Improving GFlowNets for Text-to-Image Diffusion Alignment

Anonymous Authors[1]

Abstract

Diffusion models have become the *de-facto* approach for generating visual data, which are trained to match the distribution of the training dataset. In addition, we also want to control generation to fulfill desired properties such as alignment to a text description, which can be specified with a black-box reward function. Prior works fine-tune pretrained diffusion models to achieve this goal through reinforcement learning-based algorithms. Nonetheless, they suffer from issues including slow credit assignment as well as low quality in their generated samples. In this work, we explore techniques that do not directly maximize the reward but rather generate high-reward images with relatively high probability — a natural scenario for the framework of generative flow networks (GFlowNets). To this end, we propose the **D**iffusion **A**lignment with **G**FlowNet (DAG) algorithm to post-train diffusion models with black-box property functions. Extensive experiments on Stable Diffusion and various reward specifications corroborate that our method could effectively align large-scale text-to-image diffusion models with given reward information.

1. Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have drawn significant attention in machine learning due to their impressive capability to generate high-quality visual data and applicability across a diverse range of domains, including text-to-image synthesis (Rombach et al., 2021), 3D generation (Poole et al., 2022), material design (Yang et al., 2023), protein conformation modeling (Abramson et al., 2024), and continuous control (Janner et al., 2022). These models, through a process of gradually denoising a random distribution, learn to replicate complex data distributions, showcasing their robustness and flexibility. The traditional training of diffusion models typically relies on large datasets, from which the models learn to generate new samples that mimic and interpolate the observed examples.

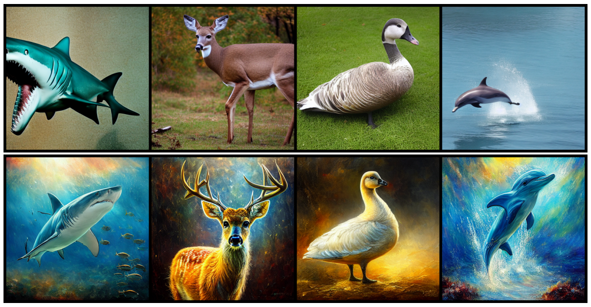


Figure 1: Generated samples before (top) and after (bottom) the proposed training with Aesthetic reward.

However, such a dataset-dependent approach often overlooks the opportunity to control and direct the generation process towards outputs that not only resemble the training data but also possess specific, desirable properties (Lee et al., 2023). These properties are often defined through explicit reward functions that assess certain properties, such as the aesthetic quality of images. Such a requirement is crucial in fields where adherence to particular characteristics is necessary, such as alignment or drug discovery. The need to integrate explicit guidance without relying solely on datasets presents a unique challenge for training methodologies. Previous works have utilized methods such as reinforcement learning (RL) (Black et al., 2023; Fan et al., 2023) to tackle this problem. Nonetheless, these methods still suffer from issues like low sample efficiency.

In this work, we propose a novel approach, diffusion alignment with GFlowNets (DAG), that fine-tunes diffusion models to optimize black-box reward functions directly. Generative flow networks (Bengio et al., 2023, GFlowNets), initially introduced for efficient probabilistic inference with given densities in structured spaces, provide a unique framework for this task. Though initially proposed for composite graph-like structures, prior works have extended the GFlowNet framework to diffusion modeling (Zhang et al., 2022a; Lahlou et al., 2023). This work further investigates GFlowNet-inspired algorithms for the task of text-to-image

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2024 Workshop on Foundation Models in the Wild. Do not distribute.

diffusion alignment. By aligning the learning process to focus on generating samples with probability proportional to reward functions rather than maximizing them, our method allows the diffusion model to directly target and generate samples that are not only high in quality but also fulfill specific predefined criteria. Besides developing a denoising diffusion probabilistic model-specific GFlowNet algorithm, we also propose a new KL-based way to optimize our models. In summary, our contributions are as follows:

- We propose Diffusion Alignment with GFlowNet (DAG), a GFlowNet-based algorithm using the denoising structure of diffusion models, to improve large-scale text-to-image alignment with a black-box reward function.
- We propose a KL-based objective for optimizing GFlowNets that achieves comparable or better sample efficiency. We further called the resulting algorithm for the alignment problem DAG-KL.
- Our methods achieve better sample efficiency than the reinforcement learning baseline within the same number of trajectory rollouts across a number of different learning targets.

2. Methodology

2.1. Diffusion alignment with GFlowNets

We review the recipe about viewing the denoising process as a MDP in Section D.1. In this section, we describe our proposed algorithm, diffusion alignment with GFlowNets (DAG). Rather than directly optimizing the reward targets as in RL, we aim to train the generative models so that in the end they could generate objects with a probability *proportional* to the reward function: $p_{\boldsymbol{\theta}}(\mathbf{x}_0) \propto R(\mathbf{x}_0)$. To achieve this, we construct the following DB-based training objective based on Equation 14, by regressing its one side to another in the logarithm scale for any diffusion step transition $(\mathbf{x}_t, \mathbf{x}_{t-1})$.

$$\begin{aligned}\ell_{\text{DB}}(\mathbf{x}_t,\mathbf{x}_{t-1}) = (\log F_{\boldsymbol{\phi}}(\mathbf{x}_t, t) + \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, t) \quad (1)\\ - \log F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}, t-1) - \log q(\mathbf{x}_t|\mathbf{x}_{t-1}))^2 \quad (2)\end{aligned}$$

We additionally force $F_{\boldsymbol{\phi}}(\mathbf{x}_t, t = 0) = R(\mathbf{x}_0)$ to introduce the reward signal. Here $\boldsymbol{\theta}, \boldsymbol{\phi}$ are the parameters of the diffusion U-Net model and the GFlowNet state flow function (which is another neural network), respectively. One can prove that if the optimization is perfect, the resulting model will generate a distribution whose density value is proportional to the reward function $R(\cdot)$ (Bengio et al., 2023; Zhang et al., 2023a).

One way to parameterize the state flow function F is through the so-called forward-looking (Pan et al., 2023b, FL) technique in the way of $F_{\boldsymbol{\phi}}(\mathbf{x}_t, t) = \tilde{F}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)R(\mathbf{x}_t)$, where $\tilde{F}_{\boldsymbol{\phi}}$ is the actual neural network to be learned. Intuitively, this is equivalent to initializing the state flow function to be the reward function in a functional way; therefore, learning of the state flow would become an easier task. Note that to ensure $F_{\boldsymbol{\phi}}(\mathbf{x}_0, 0) = R(\mathbf{x}_0)$, we need to force $\tilde{F}_{\boldsymbol{\phi}}(\mathbf{x}_0, 0) = 1$ for all $\mathbf{x}_0$ at the terminal step.

Algorithm 1 Diffusion alignment with GFlowNets (DAG-DB & DAG-KL)

```
Require: Denoising policy $p_\theta(x_{t-1}|x_t, t)$, noising policy
    $q(x_t|x_{t-1})$, flow function $F_\phi(x_t, t)$, black-box reward
    function $R(\cdot)$
repeat
  Rollout $\tau = \{x_t\}_t$ with $p_\theta(x_{t-1}|x_t, t)$
  For each transition $(x_t, x_{t-1}) \in \tau$:
  if algorithm is DAG-DB then
    # normal DB-based update
    Update $\theta$ and $\phi$ with Equation 5
  else if algorithm is DAG-KL then
    # KL-based update
    Update $\phi$ with Equation 5
    Update $\theta$ with Equation 11
  end if
until some convergence condition =0
```

Incorporating denoising diffusion-specific structure However, the intermediate state $\mathbf{x}_t$ is noisy under our context, and thus not appropriate for being evaluated by the given reward function, which would give noisy result. What's more, what we are interested here is to "foresee" the reward of the terminal state $\mathbf{x}_0$ taken from the (partial) trajectory $\mathbf{x}_{t:0}$ starting from given $\mathbf{x}_t$. As a result, we can do the FL technique utilizing the particular structure of diffusion model as in $F_{\boldsymbol{\phi}}(\mathbf{x}_t, t) = \tilde{F}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)R(\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t))$, where $\hat{\mathbf{x}}_{\boldsymbol{\theta}}$ is the data prediction network. We notice that a similar technique has been used to improve classifier guidance (Bansal et al., 2023). In short, our innovation in FL technique is

$$F_{\boldsymbol{\phi}}(\mathbf{x}_t, t) = \tilde{F}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)R(\mathbf{x}_t) \implies \quad (3)$$

$$F_{\boldsymbol{\phi}}(\mathbf{x}_t, t) = \tilde{F}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)R(\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)). \quad (4)$$

Then the FL-DB training objective $\ell_{\text{FL}}(\mathbf{x}_t, \mathbf{x}_{t-1})$ becomes

$$\left(\log \frac{\tilde{F}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)R(\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t))p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\tilde{F}_{\boldsymbol{\phi}}(\mathbf{x}_{t-1}, t-1)R(\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}, t-1))q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right)^2. \quad (5)$$

Since in this work the reward function is a black-box, the gradient flow would not go through $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ when we take the gradient of $\boldsymbol{\theta}$. We summarize the algorithm in Algorithm 1 and refer to it as DAG-DB.

Remark 1 (GPU memory and the choice of GFlowNet objectives)**.** Similar to the temporal difference-λ in RL (Sutton, 1988), it is possible to use multiple connected transition steps rather than a single transition step to construct the learning objective. Other GFlowNet objectives such as Malkin et al. (2022); Madan et al. (2022) use partial trajectories with a series of transition steps to construct the training loss and provide a different trade-off between variance and bias in credit assignment. However, for large-scale setups, this is not easy to implement, as computing policy probabilities for multiple transitions would correspondingly increase the GPU memory and computation multiple times. For example, in the Stable Diffusion setting, we could only use a batch size of 8 on each GPU for single transition computation. If we want to use a two transition based training loss, we would need to decrease the batch size by half to 4. Similarly, we will have to shorten the trajectory length by a large margin if we want to use trajectory balance. This may influence the image generation quality and also make it tricky to compare with the RL baseline, which can be implemented with single transitions and does not need to decrease batch size or increase gradient accumulation. In practice, we find that single transition algorithms (such as our RL baseline) perform reasonably well.

2.2. A KL-based GFlowNet algorithm with REINFORCE gradient

GFlowNet detailed balance is an off-policy algorithm that uses training data from arbitrary distributions. In this section, we derive a different KL-based on-policy objective, which has been rarely investigated in GFlowNet literature. We can reformulate DB (Equation 1) from a square loss form to a KL divergence form

$$\min_{\boldsymbol{\theta}} \mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\|\frac{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)}\right). \tag{6}$$

In theory, when DB is perfectly satisfied, the right term $F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})/F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)$ is a normalized density; in practice, it could be an unnormalized one but does not affect the optimization. Next, define $b(\mathbf{x}_t,\mathbf{x}_{t-1})$ to be

$$\text{stop-gradient}\left(\log\frac{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right), \tag{7}$$

then the KL value of Equation 6 becomes $\mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}\left[b(\mathbf{x}_t,\mathbf{x}_{t-1})\right]$. We have the following result for deriving a practical REINFORCE-style objective.

Proposition 2. *The KL term in Equation 6 has the same expected gradient with $b(\mathbf{x}_t,\mathbf{x}_{t-1})\log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$:*

$$\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\|\frac{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)}\right) \tag{8}$$

$$= \mathbb{E}_{\mathbf{x}_{t-1}\sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_t)}\left[b(\mathbf{x}_t,\mathbf{x}_{t-1})\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right]. \tag{9}$$

We defer its proof to Section C.1 and make the following remarks:.

Remark 3 (gradient equivalence to detailed balance)**.** Recalling Equation 1, since we have $\nabla_{\boldsymbol{\theta}}\ell_{\mathrm{DB}}(\mathbf{x}_t,\mathbf{x}_{t-1}) = b(\mathbf{x}_t,\mathbf{x}_{t-1})\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$, it is clear that this KL-based objective would lead to the same expected gradient on $\boldsymbol{\theta}$ with Equation 1, if $\mathbf{x}_{t-1}\sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_t)$ (*i.e.*, samples being on-policy). Nonetheless, this on-policy property may not be true in practice since the current model is usually not the same as the model used for rollout trajectories after a few optimization steps.

Note that this REINFORCE style objective in Equation 8 is on-policy; the data has to come from the same distribution as the current model. In practice, the model would become not exactly on-policy after a few optimization steps, under which scenario we need to introduce the probability ratio $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)/p_{\boldsymbol{\theta}_{\mathrm{old}}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ via importance sampling:

$$\mathbb{E}_{\mathbf{x}_{t-1}\sim p_{\boldsymbol{\theta}_{\mathrm{old}}}(\cdot|\mathbf{x}_t)}\left[b(\mathbf{x}_t,\mathbf{x}_{t-1})\frac{\nabla_{\boldsymbol{\theta}}p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{p_{\boldsymbol{\theta}_{\mathrm{old}}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right]. \tag{10}$$

Therefore, we can define a new objective $\ell_{\mathrm{KL}}(\mathbf{x}_t,\mathbf{x}_{t-1})$

$$b(\mathbf{x}_t,\mathbf{x}_{t-1})\ \mathrm{clip}\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{p_{\boldsymbol{\theta}_{\mathrm{old}}}(\mathbf{x}_{t-1}|\mathbf{x}_t)},1-\epsilon,1+\epsilon\right), \tag{11}$$

where $\mathbf{x}_{t-1}\sim p_{\boldsymbol{\theta}_{\mathrm{old}}}(\cdot|\mathbf{x}_t)$. Here we also introduce a clip operation to remove too drastic update, following PPO (Schulman et al., 2017). We use this to update the policy parameter $\boldsymbol{\theta}$ and use FL-DB to only update $\boldsymbol{\phi}$. We call this "diffusion alignment with GFlowNet and REINFORCE gradient" method to be DAG-KL. Note that when calculating $b(\mathbf{x}_t,\mathbf{x}_{t-1})$, we also adopt the diffusion-specific FL technique developed in Section 2.1. We also put the algorithmic pipeline of DAG-KL in Algorithm 1.

3. Experiments

Experimental setups We choose Stable Diffusion v1.5 (Rombach et al., 2021) as our base generative model. For training, we use low-rank adaptation (Hu et al., 2021, LoRA) for parameter efficient computation. As for the reward functions, we do experiments with the LAION Aesthetics predictor, a neural aesthetic score trained from human feedback to give an input image an aesthetic rating. For text-image alignment rewards, we choose ImageReward (Xu et al., 2023) and human preference score (HPSv2) (Wu et al., 2023). They are both CLIP (Radford et al., 2021)-type models, taking a text-image pair as input and output a scalar score about to what extent the image follows the text description. We also test with the (in)compressibility reward, which computes the file size if the input image is stored in hardware storage. As for the prompt distribution, we use a set of 45 simple animal prompts from Black et al. (2023) for the Aesthetics task; we use the whole imagenet classes

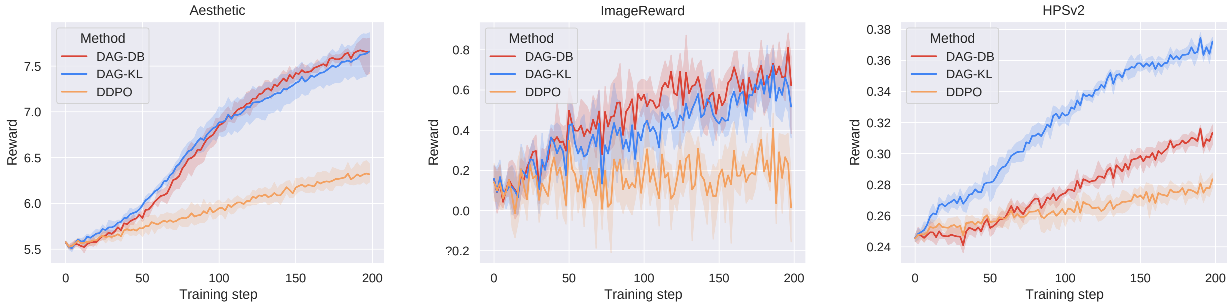


Figure 2: Sample efficiency results of our proposed methods and our RL baseline (DDPO). The experiments are conducted on reward functions including aesthetic score, ImageReward, and HPSv2.

for the (in)compressibility task; we use the DrawBench (Saharia et al., 2022) prompt set for the ImageReward task; we use the photo and painting prompts from the human preference dataset (HPDv2) (Wu et al., 2023) for the HPSv2 task. We notice that in our experiments, we use prompt set containing hundreds of prompts which is more than some previous work such as Black et al. (2023).

Effectiveness of the proposed methods We first demonstrate that our proposed methods could generated images that have meaningful improvements corresponding to the rewards being used. In Figure 1, we compare the images from the original Stable Diffusion pretrained model and our proposed method. After our post-training, the generated images become more vibrant and vivid; we also notice that these images have slightly higher saturation, which we believe is aligned with the human preference on good-looking pictures. We also visualize the experiment results on compressibility and incompressibility tasks in Figure 3. The second row shows the generated images from the model trained with the compressibility reward, which have low details and smooth textures, and also have very limited colors. On the other hand, the model trained with incompressibility reward would generate images with high frequency texture, as shown in the third row. These results indicate that our method could effectively incorporate the reward characteristics into the generative models. We defer more experimental details to Section D.2.

Algorithmic comparisons The main baseline we compare with is denoising diffusion policy optimization (Black et al., 2023, DDPO), an RL algorithm that is specifically designed for denoising diffusion alignment and has been shown to outperform other align-from-black-box-reward methods including (Lee et al., 2023; Fan et al., 2023). We show the reward curves w.r.t. the training steps of the aesthetic, ImageReward, and HPSv2 rewards in Figure 2. Here, the number of training steps corresponds proportionally to the number of trajectories collected. Both our proposed methods, DAG-DB and DAG-KL, achieve faster credit assignment than the DDPO baseline by a large margin. We also put corresponding curve plots for compressibility and incompressibility rewards in Figure 6, which also demonstrates the advantage of our methods. We defer related training details to Section D.2.

Apart from quantitative comparisons, we also visualize the alignment improvement for models trained in the HPSv2 task. In Figure 4 and Figure 8 in Appendix, we exhibit generation results for different prompts across the original Stable Diffusion, DDPO, DAG-DB, and DAG-KL models. For example, in the first “a counter top with food sitting on some towels” example, images from the original Stable Diffusion either do not have food or the food is not on towels, which is also the case for DDPO generation. This is improved for both DAG-DB and DAG-KL generation in that they capture the location relationship correctly. In the “personal computer desk room with large glass double doors” example, both the original and DDPO models cannot generate any double doors in the image, and DAG-DB model sometimes also fails. In contrast, the DAG-KL model seems to understand the concept well. Generation with other prompts also has similar results.

In Figure 5, we visualize the gradual alignment improvement of our DAG-KL method with regard to the training progress for the HPSv2 task. We show the images of our methods at $0\%, 25\%, 50\%, 75\%$, and 100% training progress. In the example of “a helmet-wearing monkey skating”, the DDPO baseline could generate a skating monkey but seems to fail to generate a helmet. For the proposed method, the model gradually learns to handle the concept of a helmet over the course of training. In the “anthropomorphic Virginia opossum playing guitar” example, the baseline understands the concept of guitar well, but the generated images are not anthropomorphic, while our method manages to generate anthropomorphic opossums decently.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pp. 1–3, 2024.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

Bansal, A., Chu, H.-M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., and Goldstein, T. Universal guidance for diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 843–852, 2023.

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. Flow network based generative models for non-iterative diverse candidate generation. *Neural Information Processing Systems (NeurIPS)*, 2021.

Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., and Bengio, E. GFlowNet foundations. *Journal of Machine Learning Research*, (24):1–76, 2023.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. Training diffusion models with reinforcement learning. *ArXiv*, abs/2305.13301, 2023.

Chen, C., Wang, A., Wu, H., Liao, L., Sun, W., Yan, Q., and Lin, W. Enhancing diffusion models with text-encoder reinforcement learning. *ArXiv*, abs/2311.15657, 2023. URL https://api.semanticscholar.org/CorpusID:265457291.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards, 2023.

Deleu, T., Góis, A., Emezue, C., Rankawat, M., Lacoste-Julien, S., Bauer, S., and Bengio, Y. Bayesian structure learning with generative flow networks. *Uncertainty in Artificial Intelligence (UAI)*, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis, 2021.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. Raft: Reward ranked finetuning for generative foundation model alignment, 2023.

Fan, Y., Watkins, O., Du, Y., Liu, H., Ryu, M., Boutilier, C., Abbeel, P., Ghavamzadeh, M., Lee, K., and Lee, K. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *ArXiv*, abs/2305.16381, 2023.

Fox, R., Pakman, A., and Tishby, N. Taming the noise in reinforcement learning via soft updates, 2017.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. *International Conference on Machine Learning (ICML)*, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. URL https://api.semanticscholar.org/CorpusID:219955663.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in atari, 2018.

Jain, M., Bengio, E., Hernandez-Garcia, A., Rector-Brooks, J., Dossou, B. F., Ekbote, C., Fu, J., Zhang, T., Kilgour, M., Zhang, D., Simine, L., Das, P., and Bengio, Y. Biological sequence design with GFlowNets. *International Conference on Machine Learning (ICML)*, 2022.

Jain, M., Deleu, T., Hartford, J. S., Liu, C.-H., Hernández-García, A., and Bengio, Y. Gflownets for ai-driven scientific discovery. *ArXiv*, abs/2302.00615, 2023a. URL https://api.semanticscholar.org/CorpusID:256459319.

Jain, M., Raparthy, S. C., Hernandez-Garcia, A., Rector-Brooks, J., Bengio, Y., Miret, S., and Bengio, E. Multi-objective GFlowNets. *International Conference on Machine Learning (ICML)*, 2023b.

Janner, M., Du, Y., Tenenbaum, J. B., and Levine, S. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Kim, K., Jeong, J., An, M., Ghavamzadeh, M., Dvijotham, K., Shin, J., and Lee, K. Confidence-aware reward optimization for fine-tuning text-to-image models, 2024a.

Kim, M., Yun, T., Bengio, E., Zhang, D., Bengio, Y., Ahn, S., and Park, J. Local search gflownets. *ArXiv*, abs/2310.02710, 2023.

Kim, M., Ko, J., Yun, T., Zhang, D., Pan, L., Kim, W., Park, J., Bengio, E., and Bengio, Y. Learning to scale logits for temperature-conditional gflownets, 2024b.

Lahlou, S., Deleu, T., Lemos, P., Zhang, D., Volokhova, A., Hernández-García, A., Ezzine, L. N., Bengio, Y., and Malkin, N. A theory of continuous generative flow networks. *International Conference on Machine Learning (ICML)*, 2023.

Lee, K., Liu, H., Ryu, M., Watkins, O., Du, Y., Boutilier, C., Abbeel, P., Ghavamzadeh, M., and Gu, S. S. Aligning text-to-image models using human feedback. *ArXiv*, abs/2302.12192, 2023.

Liu, D., Jain, M., Dossou, B. F. P., Shen, Q., Lahlou, S., Goyal, A., Malkin, N., Emezue, C. C., Zhang, D., Hassen, N., Ji, X., Kawaguchi, K., and Bengio, Y. Gflowout: Dropout with generative flow networks. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:253097963.

Ma, J., Bengio, E., Bengio, Y., and Zhang, D. Baking symmetry into gflownets.

Madan, K., Rector-Brooks, J., Korablyov, M., Bengio, E., Jain, M., Nica, A., Bosc, T., Bengio, Y., and Malkin, N. Learning GFlowNets from partial episodes for improved convergence and stability. *International Conference on Machine Learning (ICML)*, 2022.

Malkin, N., Jain, M., Bengio, E., Sun, C., and Bengio, Y. Trajectory balance: Improved credit assignment in GFlowNets. *Neural Information Processing Systems (NeurIPS)*, 2022.

Malkin, N., Lahlou, S., Deleu, T., Ji, X., Hu, E., Everett, K., Zhang, D., and Bengio, Y. GFlowNets and variational inference. *International Conference on Learning Representations (ICLR)*, 2023.

Marion, P., Korba, A., Bartlett, P., Blondel, M., Bortoli, V. D., Doucet, A., Llinares-López, F., Paquette, C., and Berthet, Q. Implicit diffusion: Efficient optimization through stochastic sampling, 2024.

Pan, L., Jain, M., Madan, K., and Bengio, Y. Pre-training and fine-tuning generative flow networks, 2023a.

Pan, L., Malkin, N., Zhang, D., and Bengio, Y. Better training of GFlowNets with local credit and incomplete trajectories. *International Conference on Machine Learning (ICML)*, 2023b.

Pan, L., Zhang, D., Courville, A., Huang, L., and Bengio, Y. Generative augmented flow networks. *International Conference on Learning Representations (ICLR)*, 2023c.

Pan, L., Zhang, D., Jain, M., Huang, L., and Bengio, Y. Stochastic generative flow networks. *Uncertainty in Artificial Intelligence (UAI)*, 2023d.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Prabhudesai, M., Goyal, A., Pathak, D., and Fragkiadaki, K. Aligning text-to-image diffusion models with reward backpropagation, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015. URL https://api.semanticscholar.org/CorpusID:3719281.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models, 2022.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shen, M. W., Bengio, E., Hajiramezanali, E., Loukas, A., Cho, K., and Biancalani, T. Towards understanding and improving gflownet training. *ArXiv*, abs/2305.07170, 2023. URL https://api.semanticscholar.org/CorpusID:258676487.

Sohl-Dickstein, J. N., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015. URL https://api.semanticscholar.org/CorpusID:14888175.

Song, Y., Sohl-Dickstein, J. N., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020. URL https://api.semanticscholar.org/CorpusID:227209335.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Tiapkin, D., Morozov, N., Naumov, A., and Vetrov, D. P. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pp. 4213–4221. PMLR, 2024.

Uehara, M., Zhao, Y., Black, K., Hajiramezanali, E., Scalia, G., Diamant, N. L., Tseng, A. M., Biancalani, T., and Levine, S. Fine-tuning of continuous-time diffusion models as entropy-regularized control, 2024.

Venkatraman*, S., Jain*, M., Scimeca*, L., Kim*, M., Sendera*, M., Hasan, M., Rowe, L., Mittal, S., Lemos, P., Bengio, E., Adam, A., Rector-Brooks, J., Bengio, Y., Berseth, G., and Malkin, N. Amortizing intractable inference in diffusion models for vision, language, and control. 2024.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. URL https://api.semanticscholar.org/CorpusID:5560643.

Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion model alignment using direct preference optimization, 2023a.

Wallace, B., Gokul, A., Ermon, S., and Naik, N. V. End-to-end diffusion latent optimization improves classifier guidance. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7246–7256, 2023b.

Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., and Li, H. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *ArXiv*, abs/2306.09341, 2023.

Wu, X., Hao, Y., Zhang, M., Sun, K., Huang, Z., Song, G., Liu, Y., and Li, H. Deep reward supervisions for tuning text-to-image diffusion models, 2024.

Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation. *ArXiv*, abs/2304.05977, 2023.

Yang, K., Tao, J., Lyu, J., Ge, C., Chen, J., Li, Q., Shen, W., Zhu, X., and Li, X. Using human feedback to fine-tune diffusion models without any reward model, 2024.

Yang, M., Cho, K., Merchant, A., Abbeel, P., Schuurmans, D., Mordatch, I., and Cubuk, E. D. Scalable diffusion for materials generation. *arXiv preprint arXiv:2311.09235*, 2023.

Zhang, D., Chen, R. T. Q., Malkin, N., and Bengio, Y. Unifying generative models with GFlowNets and beyond. *arXiv preprint arXiv:2209.02606v2*, 2022a.

Zhang, D., Malkin, N., Liu, Z., Volokhova, A., Courville, A., and Bengio, Y. Generative flow networks for discrete probabilistic modeling. *International Conference on Machine Learning (ICML)*, 2022b.

Zhang, D., Chen, R. T. Q., Liu, C.-H., Courville, A. C., and Bengio, Y. Diffusion generative flow samplers: Improving learning signals through partial trajectory optimization. *ArXiv*, abs/2310.02679, 2023a.

Zhang, D., Courville, A., Bengio, Y., Zheng, Q., Zhang, A., and Chen, R. T. Q. Latent state marginalization as a low-cost approach for improving exploration, 2023b.

Zhang, D., Dai, H., Malkin, N., Courville, A. C., Bengio, Y., and Pan, L. Let the flows tell: Solving graph combinatorial optimization problems with gflownets. *ArXiv*, abs/2305.17010, 2023c. URL https://api.semanticscholar.org/CorpusID:258947700.

Zhang, D., Pan, L., Chen, R. T. Q., Courville, A. C., and Bengio, Y. Distributional gflownets with quantile flows. *arXiv preprint arXiv:2302.05793*, 2023d.

Zhang, D. W., Rainone, C., Peschl, M. F., and Bondesan, R. Robust scheduling with gflownets. *ArXiv*, abs/2302.05446, 2023e. URL https://api.semanticscholar.org/CorpusID:256827133.

Zhou, M., Yan, Z., Layne, E., Malkin, N., Zhang, D., Jain, M., Blanchette, M., and Bengio, Y. Phylogfn: Phylogenetic inference with generative flow networks, 2024.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

Zimmermann, H., Lindsten, F., van de Meent, J.-W., and Naesseth, C. A. A variational perspective on generative flow networks. *ArXiv*, abs/2210.07992, 2022. URL https://api.semanticscholar.org/CorpusID:252907672.

A. Preliminaries

A.1. Diffusion models

Denoising diffusion model (Vincent, 2011; Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020) is a class of hierarchical latent variable models. The latent variables are initialized from a white noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then go through a sequential denoising (reverse) process $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Therefore, the resulting generated distribution takes the form of

$$p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})\,\mathrm{d}\mathbf{x}_{1:T} = \int p(\mathbf{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\,\mathrm{d}\mathbf{x}_{1:T}. \tag{12}$$

On the other hand, the variational posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, also called a diffusion or forward process, can be factorized as a Markov chain $\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$ composed by a series of conditional Gaussian distributions $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \alpha_t/\alpha_{t-1}\mathbf{x}_{t-1}, (1-\alpha_t^2/\alpha_{t-1}^2)\mathbf{I})$, where $\{\alpha_t, \sigma_t\}_t$ is a set of pre-defined signal-noise schedule. Specifically, in Ho et al. (2020) we have $\alpha_t^2 + \sigma_t^2 = 1$. The benefit of such a noising process is that its marginal has a simple close form: $q(\mathbf{x}_t|\mathbf{x}_0) = \int q(\mathbf{x}_{1:t}|\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_{1:t-1} = \mathcal{N}(\mathbf{x}_t; \alpha_t\mathbf{x}_0, \sigma_t^2\mathbf{I})$.

Given a data distribution $p_{\text{data}}(\cdot)$, the variational lower bound of model log likelihood can be written in the following simple denoising objective:

$$\mathcal{L}_{\text{denoising}}(\boldsymbol{\theta}) = \mathbb{E}_{t,\mathbf{x}_0\sim p_{\text{data}},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\mathbf{x}_0 - \hat{\mathbf{x}}_{\boldsymbol{\theta}}(\alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}, t)\|^2\right], \tag{13}$$

where $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ is a deep neural network to predict the original clean data $\mathbf{x}_0$ given the noisy input $\mathbf{x}_t = \alpha_t\mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}$, which can be used to parameterize the denoising process $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \left(\sigma_{t-1}^2\alpha_t\mathbf{x}_t + (\alpha_{t-1}^2 - \alpha_t^2)\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right)/\sigma_t^2\alpha_{t-1}, (1-\alpha_t^2/\alpha_{t-1}^2)\mathbf{I})$. In practice, the network can also be parameterized with noise prediction or v-prediction (Salimans & Ho, 2022). The network architecture usually has a U-Net (Ronneberger et al., 2015) structure.

In multimodal applications such as text-to-image tasks, the denoising diffusion model would have a conditioning $\mathbf{c}$ in the sense of $p_{\boldsymbol{\theta}}(\mathbf{x}_0; \mathbf{c}) = \int p(\mathbf{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t; \mathbf{c})\,\mathrm{d}\mathbf{x}_{1:T}$. The data prediction network, $\hat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \mathbf{c})$ in this case, will also take $\mathbf{c}$ as a conditioning input. We ignore the notation of $\mathbf{c}$ without loss of generality.

A.2. GFlowNets

Generative flow network (Bengio et al., 2021, GFlowNet) is a high-level algorithmic framework of amortized inference, also known as training generative models with a given unnormalized target density function. Let $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ be a directed acyclic graph, where $\mathcal{S}$ is the set of states and $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ are the set of actions. We assume the environmental transition is deterministic, *i.e.*, one action would only lead to one next state. There is a unique *initial state* $\mathbf{s}_0 \in \mathcal{S}$ which has no incoming edges and a set of *terminal states* $\mathbf{s}_N$ without outgoing edges. A GFlowNet has a stochastic *forward policy* $P_F(\mathbf{s}'|\mathbf{s})$ for transition $(\mathbf{s} \to \mathbf{s}')$ as a conditional distribution over the children of a given state $\mathbf{s}$, which can be used to induce a distribution over trajectories via $P(\tau) = \prod_{n=0}^{N-1} P_F(\mathbf{s}_{n+1}|\mathbf{s}_n)$, where $\tau = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_N)$. On the other hand, the *backward policy* $P_B(\mathbf{s}|\mathbf{s}')$ is a distribution over the parents of a given state $\mathbf{s}'$. The *terminating distribution* defined by $P_T(\mathbf{x}) = \sum_{\tau\to\mathbf{x}} P_F(\tau)$ is the ultimate terminal state distribution generated by the GFlowNet. The goal of training GFlowNet is to obtain a forward policy such that $P_T(\cdot) \propto R(\cdot)$, where $R(\cdot)$ is a black-box *reward function* or unnormalized density that takes only non-negative values. Notice that we do not know the normalizing factor $Z = \sum_{\mathbf{x}} R(\mathbf{x})$. We can use the *trajectory flow* function $F(\tau) = ZP_F(\tau)$ to take in the effect of the normalizing factor, and the corresponding *state flow* function $F(\mathbf{s}) = \sum_{\tau\ni\mathbf{s}} F(\tau)$ to model the unnormalized probability flow of intermediate state $\mathbf{s}$.

Detailed balance (DB) The GFlowNet detailed balance condition provides a way to learn the above mentioned GFlowNet modules. For any single transition $(\mathbf{s} \to \mathbf{s}')$, the following DB criterion holds:

$$F(\mathbf{s})P_F(\mathbf{s}'|\mathbf{s}) = F(\mathbf{s}')P_B(\mathbf{s}|\mathbf{s}'), \quad \forall(\mathbf{s} \to \mathbf{s}') \in \mathcal{A}. \tag{14}$$

Furthermore, for any terminating state $\mathbf{x}$, we require $F(\mathbf{x}) = R(\mathbf{x})$. In practice, these constraints can be transformed into tractable training objectives, as will be shown in Section 2. Based on GFlowNet theories in Bengio et al. (2023), if the DB criterion is satisfied for any transition, then the terminating distribution $P_T(\cdot)$ will be the same desired target distribution whose density is proportional to $R(\cdot)$.

B. Related Works

Diffusion alignment People have been modeling human values to a reward function in areas such as game (Ibarz et al., 2018) and language modeling (Bai et al., 2022) to make the model more aligned. In diffusion models, early researchers used various kinds of guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) to achieve the goal of steerable generation under the reward. This approach is as simple as plug-and-play but requires querying the reward function during inference time. Another way is to post-train the model to incorporate the information from the reward function, which has a different setup from guidance methods; there is also work showing that this outperforms guidance methods (Uehara et al., 2024). Lee et al. (2023); Dong et al. (2023) achieve this through maximum likelihood estimation on model-generated samples, which are reweighted by the reward function. These works could be thought of as doing RL in one-step MDPs. Black et al. (2023); Fan et al. (2023) design RL algorithm by taking the diffusion generation process as a MDP (Section D.1). In this work, we focus on black-box rewards where it is appropriate to use RL or GFlowNet methods. Furthermore, there are methods developed specifically for differentiable rewards setting (Clark et al., 2023; Wallace et al., 2023b; Prabhudesai et al., 2023; Wu et al., 2024; Xu et al., 2023; Uehara et al., 2024; Marion et al., 2024). Besides, Chen et al. (2023) study the effect of finetuning text encoder rather than diffusion U-Net. There is also work that relies on preference data rather than an explicit reward function (Wallace et al., 2023a; Yang et al., 2024). Kim et al. (2024a) investigate how to obtain a robust reward based on multiple different reward functions.

GFlowNets GFlowNet is a family of generalized variational inference algorithms that treats the data sampling process as a sequential decision-making one. It is useful for generating diverse and high-quality samples in structured scientific domains (Jain et al., 2022; 2023b; Liu et al., 2022; Jain et al., 2023a; Shen et al., 2023; Zhang et al., 2023d; Pan et al., 2023a; Kim et al., 2023; 2024b). A series of works have studied the connection between GFlowNets and probabilistic modeling methods (Zhang et al., 2022b; Zimmermann et al., 2022; Malkin et al., 2023; Zhang et al., 2022a; Ma et al.), and between GFlowNets and control methods (Pan et al., 2023c;d;b; Tiapkin et al., 2024). GFlowNets also have wide application in causal discovery (Deleu et al., 2022), phylogenetic inference (Zhou et al., 2024), and combinatorial optimization (Zhang et al., 2023e;c). A concurrent work (Venkatraman* et al., 2024) also studies GFlowNet on diffusion alignment which is similar to this work but has different scope and different developed algorithm. Specifically, this work is aiming for posterior approximate inference that the reward function is treated as likelihood information, and develops a trajectory balance (Malkin et al., 2022) based algorithm on length modified trajectories.

C. Proof

C.1. Proof of Proposition 2

Proof. Recalling that $b(\mathbf{x}_t, \mathbf{x}_{t-1}) = \text{stop-gradient}\left(\log \frac{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right)$,

$$
\begin{aligned}
&\nabla_{\boldsymbol{\theta}}\mathcal{D}_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\|\frac{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)}\right)\\
=&\nabla_{\boldsymbol{\theta}}\int p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\log\frac{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}\,\mathrm{d}\mathbf{x}_{t-1}\\
=&\int\nabla_{\boldsymbol{\theta}}p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\log\frac{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}\,\mathrm{d}\mathbf{x}_{t-1}+\int p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\nabla_{\boldsymbol{\theta}}\log\frac{F_{\boldsymbol{\phi}}(\mathbf{x}_t,t)p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{F_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},t-1)q(\mathbf{x}_t|\mathbf{x}_{t-1})}\,\mathrm{d}\mathbf{x}_{t-1}\\
=&\int p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)b(\mathbf{x}_t,\mathbf{x}_{t-1})\,\mathrm{d}\mathbf{x}_{t-1}+\underbrace{\int p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\,\mathrm{d}\mathbf{x}_{t-1}}_{=\nabla_{\boldsymbol{\theta}}\int p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\,\mathrm{d}\mathbf{x}_{t-1}=\nabla_{\boldsymbol{\theta}}1=0}\\
=&\mathbb{E}_{\mathbf{x}_{t-1}\sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_t)}\left[b(\mathbf{x}_t,\mathbf{x}_{t-1})\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right].
\end{aligned}
$$

□

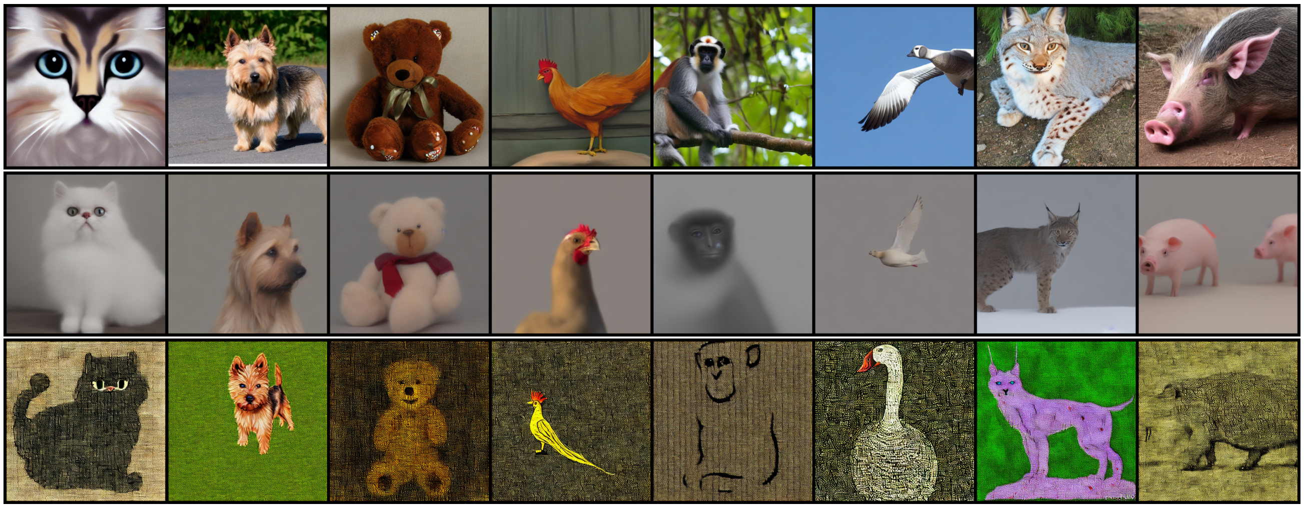


Figure 3: *Top*: samples from the original Stable Diffusion model. *Middle*: the proposed method trained with compressibility reward; these images have very smooth texture. *Down*: the proposed method trained with incompressibility reward; the texture part of images contains high frequency noise.

D. More about methodology

D.1. Denoising Markov decision process

The denoising process for text-to-image diffusion models can be easily reformulated as a multi-step Markov decision process (MDP) with finite horizon (Fan et al., 2023; Black et al., 2023) as follows:

$$\mathbf{s}_t = (\mathbf{x}_{T-t}, \mathbf{c}), \quad p(\mathbf{s}_0) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}) \otimes p(\mathbf{c}), \quad \pi_{\boldsymbol{\theta}}(\mathbf{a}_t|\mathbf{s}_t) = p_{\boldsymbol{\theta}}(\mathbf{x}_{T-t-1}|\mathbf{x}_{T-t}, \mathbf{c}), \tag{15}$$

$$\mathbf{a}_t = \mathbf{x}_{T-t-1}, \quad r(\mathbf{s}_t, \mathbf{a}_t) = R(\mathbf{s}_{t+1}, \mathbf{c}) \text{ only if } t = T-1, \quad p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) = \delta_{\mathbf{a}_t} \otimes \delta_{\mathbf{c}}. \tag{16}$$

Here $\mathbf{s}_t, \mathbf{a}_t$ is the state and action at time step t under the context of MDP. The state space is defined to be the product space (denoted by $\otimes$) of $\mathbf{x}$ in reverse time ordering and conditional prompt $\mathbf{c}$. The RL policy π is just the denoising conditional distribution. In this MDP, when time t has not reached the terminal step, we define the reward $r(\mathbf{s}_t, \mathbf{a}_t)$ to be 0. δ here denotes the Dirac distribution.

Remark 4 (RL optimal solutions)**.** Training a standard RL algorithm within this diffusion MDP to perfection means the model would only generate a single trajectory with the largest reward value. This usually comes with the disadvantage of mode collapse in generated samples in practice. One direct solution is soft / maximum entropy RL (Ziebart et al., 2008; Fox et al., 2017; Haarnoja et al., 2017; Zhang et al., 2023b), whose optimal solution is a trajectory-level distribution and the probability of generating each trajectory is proportional to its trajectory cumulative reward, *i.e.*, $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) \propto \sum_t R_t(\mathbf{x}_t) = R(\mathbf{x}_0)$. However, in theory this means $p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})\,\mathrm{d}\mathbf{x}_{1:T} \propto \int R(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_{1:T} = R(\mathbf{x}_0) \cdot \int \mathrm{d}1\mathbf{x}_{1:T}$, which is not a well-defined finite term for unbounded continuous spaces. In contrast, the optimal solution of GFlowNet is $p_{\boldsymbol{\theta}}(\mathbf{x}_0) \propto R(\mathbf{x}_0)$.

Remark 5 (diffusion model as GFlowNet)**.** This formulation has a direct connection to the GFlowNet MDP definition in Section A.2, which has been pointed out by Zhang et al. (2022a) and developed in Lahlou et al. (2023); Zhang et al. (2023a); Venkatraman* et al. (2024). To be specific, the action transition $(\mathbf{s}_t, \mathbf{a}_t) \to \mathbf{s}_{t+1}$ is a Dirac distribution and can be directly linked with the $(\mathbf{s}_t \to \mathbf{s}_{t+1})$ edge transition in the GFlowNet language. More importantly, the conditional distribution of the denoising process $p_{\boldsymbol{\theta}}(\mathbf{x}_{T-t-1}|\mathbf{x}_{T-t})$ corresponds to the GFlowNet forward policy $P_F(\mathbf{s}_{t+1}|\mathbf{s}_t)$, while the conditional distribution of the diffusion process $q(\mathbf{x}_{T-t}|\mathbf{x}_{T-t-1})$ corresponds to the GFlowNet backward policy $P_B(\mathbf{s}_t|\mathbf{s}_{t+1})$. Besides, $\mathbf{x}_t$ is a GFlowNet terminal state if and only if $t = 0$.

The above discussion could be summarized in the right table. In the following text, we use the denoising diffusion notation instead of GFlowNet notation as it is familiar to more broad audience. What's more, we ignore conditioning $\mathbf{c}$ for the sake of simplicity.

Denoising diffusion	GFlowNet
$(\mathbf{x}_{T-t}, \mathbf{c})$	$\mathbf{s}_t$
$p(\mathbf{x}_{T-t-1}\|\mathbf{x}_{T-t}, \mathbf{c})$	$P_F(\mathbf{s}_{t+1}\|\mathbf{s}_t)$
$q(\mathbf{x}_{T-t}\|\mathbf{x}_{T-t-1})$	$P_B(\mathbf{s}_t\|\mathbf{s}_{t+1})$

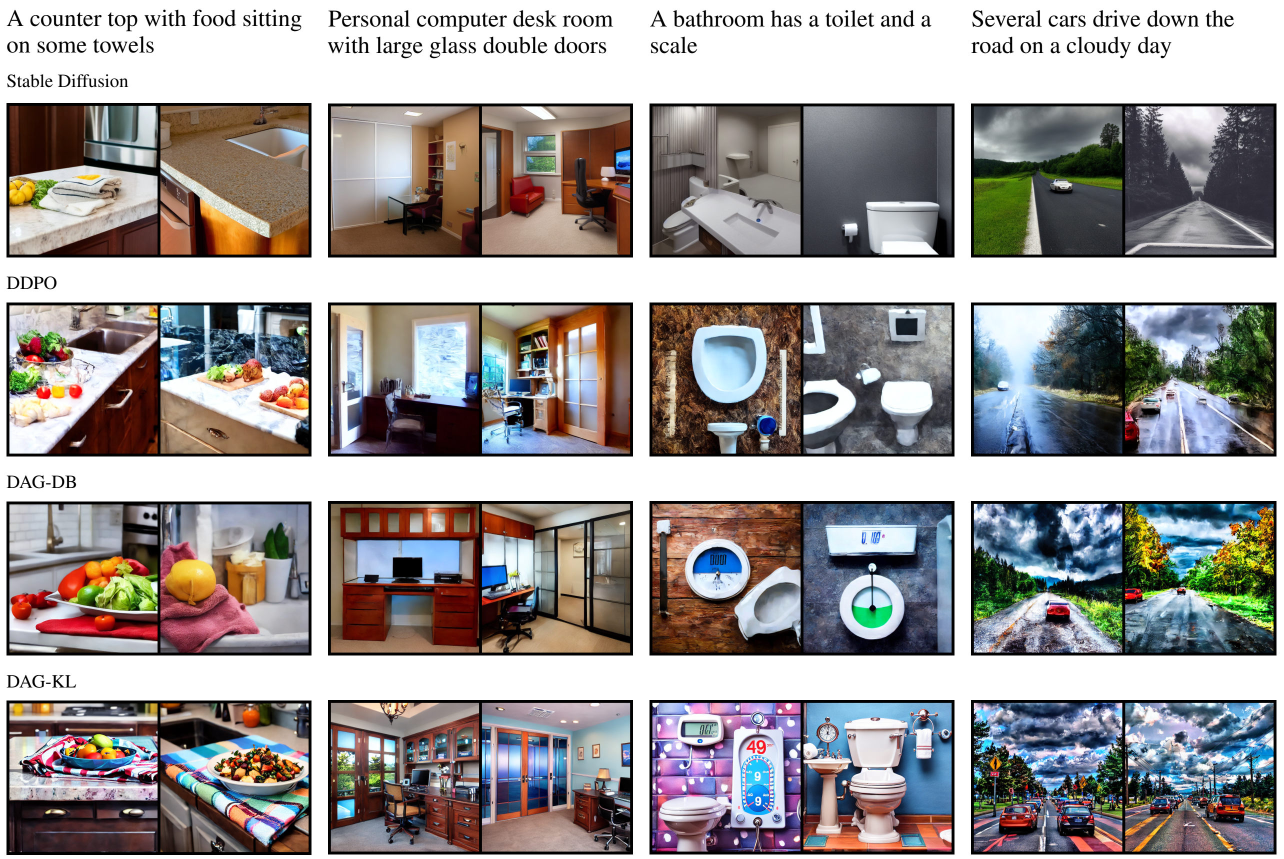

A counter top with food sitting on some towels
Personal computer desk room with large glass double doors
A bathroom has a toilet and a scale
Several cars drive down the road on a cloudy day
Stable Diffusion
DDPO
DAG-DB
DAG-KL

Figure 4: Text-image alignment results. We display four prompts and the corresponding generation visualization from the original Stable Diffusion (1st row), DDPO (2nd row), DAG-DB (3rd row), and DAG-KL (4th row) models to compare their alignment abilities. See Figure 8 for more results.

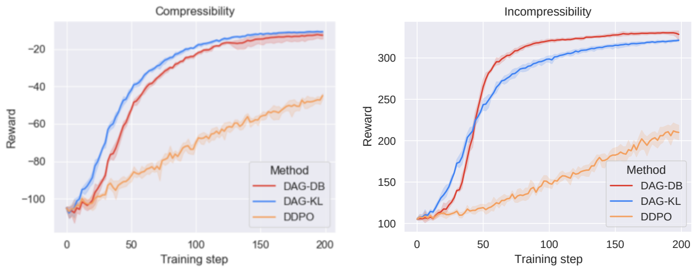
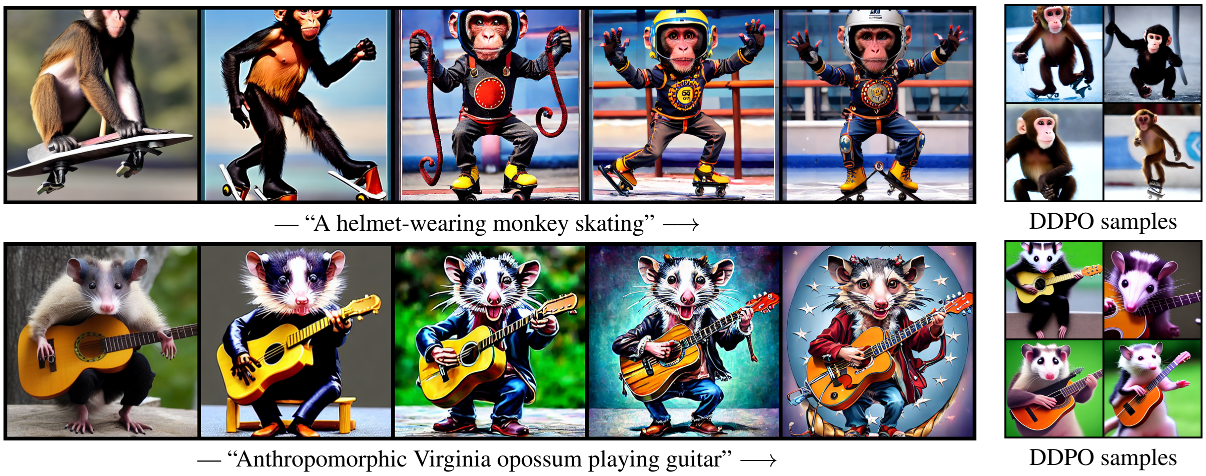


Figure 6: Sample efficiency results of our proposed methods and our RL baseline (DDPO) on learning from compressibility and incompressibility rewards.

D.2. Experimental details

Regarding training hyperparameters, we follow the DDPO github repository implementation and describe them below for completeness. We use classifier-free guidance (Ho & Salimans, 2022, CFG) with guidance weight being 5. We use a 50-step DDIM schedule. We use NVIDIA 8×A100 80GB GPUs for each task, and use a batch size of 8 per single GPU. We do 4 step gradient accumulation, which makes the essential batch size to be 256. For each "epoch", we sample 512 trajectories during the rollout phase and perform 8 optimization steps during the training phase. We train for 100 epochs. We use a 3×10^{-4} learning rate for both the diffusion model and the flow function model without further tuning. We use the AdamW optimizer and gradient clip with the norm being 1. We set $\epsilon = 1 \times 10^{-4}$ in Equation 11. We use bfloat16 precision.

The GFlowNet framework requires the reward function to be always non-negative, so we just take the exponential of the reward to be used as the GFlowNet reward. We also set the reward exponential to $\beta = 100$ (*i.e.*, setting the distribution temperature to be $1/100$). Therefore, $\log R(\cdot) = \beta R_{\text{original}}(\cdot)$. Note that in GFlowNet training practice, we only need to use the logarithm of the reward rather than the original reward value. We linearly anneal β from 0 to its maximal value in the first half of the training. We found that this almost does not change the final result but is helpful for training stability.For DAG-KL, we put the final β coefficient on the KL gradient term. We also find using a KL regularization $\mathcal{D}_{\text{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) \| p_{\boldsymbol{\theta}_{\text{old}}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)$ to be helpful for stability (this is also mentioned in Fan et al. (2023)). In practice, it is



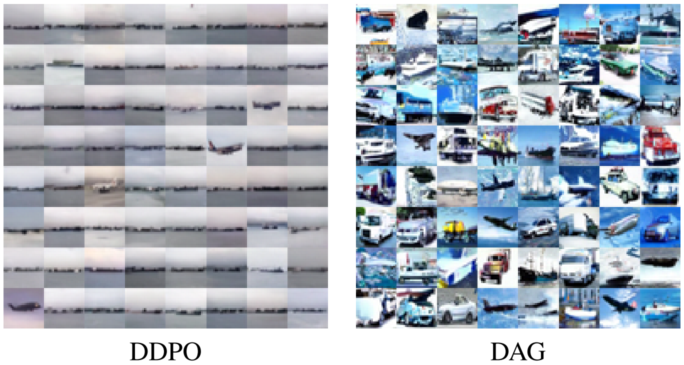
— “A helmet-wearing monkey skating” ⟶

DDPO samples

— “Anthropomorphic Virginia opossum playing guitar” ⟶

DDPO samples

Figure 5: Visualization of alignment with regard to training progress. *Left*: the generated images from the proposed method become more aligned to the text prompt over the course of training. *Right*: samples from the DDPO baseline.



DDPO

DAG

Figure 7: Samples on CIFAR-10 diffusion alignment experiments. The reward function is the probability of the generated image falling into the categories of car, truck, ship, and plane calculated by a pretrained classifier. The RL baseline shows mode collapse behaviors while the target distribution is actually multimodal.

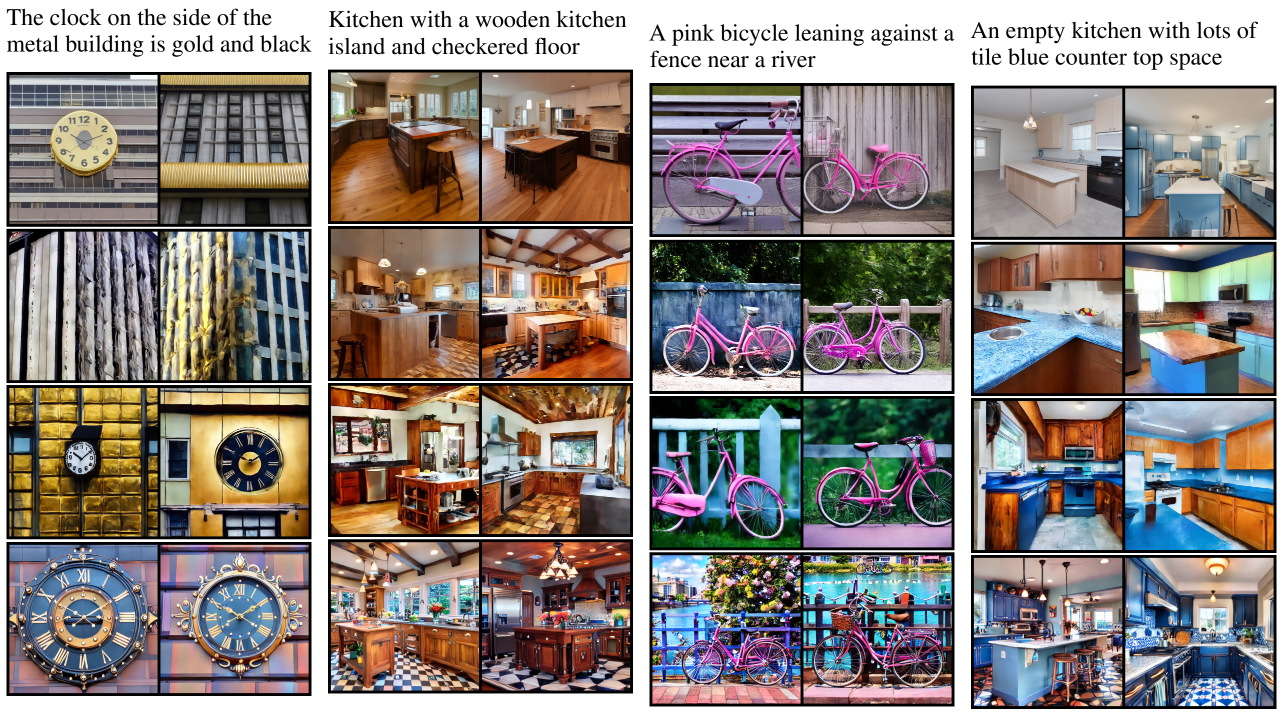
essentially adding a ℓ_2 regularization term on the output of the U-Net after CFG between the current model and previous rollout model. We simply use a coefficient 1 on this term without further tuning.

We use Stable Diffusion v1.5 as base model and use LoRA for post-training following Black et al. (2023). For the architecture of the state flow function, we take a similar structure to the downsample part of the U-Net. The implementation is based on the hugging face diffusers package. We use 3 “CrossAttnDownBlock2D” blocks and 1 “DownBlock2D” and do downsampling on all of them. We set the layers per block to be 1, and set their block out channels to be $64, 128, 256, 256$. We use a final average pooling layer with kernel and stride size 4 to output a scalar given inputs including latent image, time step, and prompt embedding. We do not report diversity metric as in previous GFlowNet literature, as the average pairwise Euclidean distance in high dimensional space ($64 \times 64 \times 4 > 10,000$ dim.) is not a meaningful metric.

D.3. CIFAR-10 toy example

We also include a toy experiment on a CIFAR-10 pretrained DDPM[1]. We train a ResNet18 classifier and set the reward function to be the probability of the generated image falling into the categories of car, truck, ship, and plane. We use same hyperparameters with the Stable Diffusion setting, except we only use 1 GPU with 256 batch size for each run without gradient accumulation. We illustrate the generation results in Figure 7. We use DAG-DB here, and the DAG-KL generation is similar and non-distinguishable with it. We can see that in this relative toy task, the RL baseline easily optimizes the

[1] `https://huggingface.co/google/ddpm-cifar10-32`



The clock on the side of the metal building is gold and black

Kitchen with a wooden kitchen island and checkered floor

A pink bicycle leaning against a fence near a river

An empty kitchen with lots of tile blue counter top space

Figure 8: More text-image alignment results. We display four different prompts and the corresponding generation visualization from the original Stable Diffusion (1st row), DDPO (2nd row), DAG-DB (3rd row), and DAG-KL (4th row) models to compare their alignment ability.

problem to extreme and behaves mode collapse to some extent (only generating samples of a particular plane). While for our methods, the generation results are diverse and cover different classes of vehicles. Both methods achieve average log probability larger than -0.01, which means the probability of falling into target categories are very close to 1.