffusion models are particularly well-suited for conditional image generation due to their score-based formulation, which estimates the gradient of the data distribution's log density [12]. This formulation naturally accommodates the integration of conditioning signals into the generation process. Conditional image generation can be further classified into two broad categories: training-based and training-free.

A.3. Training-Based Methods.

This includes methods that train the model with conditioning signals (which can be guided at inference time for better performance using methods like classifier-free guidance (`CFG`)[13]) or fine-tuning-based methods that align the pre-trained foundational model to the specific conditioning. Aligning pre-trained models through fine-tuning is used a lot not just in vision but also for language [14, 15]. For diffusion models, there are several ways of doing that - direct backpropagation [16, 17], RL-based fine-tuning [18, 19], preference-based supervised fine-tuning [20, 21], domain adaption [22], etc. These methods are not scalable and require a lot of computation for fine-tuning each different control signal, therefore, we look into training-free inference-time alignment.

A.4. Training-Free Methods.

These methods leverage expected rewards during the denoising process to perturb or select optimal samples at **inference time**. This guidance can be categorized into two main types: gradient guidance and sampling-based guidance.

A.5. Gradient Guidance.

[23–28] They leverage the expected predicted sample $x_{0|t}$ at each denoising step and compute the gradient of the reward for this estimate. This gradient guides the denoising trajectory toward regions in the sample space that are expected to give higher rewards, effectively *guiding* the generative process to produce samples with improved rewards.

A.6. Sampling-Based Guidance.

[29–32] Instead of relying on gradient information, they generate multiple candidate samples at each denoising step and estimate the expected reward for each. This allows them to identify promising directions in the sample space and explore more of those areas in the denoising process, thereby guiding the generation toward samples with higher expected rewards.

A.7. Combining Gradient and Sampling-Based Guidance.

TDS [33] was the first to propose a hybrid approach that combines gradient-based optimization with sampling-based exploration. However, their method relies on the Sequential Monte Carlo (SMC) sampling, assuming the gradient to be sampling from the posterior and is evaluated only on relatively simple tasks such as class-conditional generation on MNIST and CIFAR [34], limiting its generalizability. In contrast, our work explores a broader range of tasks by leveraging off-the-shelf reward models without restricting the setting to predefined class labels. Concurrent with our work, DAS [31] extends the TDS framework by introducing a tempering scheme to better explore reward-guided generation. However, it does not consider the blockwise perspective in gradient and sampling guidance, nor does it explore complex scenarios such as (T+I)2I (Text-and-Image- to-Image) guidance.

A.8. Approximating Gradients for Non-Differential Functions.

To ensure that the proposed method remains applicable to both differentiable and non-differentiable rewards, gradient approximations are used to enable perturbations in the direction of reward improvement. Two primary strategies exist for this purpose. The first involves training a surrogate model to approximate the reward function, however, this approach conflicts with the overarching goal of preserving a training-free framework. Therefore, this work adopts the second strategy: zero-order optimization [35, 36], which allows for gradient approximation using only forward evaluations of the reward function, thereby avoiding the need for additional model training.

B. Additional Background

B.1. Noise Based Formulation.

The forward process at each time is defined in Eq. 1 where the β_t represents the variance schedule. $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{t=1}^{T} \alpha_t$ [9]

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t \tag{2}$$

The whole forward process boils down to:

$$q(x_{1:T}|x_{t-1}) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{3}$$

The diffusion model learns the reverse of this forward step:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \beta_t\mathbf{I}), \tag{4}$$

where the $\mu_\theta(x_t, t)$ is defined as:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right). \tag{5}$$

Hence, the UNet [37] given the noisy input (x_t) predicts the noise to be removed to obtain the clean sample (x_0) from Eq. 2 as:

$$\epsilon_\theta(x_t, t) \approx \epsilon_t = \frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}}. \tag{6}$$

B.2. Tweedies Formula.

For classifier-based guidance, we need the value of $p(y|x_t)$, which represents the probability of the desired outcome given a noisy image x_t. However, most reward functions available off the shelf are not designed to handle noisy inputs, rather they expect clean images. To address this issue, there are two possible approaches: either train a new reward function capable of operating directly on noisy images or estimate the clean image from the noisy input using Tweedie's formula.[38]. Since our goal is to develop a training-free method, we choose the second approach. Tweedie's formula lets us estimate the expected clean image given the noisy input and is given as:

$$\hat{x}_0 = \mathbb{E}[x_0|x_t] = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{7}$$

Using Eq. 7 we get the reward value as:

$$r(x_{0|t}, y) \approx r(\mathbb{E}[x_0|x_t], y) = r(\hat{x}_0, y). \tag{8}$$

B.3. Controlled Decoding (CoDe).

To enhance computational efficiency and scalability, CoDe proposes performing this sampling procedure in a blockwise manner, executing it every B step and generating N candidate samples per block. This parallels RL-based objectives where one sample multiple times from a base policy (the base unconditioned diffusion model) and selects the sample best aligned with the reward function.

B.4. SDEdit.

Stochastic Differential Editing [39] is an image synthesis and editing method that generates images aligning with a reference without relying on complex reward models. SDEdit modifies the standard denoising diffusion process by replacing the typical starting point of denoising, T, with $r \times T$, where r is a user-defined percentage of noise $r \in (0, 1)$. This implies that instead of beginning the denoising from pure random noise, the process initiates from an intermediate point. This is achieved by adding a controlled amount of noise to the reference image, effectively mimicking the forward diffusion process. The choice of r directly influences the output: a higher r allows for greater creative freedom, while a lower r enables the preservation of more structural and stylistic properties from the reference image. SDEdit is employed for (T+I)2I (Text-and-Image-to-Image) generation, enabling style transfer while offering users multiple adjustable parameters to control different aspects of the output image.

B.5. Zero-Order Optimization.

Zero-order optimization is used when we have to approximate the $\nabla_x f(x)$ using just the forward pass $f(x)$.

$$\nabla f(x) \approx \frac{1}{N'} \sum_{i=1}^{N'} \frac{f(x + \sigma\epsilon_i) - f(x - \sigma\epsilon_i)}{2\sigma} \epsilon_i, \tag{9}$$

where ϵ is a n-dimensional vector for n-dimensional samples x sampled from $(0, \mathbf{I}_d)$ and σ is a scalar constant. N' is the number of samples the more samples we have the better the estimate at the cost of time and computation.

C. Guidance Rescaling

We rescale the guidance scale using the same mechanism as the one used in FreeDoM [26]. What they do is that they basically guide the model guidance scale times in the direction of the gradient and rescale this scale based on the CFG guidance. Thus they guide the model even more if there is a big difference between the text conditional and the unconditional noise prediction.

$$\text{scale}_{\text{new}} = \frac{\|\text{ correction }\|_2 \cdot \text{scale}_{\text{CFG}} \cdot \text{scale}_{\text{grad}}}{\|\text{ grad }\|_2 + \varepsilon} \tag{10}$$

where correction is just the CFG [13] correction term:

$$\text{correction} = \hat{\epsilon}_\theta(x_t, prompt) - \hat{\epsilon}_\theta(x_t) \tag{11}$$

We found the dynamic rescaling strategy to be the most effective within our evaluation setting and thus adopted it in place of a fixed guidance scale. By normalizing the guidance using the gradient norm and scaling it proportionally to the correction norm it reduces the sensitivity to the choice of guidance scale. Therefore, a consistent range of values (generally between 0.2 and 0.6) performs reliably well across different reward models. This not only improved performance stability but also saved considerable time, as it decreased the need to manually tune the guidance scale for each individual reward function checking for a ton of different values based on the reward scale.

D. Algorithms

The algorithms for sampling and gradient based guidance used are as follows:

Algorithm 1 GRAD(z_t, t)

```
Require: latent $z$, timestep $t$
$\hat{x}_0 \leftarrow D(\mathbb{E}_{p_\theta}(z_0|z_t, t))$          ▷ Expected clean sample
$\hat{r}(z_t) \leftarrow r(\hat{x}_0)$          ▷ Compute reward
$g \leftarrow \nabla_{z_t} \hat{r}(z_t)$          ▷ Compute gradient w.r.t. $z_t$
return $g$
```

Algorithm 2 SAMPLE$(\{z_{t-1}^{(n)}\}, r(D(\{\hat{z}_0^{(n)}\})), \tau)$

```
Require: current images $\{z_{t-1}^{(n)}\}_{n=1}^{N}$, reward vector $\mathbf{r} = r(D(\{\hat{z}_0^{(n)}\}))$, temperature $\tau$, empty list $\{z_{temp}^{(n)}\}_{n=1}^{N}$
Compute softmax probabilities:
    $$P_n \leftarrow \frac{\exp(\mathbf{r}_n/\tau)}{\sum_{j=1}^{N}\exp(\mathbf{r}_j/\tau)} \quad \text{for } n = 1, \ldots, N$$
for $i = 1$ to $N$ do
    Sample index $i \sim$ Multinomial$(\{P_n\}_{n=1}^{N})$
    Append $z_{t-1}^{(i)}$ to $\{z_{temp}^{(n)}\}$
end for
return $\{z_{temp}^{(n)}\}_{n=1}^{N}$
```

E. Robustness of UniCoDe Across More Capable Diffusion Models (SD2.1)

We further evaluate whether the efficiency and alignment benefits of UniCoDe hold when applied to a stronger base model, namely Stable Diffusion 2.1. As shown in Table 1, the trends observed with weaker diffusion priors consistently translate to this upgraded setting. While CoDe achieves a slightly higher pickscore reward, UniCoDe requires nearly $4\times$ less sampling time yet remains highly competitive in prompt adherence. More importantly, UniCoDe exhibits lower CMMD, this demonstrates that UniCoDe maintains its core advantages of efficiency, improved prior preservation, and strong prompt alignment even when operating on higher-capacity diffusion models such as SD 2.1.

Method	**Prompt Alignment (T2I)**			
	Pickscore	**CMMD**	**CLIP**	**Time**
SD2.1	1.000	1.000	1.00	1.00
CoDe$_{30}$	**1.092**	4.415	**1.012**	43.39
UniCoDe$_4$	1.091	**3.961**	1.010	**11.88**

Table 1. Results on Stable Diffusion 2.1 under prompt alignment (T2I) guidance. UniCoDe provides comparable prompt adherence with substantially lower computation time and reduced prior deviation compared to CoDe.

F. Hyperparameter Setting

F.1. Image Based Guidance

For the aesthetic reward model, we use the ViT-L/14 as the backbone model for the CLIP. We use a blocksize of 5 for both the sampling and gradient guidance in this case and set the guidance scale to be 0.2. We do the denoising for 500 DDPM steps and the CFG guidance scale is also set to 5. Also, we start adding the gradients from the 0.6 noise ratio i.e. if $T = 1000$ we start adding the gradients from the 600 timestep. We use the schedule $[2, 2, 2, 4, 4, 4, 4, 6, 6, 6]$, which allocates a larger sampling budget toward the later stages of denoising. This design aligns with the intuition that the denoising process progressively refines the sample, moving from coarse to fine details. We selected this particular schedule after ablation studies on several alternatives, as it consistently offered the best performance. This preserves the image structure, doesn't reward hack, and guides it towards achieving a better reward. The evaluation set contains 51 prompts (from the ImageNet evaluation set) and we generate 10 images for each prompt.

For gradient guidance experiments for MPGD and FreeDoM we do the guidance only for noise ratios between 0.7 and 0.3. Thus for timesteps $70-30$ in a 100 step DDIM scheduler. The guidance scales used are 7.5 for MPGD and 0.2 for FreeDoM. Additionally, for FreeDoM, we perform 10 optimization iterations per denoising step. For the UG baseline, we also use 100 DDIM steps, with 6 optimization steps and a forward guidance weight of 30.

F.2. T2I Setting

For the pickscore reward model, we again use the ViT-L/14 as the backbone model for the CLIP. We use a blocksize of 5 for both the sampling and 4 gradient guidance in this case and set the guidance scale to be 0.2. We do the denoising for 500 DDPM steps and the CFG guidance scale is also set to 5. We do the gradient addition during the whole denoising as we also want to alter the structural properties for the alignment to complex prompts. This preserves the image structure, doesn't reward hack, and guides it towards achieving a better reward. The schedule used is $[2, 6, 6, 2, 2, 2, 4, 4, 6, 6]$ as it preserves coarser details like the spatial structure early on and still preserves the quality. The evaluation set contains 50 prompts (from the HPD [40] evaluation dataset) and we generate 10 images for each prompt.

The setting for gradient guidance remains the same except for UG where we increase the weight to 150.

F.3. Multireward Setting

In the multireward setting, we use the same prompts as the Aesthetic case but only a subset of 6 out of the 51 and generate 10 images for each as it takes longer due to the weighted addition of the two reward models. The rest of the settings are the same and we add the gradients during the whole denoising.The values of γ_1 and γ_2 are taken as $(1, 0), (0, 1)$ and we set γ_1 as 1 and change γ_2 in $[2, 3, 5, 10, 15, 20, 25, 30, 50, 70, 100, 150, 200, 250, 300, 350, 400, 450, 500, 750, 1000]$

F.4. (T+I)2I Setting

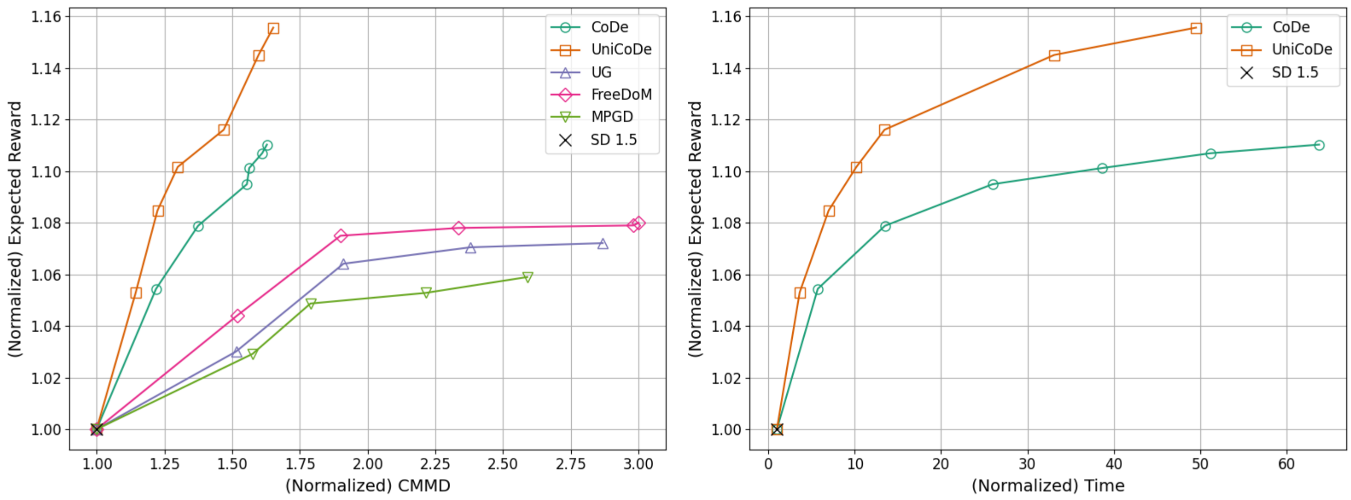
For this, we use the same 50 prompts as the T2I setting, use the three style images as the reference images, and generate 10 images for each prompt and each style. We use the 100 step DDPM scheduler and we set η to be 0.6. The sampling blocksize is 5 and the gradient blocksize is 2 with a guidance scale of 0.4. The rest of the settings are the same.

F.5. Non-Differentiable Reward (Compressibility)

For this experiment, we use a guidance scale of 0.2 and perform an ablation over the number of forward passes, ranging from 1 to 50. We observe that increasing the number of forward passes improves gradient stability; however, even at 50 passes, the gradient remains noisy and computational cost increases significantly. To address this, we set the number of sampling streams to N=35, allowing performance gains to arise more from exploring diverse directions rather than relying solely on gradient information. Evaluation is conducted using four simple prompts: ”monkey”, ”llama”, ”wolf”, and ”butterfly”, generating 10 samples for each case.

G. Ablation Tradeoff Curves for T2I Scenario

For the text-to-image (T2I) guidance scenario using pickscore as the reward, we present ablation trade-off curves by varying the number of samples N for both CoDe and UniCoDe. In the gradient-based guidance scenario, we vary either the guidance scale or the number of recurrent time steps. These experiments are designed to evaluate how the proposed method performs across the overall trade-off frontier. We compare performance across multiple axes, including reward versus divergence, reward versus compute, divergence versus compute, and T-CLIP score versus divergence. The corresponding results are presented in Figures 1a, 1b.



(a) Trade off curves for Reward vs Divergence

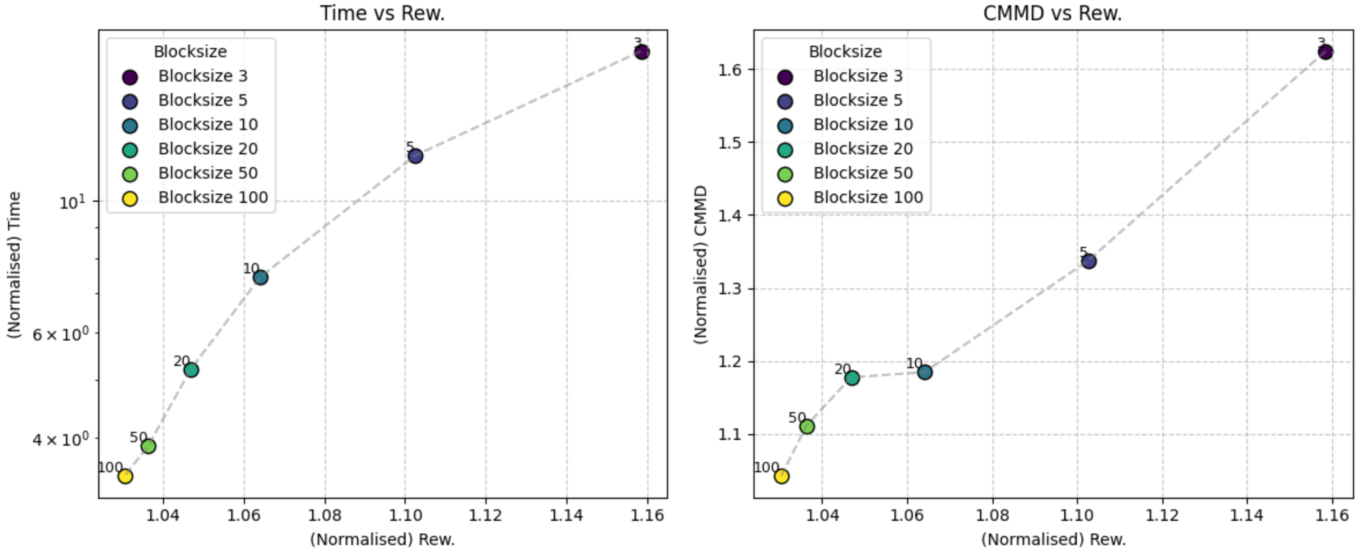
(b) Trade off curves for Reward vs Time

H. Blocksize Hyperparameter Analysis

For all these experiments we use the pickscore reward function in the T2I scenario with a fixed number of samples ($N = 4$).

H.1. Equal Blocksizes

In this experiment, we set both blocksizes equal and analyze the resulting trade-offs, such as Reward vs. Divergence and Reward vs. Time (see Figures 2a, 2b). Decreasing the sampling blocksize (B_s) leads to more aggressive sampling, as selection occurs more frequently, while simultaneously aligning the prior more closely with the posterior by incorporating gradients at finer intervals (decreasing the gradinet blocksize B_g). This improves reward alignment, but comes at the cost of higher divergence and longer runtime.



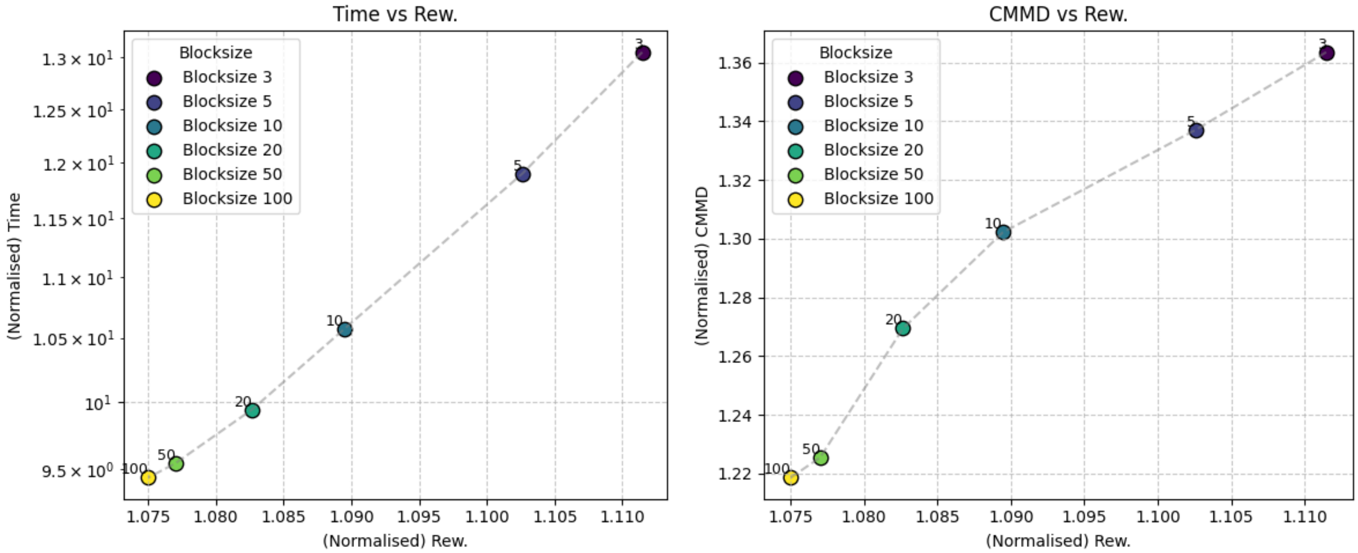
(a) Blocksize Analysis for Time vs Reward

(b) Blocksize Analysis for Divergence vs Reward

H.2. Constant B_g, Varying B_s

For this experiment, we keep the gradient block size equal ($B_g = 5$) and analyze how changing the sampling blocksize affects the performance through the resulting trade-offs, such as Reward vs. Divergence and Reward vs. Time (see Figures 2a,2b).

As expected, decreasing the blocksize leads to more frequent selection and replication of the best sample across all streams. This results in increased exploitation, which manifests as higher rewards, at the cost of higher divergence and increased computational time.

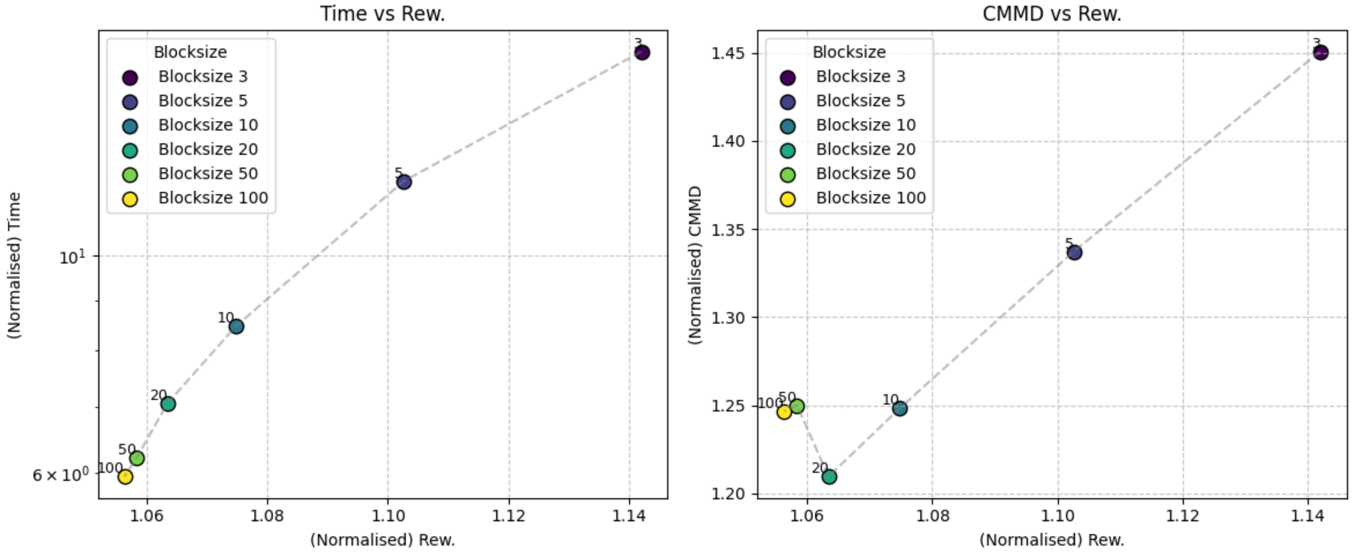


(a) Blocksize Analysis for Divergence vs Reward

(b) Blocksize Analysis for Time vs Reward

H.3. Constant B_s, Varying B_g

For this experiment, we keep the gradient block size equal ($B_s = 5$) and analyze how changing the sampling blocksize affects the performance through the resulting trade-offs, such as Reward vs. Divergence and Reward vs. Time (see Figures 2a,2b). As the blocksize decreases, gradients are incorporated at more timesteps, which aligns the prior more closely with the posterior. This results in improved rewards, but comes at the cost of increased divergence and longer computation time.



(a) Blocksize Analysis for Divergence vs Reward

(b) Blocksize Analysis for Time vs Reward

I. General Guidelines for Setting N, B_g, B_s

UniCoDe depends heavily on the parameters N, B_g, and B_s, which intuitively control the degree of exploration and the extent to which the prior is pushed towards the posterior (via B_g). As with N and B in CoDe [30] Appendix Section G, these parameters influence both reward-alignment and fidelity to the base distribution, as illustrated in Figs. 1-4.

Increasing N enhances exploration, leading to higher reward-aligned generations, but also increases divergence from the prior distribution and computational cost (Figs. 1a and 1b). Similarly, decreasing B_g significantly boosts rewards, as observed in Figs. 4a and 4b, but this comes at the cost of reward hacking, reflected in higher divergence from the prior. B_s affects the frequency of sampling, with lower values increasing reward-alignment by being more aggressive at the cost of raising divergence and computational requirements (Figs. 3a and 3b).

Overall, the interplay between N, B_g, and B_s governs the trade-off between reward maximization, adherence to the base distribution and the computational time.

J. Additional Results

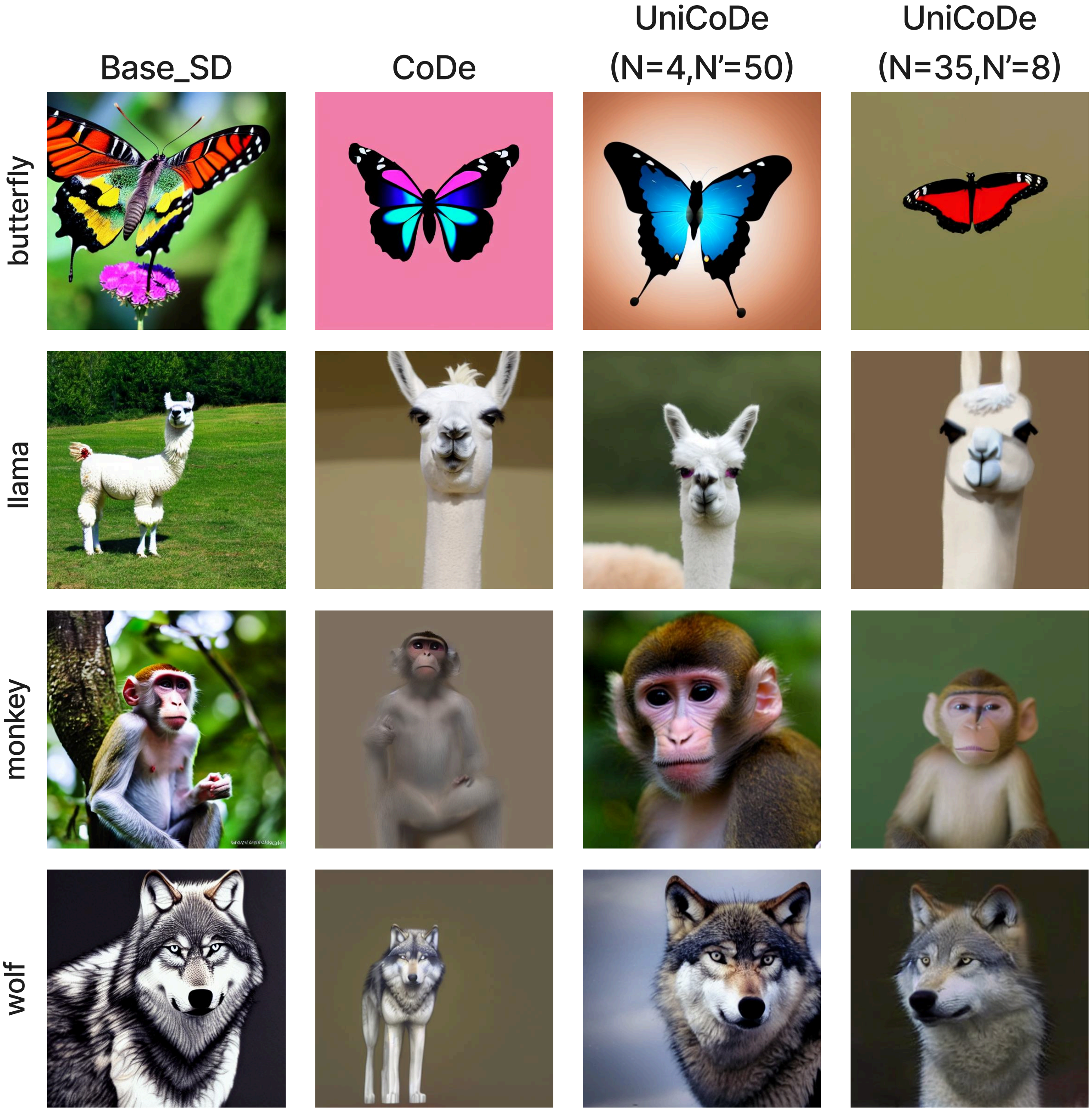


Figure 5. More qualitative samples for the compressibility scenario (non-differentiable)

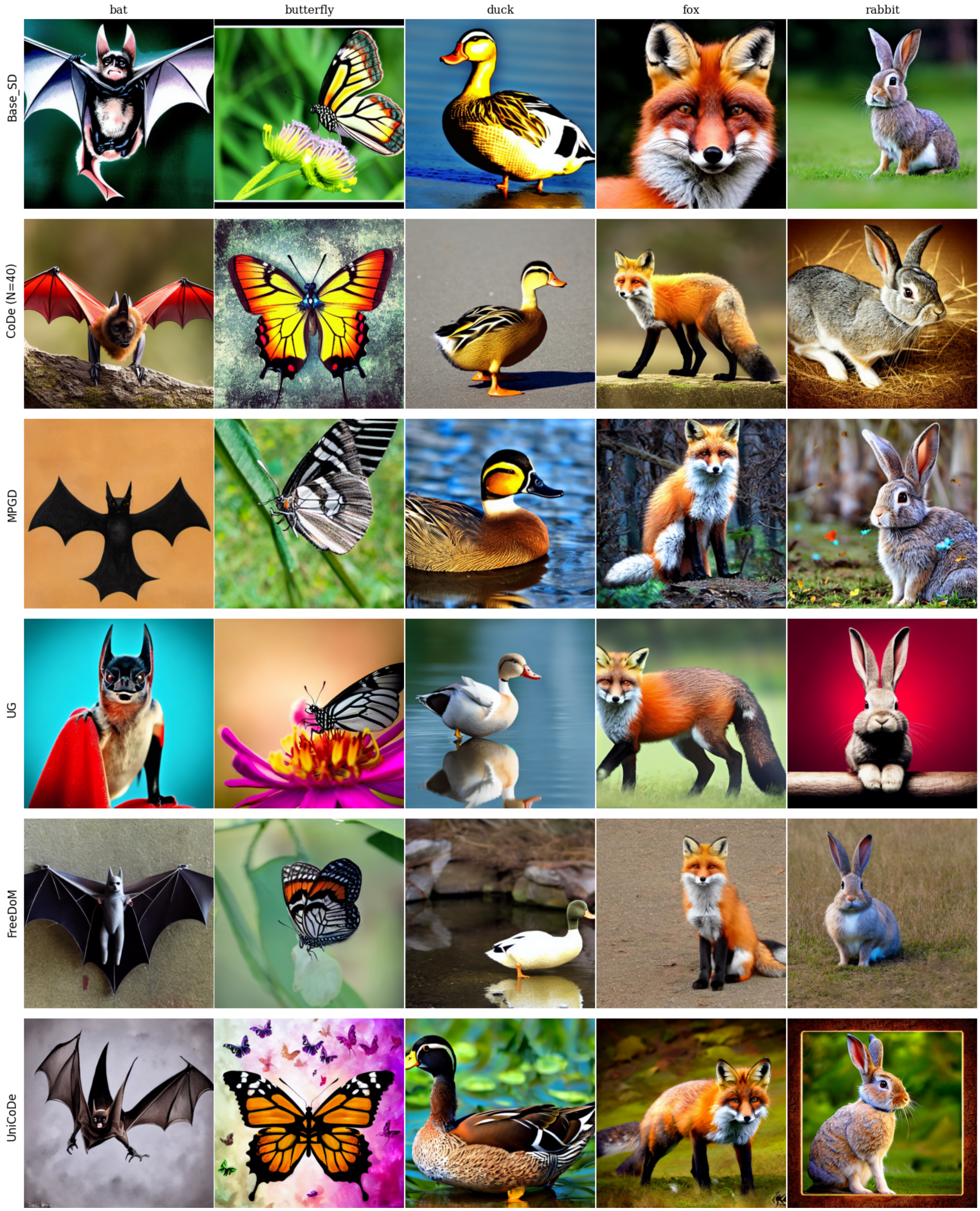


Figure 6. More qualitative examples for aesthetic guidance

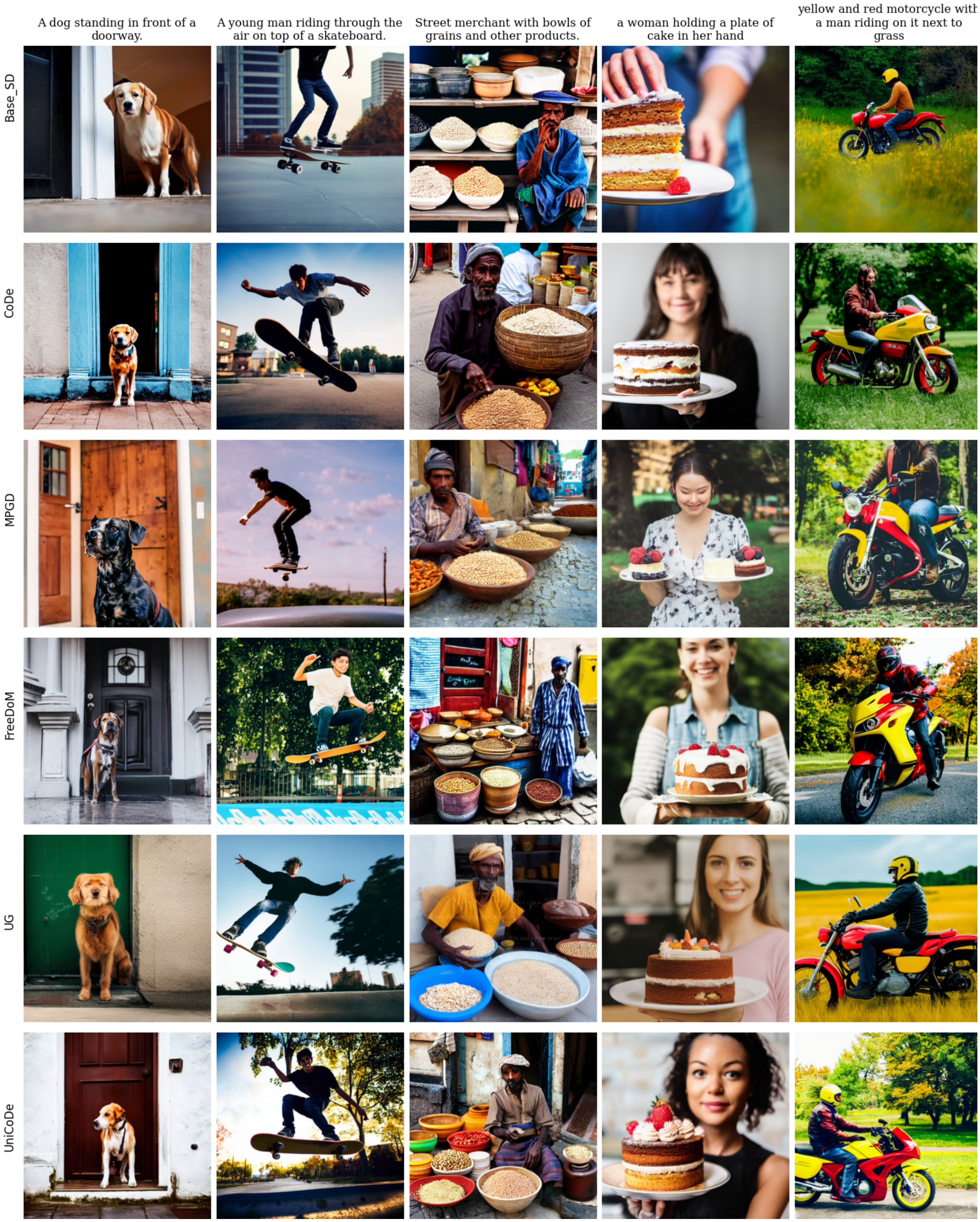

A dog standing in front of a doorway.
A young man riding through the air on top of a skateboard.
Street merchant with bowls of grains and other products.
a woman holding a plate of cake in her hand
yellow and red motorcycle with a man riding on it next to grass
Base_SD
CoDe
MPGD
FreeDoM
UG
UniCoDe

Figure 7. More qualitative examples for T2I pickscore guidance

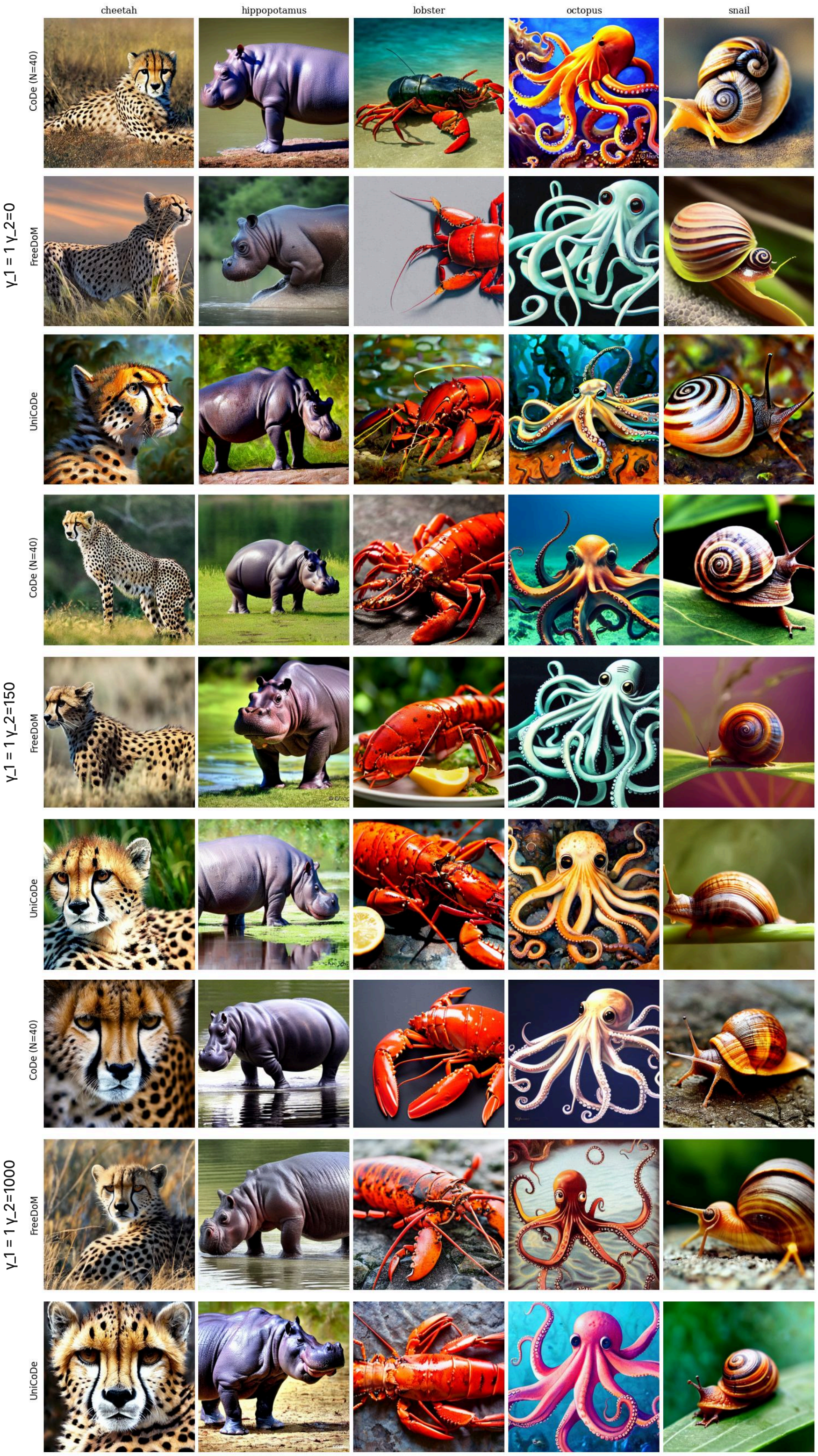


Figure 8. Each column corresponds to a different prompt. Rows are grouped by $\gamma_2 \in \{0, 150, 1000\}$ (top to bottom), with γ_1 fixed at 1. Within each group, the three rows show results from `CoDe`, `FreeDoM`, and `UniCoDe` (top to bottom). The top group focuses more on aesthetic quality, whereas the bottom group prefers Pickscore, and the middle group ($\gamma_2 = 150$) offers a balanced trade-off.

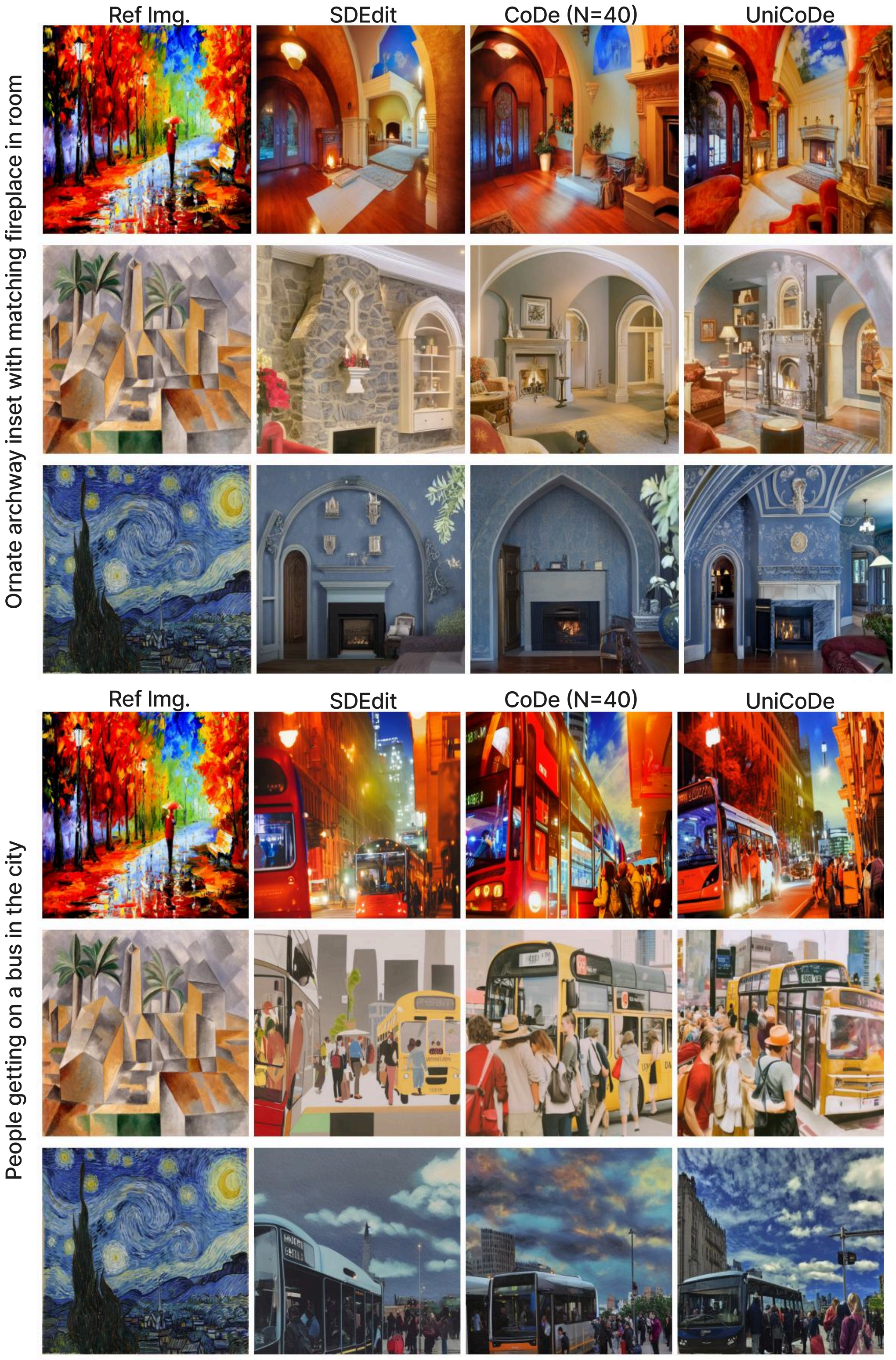

Ref Img.
SDEdit
CoDe (N=40)
UniCoDe
Ornate archway inset with matching fireplace in room
Ref Img.
SDEdit
CoDe (N=40)
UniCoDe
People getting on a bus in the city

Figure 9. More qualitative samples for the (T+I)2I scenario

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[2] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013. 1

[3] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 1

[4] Gustav Bredell, Kyriakos Flouris, Krishna Chaitanya, Ertunc Erdil, and Ender Konukoglu. Explicitly minimizing the blur error of variational autoencoders. *arXiv preprint arXiv:2304.05939*, 2023. 1

[5] Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. *Advances in neural information processing systems*, 31, 2018.

[6] Sanchayan Vivekananthan. Comparative analysis of generative models: Enhancing image synthesis with vaes, gans, and stable diffusion. *arXiv preprint arXiv:2408.08751*, 2024. 1

[7] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017. 1

[8] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] 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*, pages 10684–10695, 2022. 1

[12] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 3

[14] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. 1

[15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1

[16] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. 2023. 1

[17] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023. 1

[18] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023. 1

[19] Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024. 1

[20] Meihua Dang, Anikait Singh, Linqi Zhou, Stefano Ermon, and Jiaming Song. Personalized preference fine-tuning of diffusion models. *arXiv preprint arXiv:2501.06655*, 2025. 1

[21] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 1

[22] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 1

[23] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 1

[24] Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pages 32483–32498. PMLR, 2023.

[25] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023.

[26] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023. 3

[27] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, et al. Manifold preserving guided diffusion. *arXiv preprint arXiv:2311.16424*, 2023.

[28] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Y Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. *Advances in Neural Information Processing Systems*, 37:22370–22417, 2024. 1

[29] Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024. 2

[30] Anuj Singh, Sayak Mukherjee, Ahmad Beirami, and Hadi Jamali-Rad. Code: Blockwise control for denoising diffusion models. *arXiv preprint arXiv:2502.00968*, 2025. 7

[31] Sunwoo Kim, Minkyu Kim, and Dongmin Park. Alignment without over-optimization: Training-free solution for diffusion models. *arXiv preprint arXiv:2501.05803*, 2025. 2

[32] Yuta Oshima, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Inference-time text-to-video alignment with diffusion latent beam search. *arXiv preprint arXiv:2501.19252*, 2025. 2

[33] Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023. 2

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[35] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017. 2

[36] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020. 2

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2

[38] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. 3

[39] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 3

[40] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2096–2105, 2023. 5