DIFFUSION ALIGNMENT BEYOND KL: VARIANCE MINIMISATION AS EFFECTIVE POLICY OPTIMISER

Zijing Ou[1]*, **Jacob Si**[1]*, **Junyi Zhu**[2], **Ondrej Bohdal**[2], **Mete Ozay**[2], **Taha Ceritli**[2], **Yingzhen Li**[1]
[1]Imperial College London, [2]Samsung R&D Institute UK
z.ou22@imperial.ac.uk

ABSTRACT

Diffusion alignment adapts pretrained diffusion models to sample from reward-tilted distributions along the denoising trajectory. This process naturally admits a Sequential Monte Carlo (SMC) interpretation, where the denoising model acts as a proposal and reward guidance induces importance weights. Motivated by this view, we introduce Variance Minimisation Policy Optimisation (VMPO), which formulates diffusion alignment as minimising the variance of log importance weights rather than directly optimising a Kullback-Leibler (KL) based objective. We prove that the variance objective is minimised by the reward-tilted target distribution and that, under on-policy sampling, its gradient coincides with that of standard KL-based alignment. This perspective offers a common lens for understanding diffusion alignment. Under different choices of potential functions and variance minimisation strategies, VMPO recovers various existing methods, while also suggesting new design directions beyond KL.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2020b; Rombach et al., 2022) have recently moved beyond large-scale pretraining to incorporate downstream objectives, such as aligning with human preferences (Wallace et al., 2024). A prominent line of work, commonly referred to as diffusion alignment, formalises this idea by adapting pretrained diffusion models such that their generated samples are biased toward high-reward regions of the data space (Black et al., 2023; Fan et al., 2023; Liu et al., 2025a; Xue et al., 2025). This direction has recently attracted significant attention, as alignment with reward signals enables controllable generation at inference time, allowing pretrained denoising models to be steered toward desired behaviours without retraining from scratch or compromising sample quality (Clark et al., 2023; Domingo-Enrich et al., 2024; Potaptchik et al., 2025; 2026).

Existing approaches, including Denoising Diffusion Policy Optimisation (DDPO) (Black et al., 2023), Diffusion policy optimisation with KL regularisation (DPOK) (Fan et al., 2023), and flow-based Group Relative Policy Optimisation (Flow-GRPO) (Liu et al., 2025a), typically cast diffusion alignment as a reinforcement learning problem (Schulman et al., 2015a; 2017; Shao et al., 2024), applying KL-regularised policy updates along the denoising trajectory. While these methods have demonstrated strong empirical performance, they are often derived from specific optimisation viewpoints, which can obscure their underlying connections and make it difficult to systematically design new variants. In addition, these formulations emphasise matching reward-tilted target distributions via KL minimisation, while alternative optimisation criteria remain relatively underexplored.

In this work, we revisit diffusion alignment through the lens of Sequential Monte Carlo (SMC) (Del Moral et al., 2006). By interpreting the denoising process as a proposal distribution and reward guidance as inducing importance weights, we show that diffusion alignment naturally gives rise to a variance minimisation problem (Richter et al., 2020). Motivated by this observation, we propose Variance Minimisation Policy Optimisation (VMPO), which frames alignment as minimising the variance of log importance weights along the trajectory. We analyse its theoretical properties and show that, under on-policy sampling, its gradient recovers that of standard KL-based alignment. Beyond providing a principled interpretation, this perspective offers a flexible framework: different choices of potential functions and variance objectives lead to optimisation rules closely related to

*Work done partially during the internship at Samsung R&D Institute UK.

existing methods, while also suggesting new design directions beyond KL regularisation. Empirically, we demonstrate the effectiveness of VMPO by finetuning Stable Diffusion 1.5 and 3.5 (Rombach et al., 2022; Esser et al., 2024) across a wide range of reward functions. We hope that this variance-minimisation viewpoint provides a useful conceptual tool for understanding diffusion alignment and for guiding the development of future alignment methods.

2 Background: Diffusion Alignment

Given a pretrained diffusion model $p_{\text{ref}}(x_{t-1}|x_t)$ and a reward model r, diffusion alignment aims at drawing samples from the reward-tilted target $p_{\text{tilt}}(x_{t-1}|x_t) \propto p_{\text{ref}}(x_{t-1}|x_t)\exp(\frac{r(x_{t-1})}{\beta})$ [1]. A naive way to sample from p_{tilt} is to apply self-normalised importance sampling (SNIS) (Owen, 2013):

$$x_{t-1} \sim \text{Cat}\left(x^i; \frac{w_t^i}{\sum_i w_t^i}\right),\ \log w_t^i = \log \frac{p_{\text{ref}}(x_{t-1}^i|x_t^i)\exp(r(x_{t-1}^i)/\beta)}{p_\theta(x_{t-1}^i|x_t^i)},\ x_{t-1}^i \sim p_\theta(x_{t-1}^i|x_t^i). \quad (1)$$

While straightforward, SNIS suffers from high variance with a poor proposal p_θ, which precipitates particle degeneracy and hinders adequate exploration of the state space (Del Moral et al., 2006). Theoretically, the optimal proposal with zero variance is $p_\theta(x_{t-1}|x_t) \propto p_{\text{ref}}(x_{t-1}|x_t)\exp(r(x_{t-1})/\beta)$, which inspires us to learn an optimal proposal by minimising a divergence between p_θ and p_{tilt}. A popular choice is the KL-divergence (Kullback & Leibler, 1951) (see Appendix A.2 for the derivation):

$$\operatorname*{argmin}_\theta \mathbb{KL}\left(p_\theta||p_{\text{tilt}}\right) \Leftrightarrow \operatorname*{argmax}_\theta \mathbb{E}_{p_\theta}[r(x_{t-1})] - \beta\mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\text{ref}}(x_{t-1}|x_t)). \quad (2)$$

To optimise p_θ using Equation (2), one can apply the policy gradient (Sutton et al., 1999), which induces the following surrogate objective:

$$\mathcal{J}_{\text{KL}}(t;\theta) = \mathbb{E}_{p_{\theta_{\text{old}}}}\left[\frac{p_\theta(x_{t-1}|x_t)}{p_{\theta_{\text{old}}}(x_{t-1}|x_t)} r(x_{t-1})\right] - \beta\mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\text{ref}}(x_{t-1}|x_t)). \quad (3)$$

Equation (3) has been widely adopted in reinforcement learning algorithms, such as proximal policy optimisation (PPO) (Schulman et al., 2017) and GRPO (Shao et al., 2024), which have recently been adapted in diffusion alignment, such as DDPO (Black et al., 2023) and Flow-GRPO (Liu et al., 2025a) (see Appendix A for a brief review). Rather than following the KL perspective, in the following, we propose a new diffusion alignment method from the perspective of variance minimisation.

3 VMPO: Variance Minimisation Policy Optimiser

In this section, we introduce VMPO: a policy optimiser with variance minimisation. Rather than directly maximising expected reward, VMPO learns the policy p_θ by minimising the variance of importance weights (Richter et al., 2020) [2]

$$\mathcal{L}_{\text{Var}}^h(t;\theta) = \frac{1}{2}\mathbb{V}_h(\log w_t) = \frac{1}{2}\mathbb{E}_h\left[\log \frac{p_{\text{ref}}(x_{t-1}|x_t)\exp(\frac{r(x_{t-1})}{\beta})}{p_\theta(x_{t-1}|x_t)} - \mathbb{E}_h\left[\log \frac{p_{\text{ref}}(x_{t-1}|x_t)\exp(\frac{r(x_{t-1})}{\beta})}{p_\theta(x_{t-1}|x_t)}\right]\right]^2 \quad (4)$$

where h is an arbitrary reference distribution sharing the same support as p_θ and p_{ref}. The validity of the loss Equation (4) is formalised by the following proposition:

Proposition 1. *The optimum of* $\mathcal{L}_{\text{Var}}^h(t;\theta)$ *satisfies* $p_{\theta^*} = p_{\text{tilt}}(x_{t-1}|x_t)$, $\theta^* = \operatorname{argmin}_\theta \mathcal{L}_{\text{Var}}^h(t;\theta)$. *Moreover,* $\nabla_\theta \mathcal{L}_{\text{Var}}^h(t;\theta)\big|_{h=p_\theta} = \nabla_\theta \mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\text{tilt}}(x_{t-1}|x_t))$.

[1] We omit the condition c when clear from context. In most settings, the reward function is defined only on the clean sample x_0. We defer the discussion of how to construct intermediate reward $r(x_t)$ to Appendix C.2

[2] For ease of exposition, we focus only on consecutive timesteps $(t-1, t)$. In Appendix C.1, we justify the variance minimisation objective from the lens of SMC, and further show that $T\sum_{t=1}^T \mathcal{L}_h^{\text{Var}}(t;\theta)$ upper-bounds $\frac{1}{2}\mathbb{V}_h(\log \frac{p_{\text{ref}}(x_{0:T})}{p_\theta(x_{0:T})} + \frac{1}{\beta}\sum_{t=0}^T r(x_t))$, i.e., the sum of local variance upper-bounds the variance of the importance weight of the joint trajectory distribution.

Figure 1: Visualisation of alignment dynamics over the training progress of SD1.5 with HPSv2. The generated images become more faithful to the prompt as the training continues (from left to right).

In practice, $\mathcal{L}^h_{\text{Var}}$ can be estimated using Monte Carlo samples:

$$\hat{\mathcal{L}}^h_{\text{Var}}(t;\theta) := \frac{1}{2(K-1)} \sum_{i=1}^{K} \Bigg(\underbrace{\log R_\theta(x^i_{t-1}|x^i_t) + \frac{1}{\beta} r(x^i_{t-1}) - \overline{\log R_\theta(x_{t-1}|x_t)} - \frac{1}{\beta}\overline{r(x_{t-1})}}_{:=A^i_t(\theta)} \Bigg)^2, \quad (5)$$

where $x^i_t \sim h$, $\overline{\log R_\theta(x_{t-1}|x_t)} = \frac{1}{N}\sum_{i=1}^K \log \frac{p_{\text{ref}}(x_{t-1}|x_t))}{p_\theta(x_{t-1}|x_t)}$, and $\overline{r(x_{t-1})} = \frac{1}{K}\sum_{i=1}^K r(x^i_{t-1})$. It is noteworthy that although $\hat{\mathcal{L}}^h_{\text{Var}}(t;\theta)$ is a biased estimator of $\mathcal{L}^h_{\text{Var}}(t;\theta)$ due to the quadratic nonlinearity of $(\cdot)^2$, its gradient, which has the form of

$$\nabla_\theta \hat{\mathcal{L}}^h_{\text{Var}}(t;\theta) = \frac{1}{(K-1)} \sum_{i=1}^{K} -A^i_t(\theta) \nabla_\theta \log p_\theta(x_{t-1}|x_t), \quad (6)$$

is an unbiased estimator to $-\nabla_\theta \mathcal{J}_{\text{KL}}(t;\theta)$ if we set $h = p_\theta$. Equation (5) induces a loss to train the policy: $\text{argmin}_\theta \frac{1}{(N-1)} \sum_{i=1}^K \sum_{t=1}^T -A^i_t(\theta_{\text{sg}}) \log p_\theta(x_{t-1}|x_t)$, which closely resembles the GRPO objective (see Appendix B.2 for details). Although straightforward, simple Monte Carlo estimation of the expectation $\mathbb{E}_h[\log w_t]$ typically has high variance, which in turn requires a high number of Monte Carlo samples to improve training stability, but reduces the training efficiency. Alternatively, we can instead amortise this expectation using a learnable function approximator. Concretely, we introduce a neural estimator $M_\phi(t)$ trained by

$$\phi^* = \underset{\phi}{\text{argmin}}\, \mathbb{E}_h \left[\left(\log R_\theta(x_{t-1}|x_t) + \frac{1}{\beta} r(x_{t-1}) - M_\phi(t) \right)^2 \right]. \quad (7)$$

By the identity $\mathbb{E}_h[w] = \text{argmin}_M \mathbb{E}_h[(w-M)^2]$, the optimal satisfies $M_{\phi^*}(t) = \mathbb{E}_h[\log w_t]$. Substituting this estimator into the variance objective yields the VMPO loss

$$\mathcal{L}_{\text{VMPO}}(\theta,\phi) = \mathbb{E}_{h,t} \left[\left(\log R_\theta(x_{t-1}|x_t) + \frac{1}{\beta} r(x_{t-1}) - M_\phi(t) \right)^2 \right] \quad (8)$$

We summarise the training procedure in Algorithm 1. For the choice of intermediate reward $r(x_t)$, we consider two options: return-to-go and difference, in Appendix C.2, which lead to two variants of our method: VMPO-R2G and VMPO-Diff, respectively. Moreover, in Appendix C.3, we further show that VMPO serves as a kaleidoscopic policy optimiser, connecting to a broad class of existing diffusion alignment methods through different choices of variance minimisation strategies.

Algorithm 1 VMPO

Require: Reference model p_{ref}, denoising policy p_θ, mean estimator M_ϕ, reward function r

1: **while** not converged **do**
2: Rollout $\tau = \{x_t\}_t$ with $p_\theta(x_{t-1}|x_t)$
3: For each transition $(x_{t-1}, x_t) \in \tau$: update θ, ϕ with Equation (8)
4: **end while**

Table 1: Results on Stable Diffusion v1.5 fine-tuned with HPSv2.

Method	**HPSv2** (↑)	**CLIPScore** (↑)	**ImageReward** (↑)	**DreamSim** (↑)
SD1.5 (Base)	0.2368 ± 0.0029	0.2717 ± 0.0032	0.0331 ± 0.0779	0.4389 ± 0.0116
GRPO	0.2684 ± 0.0035	0.2653 ± 0.0034	0.3449 ± 0.0758	0.3220 ± 0.0098
VMPO-R2G	0.2723 ± 0.0032	0.2713 ± 0.0030	0.3427 ± 0.0762	0.3673 ± 0.0115
VMPO-Diff	0.2822 ± 0.0040	0.2622 ± 0.0028	0.4973 ± 0.0780	0.2916 ± 0.0104

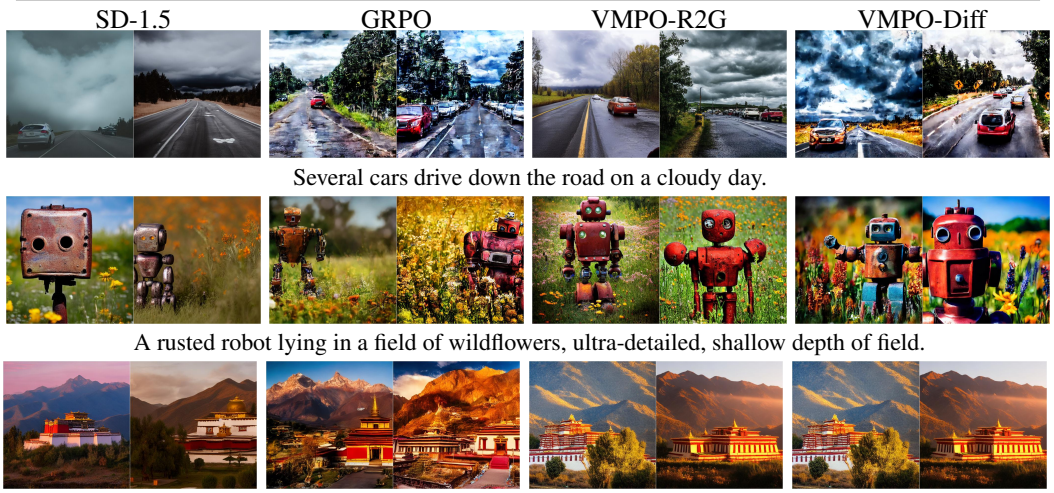

Figure 2: Illustration of the generated samples of different models.

4 Experiments

We evaluate VMPO by aligning Stable Diffusion v1.5 (Rombach et al., 2022) using Human Preference Score (HPSv2) (Wu et al., 2023b). Details of the experimental setup, along with additional results on Stable Diffusion v3.5 (Esser et al., 2024) and other reward functions, are deferred to Appendix D.

Setup. For training, we adopt low-rank adaptation (LoRA) (Hu et al., 2022) and leverage the photo and painting prompts from Wu et al. (2023b) as training prompts. For testing, we consider the 100 test prompts[3] collected by Domingo-Enrich et al. (2024). For each prompt, we generate 10 images and evaluate the CLIPScore (Hessel et al., 2021), ImageReward (Xu et al., 2023), and DreamSim (Fu et al., 2023) as the reported metrics.

Results and Discussion. We present the reward convergence curves in Figure 3. As shown in the figure, both VMPO and GRPO steadily improve HPSv2 over training, indicating effective reward optimisation. Notably, VMPO-Diff consistently attains higher reward values throughout training, suggesting improved sample efficiency for reward alignment compared to GRPO and VMPO-R2G. To further illustrate the alignment dynamics, Figure 1 visualises the evolution of generated images during training, where the images progressively align more faithfully to the prompts, qualitatively validating the effectiveness of VMPO. In Table 1, we quantitatively benchmark VMPO against GRPO. It shows that VMPO achieves higher HPSv2 and ImageReward scores, which are both preference-trained, CLIP-based rewards. However, we observe a drop in CLIPScore and DreamSim, which respectively measure text–image alignment and diversity, indicating that both VMPO and GRPO are susceptible to reward hacking. We illustrate this effect with representative samples in Figure 2 and additional examples provided in Figure 6.

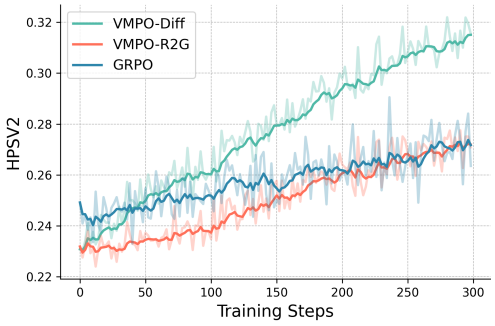

Figure 3: HPSv2 convergence curves of SD1.5.

[3] https://github.com/microsoft/soc-fine-tuning-sd/blob/main/prompt_files/benchmark_ir.json

5 Conclusion and Limitation

In this paper, we proposed variance minimisation policy optimiser (VMPO), a diffusion alignment method that learns the denoising policy by minimising the variance of importance weights. VMPO closely resembles proximal policy optimisation when Monte Carlo estimation is employed, while retaining substantial flexibility. By choosing different potential functions along sampling trajectories and alternative variance minimisation objectives, VMPO naturally connects to several existing approaches. Empirically, we demonstrated the effectiveness of VMPO by fine-tuning Stable Diffusion v1.5 and v3.5. We further discuss limitations and directions for future work in Appendix F.

References

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Cai, H., Cao, S., Du, R., Gao, P., Hoi, S., Hou, Z., Huang, S., Jiang, D., Jin, X., Li, L., et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Cardoso, G., Idrissi, Y. J. E., Corff, S. L., and Moulines, E. Monte carlo guided diffusion for bayesian linear inverse problems. *arXiv preprint arXiv:2308.07983*, 2023.

Chen, W., Ou, Z., and Li, Y. Neural flow samplers with shortcut models. *arXiv preprint arXiv:2502.07337*, 2025a.

Chen, W., Zhang, M., He, J., Ou, Z., Hernández-Lobato, J. M., Schölkopf, B., and Barber, D. Diffratio: Training one-step diffusion models without teacher supervision. *arXiv e-prints*, pp. arXiv–2502, 2025b.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Del Moral, P., Doucet, A., and Jasra, A. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Domingo-Enrich, C., Drozdzal, M., Karrer, B., and Chen, R. T. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

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. *arXiv preprint arXiv:2304.06767*, 2023.

Dou, Z. and Song, Y. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Feng, S., Kong, X., Ma, S., Zhang, A., Yin, D., Wang, C., Pang, R., and Yang, Y. Step-by-step reasoning for math problems via twisted sequential monte carlo. *arXiv preprint arXiv:2410.01920*, 2024.

Fu, S., Tamir, N., Sundaram, S., Chai, L., Zhang, R., Dekel, T., and Isola, P. DreamSim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

He, J., Hernández-Lobato, J. M., Du, Y., and Vargas, F. Rne: a plug-and-play framework for diffusion density estimation and inference-time control. *arXiv preprint arXiv:2506.05668*, 2025a.

He, J., Jeha, P., Potaptchik, P., Zhang, L., Hernández-Lobato, J. M., Du, Y., Syed, S., and Vargas, F. Crepe: Controlling diffusion with replica exchange. *arXiv preprint arXiv:2509.23265*, 2025b.

Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., and Choi, Y. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Johansen, A. A tutorial on particle filtering and smoothing: Fifteen years later. 2009.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., and Levy, O. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Kool, W., van Hoof, H., and Welling, M. Buy 4 REINFORCE samples, get a baseline for free!, 2019. URL https://openreview.net/forum?id=r1lgTGL5DE.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

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 preprint arXiv:2302.12192*, 2023.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Li, J., Cui, Y., Huang, T., Ma, Y., Fan, C., Yang, M., and Zhong, Z. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

Liang, Z., Yuan, Y., Gu, S., Chen, B., Hang, T., Li, J., and Zheng, L. Step-aware preference optimization: Aligning preference with denoising performance at each step. *arXiv preprint arXiv:2406.04314*, 2(5):7, 2024.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumbley, M. D. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., and Ouyang, W. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025a.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Liu, Z., Xiao, T. Z., Liu, W., Bengio, Y., and Zhang, D. Efficient diversity-preserving diffusion alignment via gradient-informed gflownets. *arXiv preprint arXiv:2412.07775*, 2024.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Luo, W., Hu, T., Zhang, S., Sun, J., Li, Z., and Zhang, Z. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

Ou, Z., Zhang, M., Zhang, A., Xiao, T. Z., Li, Y., and Barber, D. Improving probabilistic diffusion models with optimal diagonal covariance matching. *arXiv preprint arXiv:2406.10808*, 2024.

Ou, Z., Pani, C., and Li, Y. Inference-time scaling of discrete diffusion models via importance weighting and optimal proposal design. *arXiv e-prints*, pp. arXiv–2505, 2025a.

Ou, Z., Zhang, R., and Li, Y. Discrete neural flow samplers with locally equivariant transformer. *arXiv preprint arXiv:2505.17741*, 2025b.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Owen, A. B. Monte carlo theory, methods and examples, 2013.

Pan, L., Malkin, N., Zhang, D., and Bengio, Y. Better training of gflownets with local credit and incomplete trajectories. In *International Conference on Machine Learning*, pp. 26878–26890. PMLR, 2023.

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

Potaptchik, P., Lee, C.-K., and Albergo, M. S. Tilt matching for scalable sampling and fine-tuning. *arXiv preprint arXiv:2512.21829*, 2025.

Potaptchik, P., Saravanan, A., Mammadov, A., Prat, A., Albergo, M. S., and Teh, Y. W. Meta flow maps enable scalable reward alignment. *arXiv preprint arXiv:2601.14430*, 2026.

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

Puri, I., Sudalairaj, S., Xu, G., Xu, K., and Srivastava, A. A probabilistic inference approach to inference-time scaling of llms using particle-based monte carlo methods. *arXiv preprint arXiv:2502.01618*, 2025.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial intelligence and statistics*, pp. 814–822. PMLR, 2014.

Richter, L., Boustati, A., Nüsken, N., Ruiz, F., and Akyildiz, O. D. VarGrad: A low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33: 13481–13492, 2020.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016.

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. *arXiv preprint arXiv:2202.00512*, 2022.

Salimans, T. and Knowles, D. A. On using control variates with stochastic approximation for variational bayes and its connection to stochastic linear regression. *arXiv preprint arXiv:1401.1022*, 2014.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015a.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

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

Shah, V., Obando-Ceron, J., Jain, V., Bartoldson, B., Kailkhura, B., Mittal, S., Berseth, G., Castro, P. S., Bengio, Y., Malkin, N., et al. A comedy of estimators: On KL regularization in RL training of LLMs. *arXiv preprint arXiv:2512.21852*, 2025.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Skreta, M., Akhound-Sadegh, T., Ohanesian, V., Bondesan, R., Aspuru-Guzik, A., Doucet, A., Brekelmans, R., Tong, A., and Neklyudov, K. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

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. *arXiv preprint arXiv:2402.15194*, 2024.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Wang, L., Zhang, M., Ou, Z., and Barber, D. VarDiU: A Variational Diffusive Upper bound for one-step diffusion sistillation. *arXiv preprint arXiv:2508.20646*, 2025a.

Wang, Y., Zang, Y., Li, H., Jin, C., and Wang, J. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025b.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.-m., Bai, S., Xu, X., Chen, Y., et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023a.

Wu, L., Han, Y., Naesseth, C. A., and Cunningham, J. P. Reverse diffusion sequential monte carlo samplers. *arXiv preprint arXiv:2508.05926*, 2025b.

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 preprint arXiv:2306.09341*, 2023b.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Xue, Z., Wu, J., Gao, Y., Kong, F., Zhu, L., Chen, M., Liu, Z., Liu, W., Guo, Q., Huang, W., et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

You, Z., Cai, X., Gu, J., Xue, T., and Dong, C. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14483–14494, 2025.

Yuan, H., Chen, Z., Ji, K., and Gu, Q. Self-play fine-tuning of diffusion models for text-to-image generation. *Advances in Neural Information Processing Systems*, 37:73366–73398, 2024.

Zhang, D., Zhang, Y., Gu, J., Zhang, R., Susskind, J., Jaitly, N., and Zhai, S. Improving gflownets for text-to-image diffusion alignment. *arXiv preprint arXiv:2406.00633*, 2024.

Zhang, K., Hong, Y., Bao, J., Jiang, H., Song, Y., Hong, D., and Xiong, H. Gvpo: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*, 2025a.

Zhang, L., Potaptchik, P., He, J., Du, Y., Doucet, A., Vargas, F., Dau, H.-D., and Syed, S. Accelerated parallel tempering via neural transports. *arXiv preprint arXiv:2502.10328*, 2025b.

Zhang, M., Chen, W., He, J., Ou, Z., Hernández-Lobato, J. M., Schölkopf, B., and Barber, D. Towards training one-step diffusion models without distillation. *arXiv preprint arXiv:2502.08005*, 2025c.

Zhang, Y., Liu, Y., Yuan, H., Yuan, Y., Gu, Q., and Yao, A. C.-C. On the design of KL-regularized policy gradient algorithms for LLM reasoning. *arXiv preprint arXiv:2505.17508*, 2025d.

Zhao, S., Brekelmans, R., Makhzani, A., and Grosse, R. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024.

Zheng, K., Chen, H., Ye, H., Wang, H., Zhang, Q., Jiang, K., Su, H., Ermon, S., Zhu, J., and Liu, M.-Y. DiffusionNFT: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

Zhou, M., Zheng, H., Wang, Z., Yin, M., and Huang, H. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

Zhu, X., Cheng, D., Zhang, D., Li, H., Zhang, K., Jiang, C., Sun, Y., Hua, E., Zuo, Y., Lv, X., et al. Flowrl: Matching reward distributions for llm reasoning. *arXiv preprint arXiv:2509.15207*, 2025.

Appendix for "Diffusion Alignment Beyond KL: Variance Minimisation as Effective Policy Optimiser"

CONTENTS

A DIFFUSION ALIGNMENT: A TALE OF TWO VIEWS

Let $q(x_0)$ be the data distribution, denoising diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Kingma et al., 2021) construct a Markovian process $q(x_{0:T}) = q(x_0)\prod_{t=1}^{T} q(x_t|x_{t-1})$, with $q(x_t|x_{t-1}) = \mathcal{N}(\alpha_t/\alpha_{t-1}x_t, (1-\alpha_t^2/\alpha_{t-1}^2)I)$. By setting $\sigma_t^2 + \alpha_t^2 = 1$, this noising process has a simple marginal in the form of $q(x_t|x_0) = \mathcal{N}(x_t; \alpha_t x_0; \sigma_t^2 I)$, where $\{\alpha_t, \sigma_t\}$ are pre-defined signal-noise schedules. Diffusion models parametrise the denoising process as a Markovian process as well, which takes the form of

$$\begin{aligned} p_\theta(x_0) &= \int p_\theta(x_{0:T})\,\mathrm{d}x_{0:T} = \int p_\theta(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)\,\mathrm{d}x_{1:T}, \\ p_\theta(x_{t-1}|x_t) &= \mathcal{N}\left(x_{t-1}; \frac{\alpha_t\sigma_{t-1}^2}{\alpha_{t-1}\sigma_t^2}x_t + \frac{\alpha_{t-1}^2 - \alpha_t^2}{\alpha_{t-1}\sigma_t^2}\hat{x}_\theta(x_t, t)\right) \end{aligned} \tag{9}$$

in which $\hat{x}_\theta(x_t, t)$ is a neural network to predict the clean data x_0 given the noisy input $x_t = \alpha_t x_0 + \sigma_t\epsilon$. Thus, diffusion models can be trained via data prediction loss

$$\mathcal{L}(\theta) = \mathbb{E}_{t, x_0\sim q(x_0), \epsilon\sim\mathcal{N}(0,I)}\|x_0 - \hat{x}_\theta(\alpha_t x_0 + \sigma_t\epsilon, t)\|^2. \tag{10}$$

In condition generation, such as text-to-image generation, one can parametrise the diffusion model as

$$p_\theta(x_0; c) = \int p_\theta(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t; c)\,\mathrm{d}x_{1:T}, \tag{11}$$

where c denotes the condition (e.g., text), and replace $\hat{x}_t(\alpha_t x_0 + \sigma_t \epsilon, t)$ with $\hat{x}_t(\alpha_t x_0 + \sigma_t \epsilon, t, c)$. In this paper, we study diffusion alignment. Given a diffusion model p_θ and a reward model r, our goal is to maximise the reward of the generated samples. We present two complementary perspectives for achieving this objective and show that they ultimately lead to the same optimisation problem. In what follows, we occasionally ignore the dependence on c without loss of generality.

A.1 Diffusion Alignment as Policy Optimisation

By the Markov property of denoising diffusion models (cf. Equation (9)), the denoising process can be cast as a finite-horizon Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P_0, P, R)$ (Black et al., 2023; Fan et al., 2023), where

$$s_t = (x_{T-t}, c), \quad a_t = x_{T-t-1}, \quad P_0(s_0) = (\mathcal{N}(0, I), p(c)), \quad P(s_{t+1}|s_t, a_t) = (\delta_{a_t}, \delta_c)$$
$$R(s_t, a_t) = r(s_{t+1}) \text{ only if } t = T-1 \text{ else } 0, \quad \pi_\theta(a_t|s_t) = p_\theta(x_{T-t-1}|x_{T-t}, c).$$

Here, s_t and a_t denote the state and action at time step t; P_0 and P are the initial state distribution and the deterministic transition kernel; R is a sparse terminal reward defined only at the final denoising step; and π_θ corresponds to the denoising policy parameterised by θ. Under this formulation, diffusion alignment amounts to optimising the policy π_θ to maximise the expected terminal reward. This leads to the following diffusion alignment objective (Black et al., 2023)

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} G_t \right] \tag{12}$$

where $\tau \triangleq (x_0, \ldots, x_T)$ denotes the sample trajectory and G_t is the return-to-go, which in the diffusion alignment setting reduces to $G_t = r(x_0)$. Applying the policy gradient theorem, one can show that (Fan et al., 2023, Lemma 4.1):

$$\begin{aligned}
\nabla_\theta \mathcal{J}(\theta) &= \mathbb{E}_{\tau \sim \mu_\theta} \left[\sum_{t=0}^{T} G_t \nabla_\theta \log \pi_\theta(a_t|x_t) \right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{p_\theta(x_{t-1}|x_t)} [r(x_{t-1}) \nabla_\theta \log p_\theta(x_{t-1}|x_t)] \\
&= \nabla_\theta \mathbb{E}_{p_\theta(x_0)} [r(x_0)]
\end{aligned} \tag{13}$$

This expression shows that diffusion alignment can be viewed as reinforcing denoising transitions that lead to high-reward final samples.

A.2 Diffusion Alignment as Probability Inference

Alternatively, diffusion alignment can be formulated from a probabilistic inference perspective. Given a pretrained diffusion model $p_{\text{ref}}(x_{t-1}|x_t)$ and a reward model $r(x)$, one can define a reward-tilted target distribution $p_{\text{tilt}}(x_{t-1}|x_t) \propto p_{\text{ref}}(x_{t-1}|x_t) \exp(r(x_{t-1})/\beta)$ where $\beta > 0$ controls the strength of reward guidance. A natural approach is to learn a model p_θ that approximates p_{tilt} by minimising the KL divergence $\mathbb{KL}(p_\theta \| p_{\text{tilt}})$. Taking the gradient with respect to θ, we obtain

$$\begin{aligned}
&\nabla_\theta \mathbb{KL}\left(p_\theta(x_{t-1}|x_t) || p_{\text{tilt}}(x_{t-1}|x_t)\right) \\
&= \int \left(\log \frac{p_\theta(x_{t-1}|x_t)}{p_{\text{ref}}(x_{t-1}|x_t)} - \frac{1}{\beta} r(x_{t-1}) \right) \nabla_\theta p_\theta(x_{t-1}|x_t) \, \mathrm{d}x_{t-1} \\
&\quad + \cancelto{0}{\int \log Z \nabla_\theta p_\theta(x_{t-1}|x_t) \, \mathrm{d}x_{t-1}} + \cancelto{0}{\int p_\theta(x_{t-1}|x_t) \nabla_\theta \log p_\theta(x_{t-1}|x_t) \, \mathrm{d}x_{t-1}} \\
&= \nabla_\theta \mathbb{E}_{p_\theta(x_{t-1}|x_t)} \left[-\frac{1}{\beta} r(x_{t-1}) \right] + \nabla_\theta \mathbb{KL}(p_\theta(x_{t-1}|x_t) || p_{\text{ref}}(x_{t-1}|x_t)).
\end{aligned} \tag{14}$$

where we have used the fact that the normalisation constant of p_{tilt} does not depend on θ. This leads to the following alignment objective:

$$\operatorname*{argmax}_\theta \sum_{t=1}^{T} \mathbb{E}_{p_\theta(x_{t-1}|x_t)} \left[r(x_{t-1}) \right] - \beta \mathbb{KL}(p_\theta(x_{t-1}|x_t) || p_{\text{ref}}(x_{t-1}|x_t)) \tag{15}$$

We observe that this objective closely resembles the one in Equation (13), but includes an additional KL regularisation term that constrains the learned policy to remain close to the pretrained diffusion model. Such KL regularisation is standard in reinforcement learning (Schulman et al., 2015a; 2017; Haarnoja et al., 2018; Levine, 2018) and variational inference (Kingma & Welling, 2013; Blei et al., 2017), where it serves to stabilise optimisation, prevent mode collapse, and preserve the prior.

A.3 REINFORCE, PPO, AND GRPO

REINFORCE (Williams, 1992) is a Monte Carlo policy-gradient estimator that expresses the gradient of the expected return as an expectation of the reward weighted by the score function:

$$\nabla_\theta \mathbb{E}_{\pi_\theta}[G_t] = \mathbb{E}_{\pi_\theta}[G_t \nabla_\theta \log \pi_\theta(a_t|s_t)]. \tag{16}$$

Although unbiased, the REINFORCE estimator typically suffers from high variance. A standard variance-reduction technique is to introduce a control variate (Ranganath et al., 2014) in the form of a baseline $b(s_t)$, which is subtracted from the return-to-go G_t. In reinforcement learning, a common choice is the value function $V_{\pi_\theta}(s_t)$ (Schulman et al., 2015b), which is action-independent and therefore preserves unbiasedness, since $\mathbb{E}_{\pi_\theta}[b(s_t)\nabla_\theta \log \pi_\theta(a_t|s_t)] = 0$. This yields the following gradient estimator:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}(G_t - b(s_t))\nabla_\theta \log \pi_\theta(a_t|s_t)\right] = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\hat{A}_t \nabla_\theta \log \pi_\theta(a_t|s_t)\right], \tag{17}$$

where $\hat{A}_t = G_t - b(s_t)$ is an estimate of the advance $Q_{\pi_\theta}(s_t, a_t) - V_{\pi_\theta}(s_t)$, where Q denotes the Q-value. This leads to the REINFORCE surrogate objective:

$$\mathcal{J}_{\text{REINFORCE}}(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{\text{sg}}}}\left[\sum_{t=0}^{T}\hat{A}_t \log \pi_\theta(a_t|s_t)\right]. \tag{18}$$

However, $\mathcal{J}_{\text{REINFORCE}}$ is inefficient in practice, since it requires sample trajectries for each policy update. This can also lead to excessively large policy updates, resulting in instability and degraded performance. Proximal Policy Optimisation (PPO) (Schulman et al., 2017) addresses these issues by explicitly constraining the deviation between the updated policy π_θ and the reference policy $\pi_{\theta_{\text{sg}}}$, by applying importance sampling with the weight $\rho_t \triangleq \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{sg}}}(a_t|s_t)}$, and clipping it to remain within a trust region. The PPO clipped surrogate objective is given by

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{\tau\sim\pi_{\theta_{\text{sg}}}}\left[\sum_{t=0}^{T}\min\left(\rho_t(\theta)\hat{A}_t, \text{clip}\left(\rho_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{19}$$

where $\epsilon > 0$ controls the size of the trust region. Intuitively, PPO preserves the variance-reduction benefits of advantage-weighted policy gradients while preventing overly aggressive updates that would invalidate the on-policy approximation underlying REINFORCE.

While PPO stabilises optimisation by constraining policy updates, it relies on an explit value function V_{π_θ} to construct the advantage estimates. In many settings, particularly those with sparse, trajectory-level rewards, learning an accurate reward value function can be difficult and may introduce addtional instability. Group Relative Policy Optimisation (GRPO) (Shao et al., 2024) circumvents this issue by eliminating the value function altogether and instead using relative advantages computed within groups of samples. Concretely, for a given state s_t, GRPO samples a group of actions $\{a_t^i\}_{i=1}^K \sim \pi_{\theta_{\text{sg}}}(\cdot|s_t)$ and computes their corresponding returns $\{G_t^i\}_{i=1}^K$. The advantages for each action is then defined relative to the group mean:

$$\hat{A}_t^i = G_t^i - \frac{1}{K}\sum_{j=1}^{K} G_t^i. \tag{20}$$

This group-normalised advantage can be written as $\hat{A}_t^i = \frac{K-1}{K}\left(G_t^i - \frac{1}{K}\sum_{j\neq i} G_t^j\right)$, in which the second term is a leave-one-out (LOO) baseline (Salimans & Knowles, 2014; Kool et al., 2019). This

baseline acts as a control variate, reducing variance without requiring a learned value function.[4] GRPO combines this relative advantage estimator with a PPO-style clipped objective, yielding the objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\{\tau_i\}\sim\pi_{\theta_{\text{sg}}}}\left[\frac{1}{K}\sum_{i=1}^{K}\sum_{t=0}^{T}\min\left(\rho_t(\theta)\hat{A}_t^i, \text{clip}\left(\rho_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^i\right)\right]. \tag{21}$$

From this viewpoint, GRPO inherits the stability guarantees of PPO through proximal updates, while avoiding the need to learn a parametric value function. This makes GRPO particularly appealing in settings where rewards are defined at the trajectory level or where value estimation is challenging, as is the case in diffusion alignment.

In practice, policy optimisation is often further regularised by penalising deviations from a reference policy π_{ref} via a KL divergence term, leading to the following objective:

$$\underset{\theta}{\text{argmax}}\, \mathcal{J}_{\text{REINFORCE/PPO/GRPO}}(\theta) - \beta\sum_{t=0}^{T}\mathbb{KL}(\pi_\theta(a_t|s_t)||\pi_{\text{ref}}(a_t|s_t)). \tag{22}$$

Such KL regularisation plays a central role in stabilising training by constraining the updated policy to remain close to a fixed reference. In many practical implementations, particularly in reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) for large language models, this regularisation is applied implicitly rather than as an explicit penalty in the optimisation objective (Zhang et al., 2025d; Shah et al., 2025). Specifically, the KL term is absorbed into the reward before advantage estimation by modifying the return as

$$G_t' = G_t - \beta\log\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{ref}}(a_t|s_t)}. \tag{23}$$

The resulting modified return is then used to compute the advantage $\hat{A}_t$, effectively enforcing KL regularisation at the level of the reward signal. This implicit formulation contrasts with explicitly adding a KL penalty to the final objective, as in Equation (22), while yielding similar regularising effects in practice.

B Proofs and Derivations

B.1 Proof of Proposition 1

Proposition 1. *The optimum of* $\mathcal{L}^h_{\text{Var}}(t;\theta)$ *satisfies* $p_{\theta^*} = p_{\text{tilt}}(x_{t-1}|x_t)$, $\theta^* = \text{argmin}_\theta\, \mathcal{L}^h_{\text{Var}}(t;\theta)$. *Moreover,* $\nabla_\theta\mathcal{L}^h_{\text{Var}}(t;\theta)\big|_{h=p_\theta} = \nabla_\theta\mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\text{tilt}}(x_{t-1}|x_t))$.

Proof. We first prove that: $p_{\theta^*} = p_{\text{tilt}}(x_{t-1}|x_t)$, $\theta^* = \text{argmin}_\theta\, \mathcal{L}^h_{\text{Var}}(t;\theta)$. This can be straightforwardly validated by observing that the minimum variance implies the importance weight is a constant for all (x_{t-1}, x_t). Therefore, we have

$$\frac{p_{\text{ref}}(x_{t-1}|x_t)\exp(\frac{r(x_{t-1})}{\beta})}{p_{\theta^*}(x_{t-1}|x_t)} = const. \quad \forall x_{t-1}, x_t. \tag{24}$$

Therefore, $p_{\theta^*}(x_{t-1}|x_t) = p_{\text{tilt}}(x_{t-1}|x_t) \propto p_{\text{ref}}(x_{t-1}|x_t)\exp(\frac{r(x_{t-1})}{\beta})$. We then proceed to prove $\nabla_\theta\mathcal{L}^h_{\text{Var}}(t;\theta)\big|_{h=p_\theta} = \nabla_\theta\mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\text{tilt}}(x_{t-1}|x_t))$. To see this, we define

$$w_\theta(x_{t-1}, x_t) = \frac{p_{\text{ref}}(x_{t-1}|x_t)\exp(\frac{r(x_{t-1})}{\beta})}{p_\theta(x_{t-1}|x_t)} \quad \Rightarrow \quad \nabla_\theta\log w_\theta = -\nabla_\theta\log p_\theta(x_{t-1}|x_t) \tag{25}$$

[4] We intentionally do not normalise the advantage by the group standard deviation, following Dr. GRPO (Liu et al., 2025b). Mean-only normalisation yields a REINFORCE estimator with a leave-one-out control variate whose gradient is an unbiased estimator of $\nabla_\theta\mathcal{J}(\theta)$. In contrast, the original GRPO formulation additionally normalises by the group standard deviation, which introduces bias into the policy gradient estimator.

Using the definition from Equation (4), we see that

$$\begin{aligned}\nabla_\theta \mathcal{L}^h_{\mathrm{Var}}(t;\theta) &= \frac{1}{2}\nabla_\theta \int h \log^2 w_\theta \,\mathrm{d}x_{t-1} - \frac{1}{2}\nabla_\theta \left(\int h \log w_\theta \,\mathrm{d}x_{t-1}\right)^2 \\ &= \int h \log w_\theta \nabla_\theta \log w_\theta \,\mathrm{d}x_{t-1} - \left(\int h \log w_\theta \,\mathrm{d}x_{t-1}\right)\left(\int h \nabla_\theta \log w_\theta \,\mathrm{d}x_{t-1}\right).\end{aligned}$$

By setting $h = p_\theta$ and noting that $\nabla_\theta \log w_\theta = -\nabla_\theta \log p_\theta(x_{t-1}|x_t)$, we have

$$\begin{aligned}\nabla_\theta \mathcal{L}^h_{\mathrm{Var}}(t;\theta)\big|_{h=p_\theta} &= \int -\log w_\theta \nabla_\theta p_\theta \,\mathrm{d}x_{t-1} - \left(\int p_\theta \log w_\theta \,\mathrm{d}x_{t-1}\right) \cancelto{0}{\left(\int -p_\theta \nabla_\theta \log p_\theta \,\mathrm{d}x_{t-1}\right)} \\ &= \nabla_\theta \mathbb{E}_{p_\theta(x_{t-1}|x_t)}\left[-\frac{1}{\beta} r(x_{t-1})\right] + \int \log \frac{p_\theta(x_{t-1}|x_t)}{p_{\mathrm{ref}}(x_{t-1}|x_t)} \nabla_\theta p_\theta(x_{t-1}|x_t) \,\mathrm{d}x_{t-1} \\ &= \nabla_\theta \mathbb{E}_{p_\theta(x_{t-1}|x_t)}\left[-\frac{1}{\beta} r(x_{t-1})\right] + \nabla_\theta \mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\mathrm{ref}}(x_{t-1}|x_t)) \\ &= \nabla_\theta \mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\mathrm{tilt}}(x_{t-1}|x_t)),\end{aligned}$$

which completes the proof. □

B.2 Proof of Equation (6)

In this section, we show that although the Monte Carlo estimator in Equation (5) is biased, its gradient remains an unbiased estimator when the reference distribution is chosen as $h = p_\theta$. We establish this by explicitly analysing the gradient of the estimator:

$$\begin{aligned}\nabla_\theta \hat{\mathcal{L}}^h_{\mathrm{Var}}(t;\theta) &= \frac{1}{(K-1)} \sum_{i=1}^K A_t^i(\theta) \left(\nabla_\theta \log R_\theta(x^i_{t-1}|x^i_t) - \nabla_\theta \overline{\log R_\theta(x_{t-1}|x_t)}\right) \\ &= \frac{1}{(K-1)} \left(\sum_{i=1}^K A_t^i(\theta) \nabla_\theta \log R_\theta(x^i_{t-1}|x^i_t) - \nabla_\theta \overline{\log R_\theta(x_{t-1}|x_t)} \cancelto{0}{\textstyle\sum_{i=1}^K A_t^i}\right) \\ &= \frac{1}{(K-1)} \sum_{i=1}^K -A_t^i(\theta) \nabla_\theta \log p_\theta(x_{t-1}|x_t). \end{aligned} \tag{26}$$

To simplify notation, define

$$f_t^i \triangleq \log R_\theta(x^i_{t-1}|x^i_t) + \frac{1}{\beta} r(x^i_{t-1}), \quad \bar{f}_t = \frac{1}{K}\sum_{j=1}^K f_t^j. \tag{27}$$

Then we can rewrite

$$\nabla_\theta \hat{\mathcal{L}}^h_{\mathrm{Var}}(t;\theta) = \frac{-1}{K-1} \sum_{i=1}^K \left(f_t^i - \bar{f}_t\right) \nabla_\theta \log p_\theta(x^i_{t-1}|x^i_t) \tag{28}$$

$$= -\sum_{i=1}^K \left(\frac{1}{K} f_t^i - \frac{1}{K(K-1)} \sum_{j\neq i} f_t^j\right) \nabla_\theta \log p_\theta(x^i_{t-1}|x^i_t). \tag{29}$$

Now setting $h = p_\theta$, the second term inside the parentheses vanishes in expectation

$$\mathbb{E}_{p_\theta(x_{t-1}|x_t)}\left[\nabla_\theta \log p_\theta(x^i_{t-1}|x^i_t) \sum_{j\neq i} f_t^j\right] = \sum_{j\neq i} f_t^j \nabla_\theta \int p_\theta(x^i_{t-1}|x^i_t) \,\mathrm{d}x^i_{t-1} = 0. \tag{30}$$

since the score function integrates to zero. As a result, taking expectations yields

$$\begin{aligned}\mathbb{E}_{p_\theta}\left[\nabla_\theta \hat{\mathcal{L}}^h_{\text{Var}}(t;\theta)\Big|_{h=p_\theta}\right] &= -\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{p_\theta}\left[f_t^i \nabla_\theta \log p_\theta(x_{t-1}^i|x_t^i)\right] \\ &= -\mathbb{E}_{p_\theta}\left[\left(\log\frac{p_{\text{ref}}(x_{t-1}|x_t))}{p_\theta(x_{t-1}|x_t)} + \frac{1}{\beta}r(x_{t-1})\right)\nabla_\theta \log p_\theta(x_{t-1}|x_t)\right] \\ &= \nabla_\theta \mathbb{E}_{p_\theta(x_{t-1}|x_t)}\left[-\frac{1}{\beta}r(x_{t-1})\right] + \nabla_\theta \mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\text{ref}}(x_{t-1}|x_t)) \\ &= \nabla_\theta \mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\text{tilt}}(x_{t-1}|x_t)).\end{aligned}$$

Hence, although $\hat{\mathcal{L}}^h_{\text{Var}}(t;\theta)$ is a biased estimator of the variance objective, its gradient is an unbiased estimator of the true gradient when $h = p_\theta$.

Remark. The term $\bar{f}_t$ in Equation (28) acts as a leave-one-out baseline, reducing variance without introducing bias. Motivated by Equation (26), we may instead define the following surrogate loss:

$$\hat{\mathcal{L}}^{\text{Var}}_{h=p_\theta}(\theta) = \frac{1}{(K-1)}\sum_{i=1}^{K} -A_t^i(\theta_{\text{sg}})\log p_\theta(x_{t-1}|x_t). \tag{31}$$

Applying importance sampling and PPO-style clipping yields the following objective

$$\underset{\theta}{\text{argmax}}\frac{1}{K-1}\sum_{i=1}^{K}\sum_{t=0}^{T}\min\left(\rho_t^i(\theta)A_t^i, \text{clip}\left(\rho_t^i(\theta), 1-\epsilon, 1+\epsilon\right)A_t^i\right) \tag{32}$$

with $\rho_t^i(\theta) = \frac{p_\theta(x_{t-1}^i|x_t^i)}{p_{\theta_{\text{old}}}(x_{t-1}^i|x_t^i)}$, $\{x_t^i\} \sim p_{\theta_{\text{old}}}$, and importantly

$$A_t^i = \log R_\theta(x_{t-1}^i|x_t^i) + \frac{1}{\beta}r(x_{t-1}^i) - \overline{\log R_\theta(x_{t-1}|x_t)} - \frac{1}{\beta}\overline{r(x_{t-1})}. \tag{33}$$

Compared to the advantage used in GRPO (see Equation (20)), this advantage contains two additional components, $\log R_\theta(x_{t-1}^i|x_t^i)$ and $\overline{\log R_\theta(x_{t-1}|x_t)}$. The latter again acts as a LOO baseline, while the former corresponds to a KL-regularisation term subtracted from the reward, as in Equation (23).

C Holistic VMPO Kaleidoscopes

In Appendix 3, we derive the VMPO objective by restricting attention to consecutive state pairs (x_{t-1}, x_t). While Proposition 1 establishes the validity of this formulation, it characterises only a "local" optimisation perspective. In this section, we reinterpret the VMPO objective through the lens of sequential Monte Carlo (Ou et al., 2025a;b; He et al., 2025a). From this viewpoint, we show that the variance of the log importance weight of the full trajectory is upper-bounded by the sum of per-timestep variances,

$$\begin{gathered}\mathbb{V}_h(\log w_{0:T}) \le T\sum_t \mathbb{V}_h(\log w_t) \\ w_t = \frac{p_{\text{ref}}(x_{t-1}|x_t)\exp\left(\frac{1}{\beta}r(x_{t-1})\right)}{p_\theta(x_{t-1}|x_t)}; \quad w_{0:T} = \frac{p_{\text{ref}}(x_{0:T})\exp\left(\frac{1}{\beta}\sum_t r(x_t)\right)}{p_\theta(x_{0:T})}.\end{gathered} \tag{34}$$

This result implies that minimising $\sum_t \mathbb{V}_h(\log w_t)$ effectively controls the variance of the importance weights associated with the joint trajectory distribution. In turn, this provides a principled justification for the VMPO objective from a "global", trajectory-level perspective. We proceed by first reviewing the SMC framework and establishing the above inequality. Building on this interpretation, we then propose several VMPO variants motivated by the SMC perspective.

C.1 Demystifying VMPO

In Sequential Monte Carlo (Del Moral et al., 2006), we construct a sequence of target distributions $p_{\text{tilt}}(x_{t:T})$ by tilting the base distribution $p_{\text{ref}}(x_{t:T})$ with a collection of potential functions $U(x_{t:T})$.

Starting from the terminal distribution $p_{\text{tilt}}(x_T) \propto p_{\text{ref}}(x_T)U(x_T)$, the intermediate targets are defined recursively as

$$p_{\text{tilt}}(x_{t:T}) \propto p_{\text{ref}}(x_{t:T}) \prod_{s=t}^{T} U(x_{s:T}) = p_{\text{ref}}(x_t|x_{t+1})U(x_{t:T})p_{\text{tilt}}(x_{t+1:T}). \tag{35}$$

The second equality follows from the Markov property of the pretrained diffusion model p_{ref}, which implies $p_{\text{ref}}(x_t|x_{t+1:T}) = p_{\text{ref}}(x_t|x_{t+1})$. The potentials are required to satisfy the constraint

$$\prod_{t=0}^{T} U(x_{t:T}) = \exp\left(\frac{1}{\beta} r(x_0)\right), \tag{36}$$

which ensures that the marginal distribution over the final sample obeys $p_{\text{tilt}}(x_0) \propto p_{\text{ref}}(x_0)\exp(r(x_0)/\beta))$. Notably, this constraint can be satisfied by multiple choices of potential functions. We refer readers to Singhal et al. (2025) for a detailed discussion of different constructions.

We now consider estimating the expectation of a test function δ under the marginal $p_{\text{tilt}}(x_t)$, i.e., $\mathbb{E}_{p_{\text{tilt}}(x_t)}[\delta(x_t)]$. When $\delta(\cdot)$ is chosen as the Dirac delta function, this estimation reduces to constructing an empirical approximation of the marginal distribution $p_{\text{tilt}}(x_t)$ itself. To estimate $\mathbb{E}_{p_{\text{tilt}}(x_t)}[\delta(x_t)]$, we employ importance sampling with proposal distribution p_θ. This yields

$$\mathbb{E}_{p_{\text{tilt}}(x_t)}[\delta(x_t)] = \mathbb{E}_{p_\theta(x_{t:T})}\left[\frac{p_{\text{tilt}}(x_{t:T})}{p_\theta(x_{t:T})}\delta(x_t)\right] \approx \sum_{i=1}^{K} w^i_{t:T}\delta(x_t), \text{ where } w^i_{t:T} = \frac{p_{\text{tilt}}(x^i_{t:T})}{p_\theta(x^i_{t:T})}, x^i_{t:T} \sim p_\theta.$$

To learn an optimal proposal distribution, we minimise the log-variance of the importance weights, following the approach in Section 3. Using the identity $\mathbb{V}_h[x] = \mathbb{E}_h[(x - \mathbb{E}_h[x])^2]$ and defining $\overline{\log w_{t:T}} = \mathbb{E}_h[\log w_{t:T}]$, we obtain

$$\begin{aligned}
\mathbb{V}_h(\log w_{0:T}) &= \mathbb{E}_h\left[\left|\sum_t \log \frac{p_{\text{ref}}(x_t|x_{t+1})U(x_{t:T})}{p_\theta(x_{t-1}|x_t)} - \overline{\log w_{t:T}}\right|^2\right] \\
&= T^2\mathbb{E}_h\left[\left|\sum_t \frac{1}{T}\log \frac{p_{\text{ref}}(x_t|x_{t+1})U(x_{t:T})}{p_\theta(x_{t-1}|x_t)} - \frac{1}{T}\overline{\log w_{t:T}}\right|^2\right] \\
&\leq T^2\mathbb{E}_h\left[\sum_t \frac{1}{T}\left|\log \frac{p_{\text{ref}}(x_t|x_{t+1})U(x_{t:T})}{p_\theta(x_{t-1}|x_t)} - \overline{\log w_{t:T}}\right|^2\right] \\
&= T^2\mathbb{E}_{h,t}\left[\left|\log \frac{p_{\text{ref}}(x_t|x_{t+1})U(x_{t:T})}{p_\theta(x_{t-1}|x_t)} - \overline{\log w_{t:T}}\right|^2\right] \\
&= T\sum_t \mathbb{V}_h(\log w_{t:T}),
\end{aligned} \tag{37}$$

which establishes the bound $\mathbb{V}_h(\log w_{0:T}) \leq T\sum_t \mathbb{V}_h(\log w_{t:T})$. By choosing $U(x_{t:T}) = \exp(r(x_t)/\beta)$, we recover the result in Equation (34).

This inequality shows that minimising the variance of local importance weights effectively controls the variance of the global importance weight associated with the full trajectory. This property is particularly attractive, as it avoids backpropagating gradients through the entire trajectory; instead, optimisation can be performed using local, per-timestep objectives, leading to more stable training and improved scalability to long horizons.

C.2 VMPO KALEIDOSCOPES

In this section, we propose different design choices for the potential functions $U(x_{t:T})$ used in VMPO. While all valid potentials must satisfy the global constraint that their product recovers the desired exponential tilting of the terminal reward, this constraint alone does not uniquely specify U. We present several representative choices that offer complementary perspectives on how reward information can be propagated across timesteps.

Option 1: Return-to-Go. A natural choice of potential is the return-to-go, $U(x_{t:T}) = \exp\left(\frac{1}{T\beta}r(x_0)\right)$, which satisfies $\prod_t U(x_{t:T}) = \exp\left(\frac{1}{\beta}r(x_0)\right)$. Under this choice, the importance weight for a full trajectory simplifies to

$$w_{0:T} = \frac{p_{\text{ref}}(x_{0:T})\exp\left(\frac{1}{\beta}x_0\right)}{p_\theta(x_{0:T})} = \frac{\prod_t p_{\text{ref}}(x_{t-1}|x_t)\exp\left(\frac{1}{\beta}x_0\right)}{\prod_t p_\theta(x_{t-1}|x_t)}. \tag{38}$$

Applying the variance bound derived in Equation (37), we obtain

$$\mathbb{V}_h\left(\log \frac{p_{\text{ref}}(x_{0:T})\exp\left(\frac{1}{\beta}x_0\right)}{p_\theta(x_{0:T})}\right) \leq T\sum_t \mathbb{V}_h\left(\log \frac{p_{\text{ref}}(x_{t-1}|x_t)\exp\left(\frac{1}{T\beta}r(x_0)\right)}{p_\theta(x_{t-1}|x_t)}\right), \tag{39}$$

where minimising the right-hand side is equivalent to minimising the objective $\sum_t \mathcal{L}_h^{\text{Var}}(t;\theta)$ defined in Equation (4).[5]

Option 2: Difference. Another widely used choice of potential in SMC is the difference-based construction: $U(x_{t:T}) = \exp\left(\frac{1}{\beta}(r(x_t) - r(x_{t+1}))\right)$ and $U(x_T) = 1$, which satisfies the telescoping constraint $\prod_t U(x_{t:T}) = \exp\left(\frac{1}{\beta}r(x_0)\right)$. Under this choice, the importance weight of a full trajectory takes the form

$$w_{0:T} = \prod_t \frac{\exp(r(x_{t-1})/\beta)}{\exp(r(x_t)/\beta)}\frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)}. \tag{40}$$

Applying the variance bound in Equation (37), we obtain

$$\mathbb{V}_h\left(\sum_t \log \frac{\exp(r(x_{t-1})/\beta)}{\exp(r(x_t)/\beta)}\frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)}\right) \leq T\sum_t \mathbb{V}_h\left(\log \frac{\exp(r(x_{t-1})/\beta)}{\exp(r(x_t)/\beta)}\frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)}\right). \tag{41}$$

Minimising the right-hand side yields an alternative diffusion alignment objective,

$$\operatorname*{argmin}_\theta \sum_t \mathbb{V}_h\left(\log \frac{\exp(r(x_{t-1})/\beta)}{\exp(r(x_t)/\beta)}\frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)}\right). \tag{42}$$

which leads to the following practical loss:

$$\operatorname*{argmin}_{\theta,\phi} \mathbb{E}_{t,h}\left|\log \frac{\exp(r(x_{t-1})/\beta)}{\exp(r(x_t)/\beta)}\frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)} - M_\phi(t)\right|^2. \tag{43}$$

Notably, this objective is equivalent to the diffusion alignment formulation proposed in Ou et al. (2025a), which focuses on discrete diffusion models. In practice, rewards are often only defined on clean data x_0. In such cases, following Wu et al. (2023a), intermediate rewards can be constructed as

$$r(x_t) = \mathbb{E}_{p_\theta(x_0|x_t)}[r(x_0)] \approx r(\hat{x}_\theta(x_t, t)). \tag{44}$$

More generally, intermediate rewards can be parameterised using a neural network $F_\phi : \mathcal{X} \to \mathbb{R}$:

$$r_\phi(x_t) = F_\phi(x_t)r(\hat{x}_\theta(x_t, t)), \quad F_\phi(x_0) = 1 \tag{45}$$

where the constraint $F_\phi(x_0) = 1$ ensures satisfaction of the marginal constraint in Equation (36). This parameterisation is closely related to the forward-looking reward shaping used in GFlowNets (Pan et al., 2023).

Remark. The return-to-go and difference potentials provide two valid decompositions of the same globally tilted target distribution. The return-to-go formulation assigns the terminal reward uniformly across timesteps, resulting in a simple objective that does not require intermediate rewards, but offers limited temporal credit assignment. In contrast, the difference potential yields a telescoping decomposition that attributes reward changes to individual transitions, enabling more localised credit assignment. This can provide a more targeted optimisation signal when reliable intermediate rewards are available, but relies on their accurate estimation in diffusion models.

[5] In Equation (4), the factor $\frac{1}{T}$ is omitted from the reward term, as it can be absorbed into the temperature parameter β.

C.3 Connection to Existing Work

We then connect the proposed VMPO framework to existing works, demonstrating that a wide range of prior methods can be unified under VMPO through different choices of variance minimisation.

GVPO (Zhang et al., 2025a). We first consider Group Variance Policy Optimisation (GVPO) (Zhang et al., 2025a), which is designed to maximise the reward of a large language model (LLM) $p_\theta(x|c)$, where x denotes the generated response and c the prompt. GVPO optimises the following objective (Zhang et al., 2025a, Equantion 8)

$$\operatorname*{argmax}_\theta \sum_{i=1}^{K} \frac{p_\theta(x^i|c)}{p_{\theta_{\text{old}}}(x^i|c)} A^i, A^i = \frac{1}{\beta} r(x^i|c) + \log \frac{p_{\theta_{\text{sg}}}(x^i|c)}{p_{\theta_{\text{old}}}(x^i|c)} - \frac{1}{\beta}\overline{r(x|c)} - \overline{\log \frac{p_{\theta_{\text{sg}}}(x|c)}{p_{\theta_{\text{old}}}(x|c)}}, \quad (46)$$

where $\overline{r(x|c)} = \frac{1}{K}\sum_{i=1}^{K} r(x^i|c)$ and $\overline{\log \frac{p_{\theta_{\text{sg}}}(x|c)}{p_{\theta_{\text{old}}}(x|c)}} = \frac{1}{K}\sum_{i=1}^{K} \log \frac{p_{\theta_{\text{sg}}}(x^i|c)}{p_{\theta_{\text{old}}}(x^i|c)}$. It can be seen that the objective in Equation (46) closely resembles the VMPO formulation in Equation (32), differing primarily in the choice of model class and the tractability of likelihood evaluations. While GVPO has demonstrated strong empirical performance for LLM alignment, it does not directly extend to diffusion models, as the likelihood $p_\theta(x|c)$ is intractable in diffusion models but tractable for autoregressive LLMs. To address this limitation, we derive an upper bound on the log-variance of the importance weights, which yields a tractable VMPO objective suitable for diffusion alignment, as formalised in Equation (39).

FlowRL (Zhu et al., 2025). FlowRL is an alternative approach for reward optimisation in LLMs. Specifically, it train the model $p_\theta(x|c)$ using the following objective (Zhu et al., 2025, Equation 6)

$$\operatorname*{argmin}_{\theta,\phi} w \cdot \left(\log Z_\phi(x) + \frac{1}{|x|}\log p_\theta(x|c) - \frac{1}{|x|}\log p_{\text{ref}}(x|c) - r(x|c)\right)^2, \quad (47)$$

where $w = \text{clip}\left(\frac{p_{\theta_{\text{sg}}}(x|c)}{p_{\theta_{\text{old}}}(x|c)}, 1-\epsilon, 1+\epsilon\right)$ denotes the clipped importance weight, and Z_ϕ the learnable partition function. This objective can be interpreted as a VMPO variant in which a neural estimator is used to approximate the mean of the log importance weight, as in Equation (8). As with GVPO, FlowRL relies on explicit likelihood evaluations and is therefore not directly applicable to diffusion models. VMPO circumvents this limitation by operating on variance bounds of importance weights at each time step, enabling diffusion alignment without requiring tractable likelihoods of the clean data.

GFlowNet (Zhang et al., 2024). Under the difference potential, VMPO induces the per-timestep importance weight as in Equation (42)

$$w_t = \frac{\exp(r(x_{t-1})/\beta)}{\exp(r(x_t)/\beta)} \frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)}. \quad (48)$$

Beyond directly minimising the variance of importance weights, an alternative way to control the variance of these weights is to directly enforce consistency, via mean square errors, between the numerator and denominator at each timestep, yielding the objective

$$\operatorname*{argmin}_\theta \sum_t \left(\log \frac{\exp(r(x_{t-1})/\beta)}{\exp(r(x_t)/\beta)} \frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)}\right)^2, \quad (49)$$

whose minimum of is attained when $w_t = 1$ and thus $\mathbb{V}(\log w_t) = 0$. The objective in Equation (49) closely resembles the detailed-balance objectives used in GFlowNets (Zhang et al., 2024), with the main difference arising from how intermediate rewards are parameterised. In Zhang et al. (2024), the intermediate reward is modelled via a forward-looking estimator $\tilde{r}_\phi(x_t) = F_\phi(x_t) r(\hat{x}_\theta(x_t, t))$, subject to the terminal constraint $\tilde{r}_\phi(x_0) = r(x_0)$. This yields the following loss:

$$\operatorname*{argmin}_{\theta,\phi} \sum_t \left(\log \frac{\exp(\tilde{r}_\phi(x_{t-1})/\beta)}{\exp(\tilde{r}_\phi(x_t)/\beta)} \frac{p_{\text{ref}}(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)}\right)^2. \quad (50)$$

Due to the terminal constraint, the optimal policy still satisfies $p_{\theta^*} \propto p_{\text{ref}}(x_{t-1}|x_t)\exp(r(x_t)/\beta)$, consistent with the analysis of the difference potential in Appendix C.2. Notably, Zhang et al. (2024)

replace p_{ref} with the forward noising kernel $q(x_t|x_{t+1})$ in Equation (50), which leads to the optimal policy $p_{\theta^*} \propto \exp(r(x_t)/\beta)$. We refer to Liu et al. (2024) for further details.

∇-GFlowNet (Liu et al., 2024). Beyond the above methods, an alternative route to zero variance is to enforce the log importance weight to be constant. A sufficient condition for this is the vanishing of its gradient, since

$$\|\nabla_x \log w_t\|_2^2 = 0 \;\Rightarrow\; \log w_t = const. \;\Rightarrow\; \mathbb{V}(\log w_t) = 0. \tag{51}$$

This observation motivates learning the policy by penalising the gradient norm of the log importance weight. Under the difference potential, this leads to the following objective:

$$\underset{\theta}{\operatorname{argmin}} \sum_t \left\| \frac{1}{\beta} \nabla_{x_{t-1}} r(x_{t-1}) + \nabla_{x_{t-1}} \log p_{\text{ref}}(x_{t-1}|x_t) - \nabla_{x_{t-1}} \log p_\theta(x_{t-1}|x_t) \right\|_2^2. \tag{52}$$

This objective can be interpreted as minimising the Fisher divergence between the learned policy $p_\theta(x_{t-1}|x_t)$ and the locally tilted target $p_{\text{tilt}}(x_{t-1}|x_t) \propto p_{\text{ref}}(x_{t-1}|x_t)\exp(r(x_{t-1}/\beta))$. An equivalent formulation can be obtained by differentiating with respect to x_t, yielding

$$\underset{\theta}{\operatorname{argmin}} \sum_t \left\| \nabla_{x_t} \log p_{\text{ref}}(x_{t-1}|x_t) - \nabla_{x_t} \log p_\theta(x_{t-1}|x_t) - \frac{1}{\beta} \nabla_{x_t} r(x_t) \right\|_2^2. \tag{53}$$

In practice, the policy p_θ may be trained using either of these objectives, or a combination of both. These gradient-based objectives closely resemble those proposed in ∇-GFlowNet (Liu et al., 2024), differing mainly in the parametrisation of intermediate rewards. In Liu et al. (2024), the intermediate reward is modelled via a forward-looking parametrisation with a terminal constraint, as in Equation (50). Consequently, the reward gradient takes the form

$$\nabla_{x_t} \tilde{r}_\phi(x_t) = \nabla_{x_t} r(\hat{x}_\theta(x_t, t)) + \beta F_\phi(x_t), \qquad F_\phi(x_0) = 0. \tag{54}$$

Substituting this parametrisation into the gradient-matching objectives yields

$$\begin{aligned} \underset{\theta,\phi}{\operatorname{argmin}} \sum_t & \left\| \frac{1}{\beta} \nabla_{x_{t-1}} r(\hat{x}_\theta(x_{t-1}, t-1)) + \nabla_{x_{t-1}} F_\phi(x_{t-1}) + \nabla_{x_{t-1}} \log p_{\text{ref}}(x_{t-1}|x_t) - \nabla_{x_{t-1}} \log p_\theta(x_{t-1}|x_t) \right\|_2^2 \\ & + \left\| \nabla_{x_t} \log p_{\text{ref}}(x_{t-1}|x_t) - \nabla_{x_t} \log p_\theta(x_{t-1}|x_t) - \frac{1}{\beta} \nabla_{x_t} r(\hat{x}_\theta(x_t, t)) + \nabla_{x_t} F_\phi(x_t) \right\|_2^2, \end{aligned} \tag{55}$$

which recovers the ∇-GFlowNet objective defined in (Liu et al., 2024, Equation 19).

D Additional Experimental Details and Results

D.1 Experimental Details

Settings for SD1.5. We finetune SD1.5 (Rombach et al., 2022) using HPSv2 (Wu et al., 2023b) and ImageReward (Xu et al., 2023). For HPSv2, we adopt the photo and painting prompts from Wu et al. (2023b) as training data, while for ImageReward we use the DrawBench prompt set (Saharia et al., 2022). For evaluation, we follow Domingo-Enrich et al. (2024) and use their collected prompt set as test prompts. Both training setups largely share the same hyperparameter configuration, following the DDPO (Black et al., 2023) implementation; full details are provided here for completeness.

We use LoRA (Hu et al., 2022) with $\alpha = 4$ and $r = 4$. Trainings are performed on 2 NVIDIA A100 80GB GPUs with a per-GPU batch size of 8. With 4-step gradient accumulation, this yields an effective batch size of 64. We train for 150 epochs, where each epoch consists of sampling 128 trajectories from $p_{\theta_{\text{old}}}$ using a 50-step DDIM schedule (Song et al., 2020a) during the rollout phase, and performing 2 optimisation steps. The learning rate is fixed at 3×10^{-4} for both the diffusion model p_θ and the mean estimator M_ϕ. We employ the AdamW optimiser (Loshchilov & Hutter, 2017) with gradient clipping at a norm of 1.

During training, we adopt classifier-free guidance (Ho & Salimans, 2022) with a guidance scale of 5. Reward rescaling is employed to improve training efficiency. Specifically, we set $\beta = 0.01$ for HPSv2 and $\beta = 0.1$ for ImageReward to compute the loss in Equation (8). We also incorporate a KL

Table 2: Results on Stable Diffusion v1.5 fine-tuned with ImageReward.

Method	ImageReward (↑)	CLIPScore (↑)	HPSv2 (↑)	DreamSim (↑)
SD1.5 (Base)	0.0331 ± 0.0779	0.2717 ± 0.0032	0.2368 ± 0.0029	0.4389 ± 0.0116
GRPO	0.2490 ± 0.0792	0.2720 ± 0.0030	0.2420 ± 0.0032	0.3755 ± 0.0117
VMPO-R2G	0.1698 ± 0.0758	0.2721 ± 0.0032	0.2423 ± 0.0030	0.4008 ± 0.0117
VMPO-Diff	0.4716 ± 0.0740	0.2624 ± 0.0030	0.2645 ± 0.0031	0.3005 ± 0.0117

regularisation $\mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\theta_{\text{old}}}(x_{t-1}|x_t))$ with a coefficient of 1 to enhance training stability, consistent with Fan et al. (2023). For the GRPO baseline, we adopt the implementation[6] from Liu et al. (2025a) with a group size of 8, yielding the same effective number of samples as in our setup. All other hyperparameters are kept identical to those used for VMPO, as described above, to ensure a fair comparison.

Settings for SD3.5-M. We consider the text rendering task following Liu et al. (2025a), which fine-tunes SD3.5-M to improve the visual text rendering capability of diffusion models. The model is trained using the training prompt set provided in Liu et al. (2025a), and OCR accuracy is evaluated on their corresponding test prompt set. To assess potential reward hacking beyond task-specific accuracy, we additionally evaluate five complementary metrics on the DrawBench prompt set (Saharia et al., 2022), following Liu et al. (2025a): Aesthetic Score[7], DeQA (You et al., 2025), ImageReward (Xu et al., 2023), PickScore (Kirstain et al., 2023), and UnifiedReward (Wang et al., 2025b).

Following Liu et al. (2025a), we use LoRA with $\alpha = 64$ and $r = 32$. The model is trained on 4 NVIDIA A100 80 GPUs with a batch size of 4. We apply 4-step gradient accumulations, yielding an effective batch size of 64. For each training epoch, we sample 128 trajectories from $p_{\theta_{\text{old}}}$ using 10 steps and perform 2 optimisation steps. While at test time, the number of sampling steps is set to 40, following the setting in Liu et al. (2025a). We employ the AdamW optimiser (Loshchilov & Hutter, 2017) with the learning rate fixed as 3×10^{-4} for both p_θ and M_ϕ. Moreover, we use bf16 for p_θ while float32 for M_ϕ, which is important to stabilise training. During training, we adopt classifier-free guidance with a guidance scale of 4.5 and incorporate a KL regularisation $\mathbb{KL}(p_\theta(x_{t-1}|x_t)||p_{\theta_{\text{old}}}(x_{t-1}|x_t))$ with a coefficient of 1 to enhance training stability. We do not train GRPO ourselves; instead, we evaluate the released checkpoint[8] provided by Liu et al. (2025a). Notably, this checkpoint was trained with a larger batch size and on more GPUs than our setup.

D.2 Additional Experimental Results

Aligning SD1.5 with ImageReward. We further evaluate the performance of VMPO by fine-tuning SD1.5 using ImageReward. As illustrated by the convergence curves in Figure 4, VMPO-Diff demonstrates a significantly higher reward ceiling and faster optimisation compared to both GRPO and VMPO-R2G. This is quantitatively confirmed in Table 2, where VMPO-Diff achieves a superior ImageReward score 0.4716 ± 0.0740, more than doubling the performance of the GRPO baseline. While all methods show substantial gains over the base SD1.5 model in the target reward, we observe a similar trade-off as seen in the HPSv2 experiments: the increase in ImageReward is accompanied by a decrease in DreamSim scores. This suggests that while the model is successfully aligning with human preferences represented by ImageReward, it does so at the cost of visual diversity, a hallmark of reward hacking.

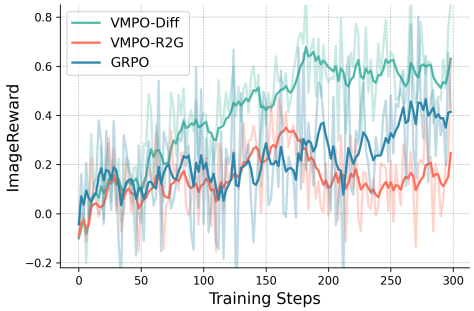

Figure 4: ImageReward convergence curves of SD1.5.

Improving Visual Text Rendering of SD3.5-M. Given the superior performance of VMPO-Diff in aligning SD1.5, we further evaluate its scalability by optimising the visual text rendering capabilities

[6] https://github.com/yifan123/flow_grpo
[7] https://laion.ai/blog/laion-aesthetics/
[8] https://huggingface.co/jieliu/SD3.5M-FlowGRPO-Text

Table 3: Visual text rendering results on Stable Diffusion v3.5-M. trained with OCR Accuracy.

Model	Task Metric	Image Quality		Preference Score		
	OCR Acc.	Aesthetic	DeQA	ImgRwd	PickScore	UniRwd
SD3.5-M	0.59	5.38	4.08	0.84	22.42	3.08
Flow-GRPO	0.92	5.32	4.10	0.95	22.50	3.12
VMPO-Diff	0.91	5.25	4.07	0.94	22.43	3.08

of the larger SD3.5-M model using OCR accuracy as the reward signal. As shown in Figure 5, VMPO consistently improves OCR accuracy throughout the training process, ultimately achieving a score of 0.91 (see Table 3). This represents a significant leap over the base SD3.5-M model's score of 0.59, demonstrating the effectiveness of our alignment method even in large-scale models. Notably, VMPO achieves performance comparable to Flow-GRPO (0.92), despite Flow-GRPO requiring significantly larger batch sizes and more extensive GPU resources for training. Qualitative results in Figure 7 further validate these findings: while the base model often struggles with legibility and character consistency, VMPO generates images with sharp, accurate, and well-aligned text. Despite a marginal decrease in image quality compared to the base model, the drastic improvement in task-specific accuracy highlights VMPO as a highly effective alternative for specialised alignment tasks. A large body of work aims to reduce the number of sampling steps required by diffusion models, enabling fast or even single-step generation.

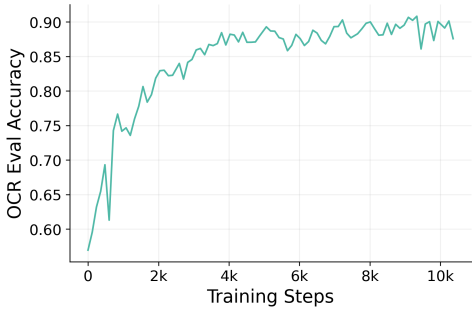

Figure 5: OCR accuracy convergence curves of SD3.5-M.

E Related Work

Diffusion and Flow Models. Diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020b) and flow models (Liu et al., 2022; Lipman et al., 2022; Albergo et al., 2023) have achieved remarkable success in modelling complex data across various domains, including image synthesis (Rombach et al., 2022), 3D generation (Poole et al., 2022), video synthesis (Ho et al., 2022), and audio generation (Liu et al., 2023). Beyond improving sample quality, recent research has increasingly focused on generation efficiency. A large body of work aims to reduce the number of sampling steps required by diffusion models, enabling fast or even single-step generation. Representative approaches include accelerated solvers (Song et al., 2020a; Lu et al., 2022), more expressive posterior parametrisation (Xiao et al., 2021; Ou et al., 2024), consistency and distillation-based methods (Salimans & Ho, 2022; Luo et al., 2023; Zhou et al., 2024; Zhang et al., 2025c; Wang et al., 2025a; Chen et al., 2025b). In parallel, aligning diffusion models with human preferences or task-specific objectives has emerged as an important research direction. Early work focuses on classifier guidance (Dhariwal & Nichol, 2021) and classifier-free guidance (Ho & Salimans, 2022), which steer generation using auxiliary models at inference time. Additionately, alternative methods formulate alignment as a probability inference problem, casting alignment as sampling from a reward-tilted distribution (He et al., 2025b; Zhang et al., 2025b). In our paper, we consider more recent methods that directly incorporate preference learning into training.

Reward Finetuning for Diffusion Alignment. Beyond the guidance methods, recent efforts on diffusion alignment focus on finetuning pretrained models to maximise expected reward. One line of research directly performs optimisation by backpropagating through the entire sampling trajectory (Clark et al., 2023; Prabhudesai et al., 2023), which requires differentiable reward signals. Another class of methods reframes alignment as a likelihood-based objective, applying maximum likelihood estimation to model-generated samples that are reweighted according to their rewards (Lee et al., 2023; Dong et al., 2023). Preference-based learning has also been adapted to diffusion models by extending direct preference optimisation with paired human preferences data (Rafailov et al., 2023). However, unlike autoregressive models with tractable likelihoods, diffusion models require additional approximations for likelihood evaluation and loss approximations (Wallace et al., 2024;

Liang et al., 2024; Yuan et al., 2024). A separate strand of work interprets the diffusion sampling procedure as a Markov decision process and applies reinforcement learning techniques to perform alignment (Black et al., 2023; Fan et al., 2023; Liu et al., 2025a; Xue et al., 2025; Li et al., 2025). Beyond standard RL formulations, alternative perspectives have been explored, including stochastic optimal control (Uehara et al., 2024; Domingo-Enrich et al., 2024; Potaptchik et al., 2026; 2025) and GFlowNet-based objectives (Zhang et al., 2024; Liu et al., 2024). More recently, DiffusionNFT (Zheng et al., 2025) demonstrates strong sample efficiency by eliminating the need for likelihood estimation and SDE-based reverse process. In contrast to these approaches, the proposed VMPO method offers a fundamentally different viewpoint on diffusion alignment, grounding optimisation in Sequential Monte Carlo and explicitly minimising the variance of importance weights to better align the model with the reward function.

Sequential Monte Carlo for Generative Modelling. SMC (Del Moral et al., 2006) constitutes a well-established class of probabilistic inference techniques, providing adaptable and efficient sampling strategies across a variety of settings, including Bayesian experimental design (Ryan et al., 2016) and particle filtering (Johansen, 2009). More recently, SMC has been combined with diffusion-based generative models (Chen et al., 2025a; Ou et al., 2025b; He et al., 2025a; Skreta et al., 2025; Wu et al., 2025b), resulting in neural sampling frameworks that enable effective approximation of complex Boltzmann distributions. These hybrid approaches have been successfully applied to posterior inference (Dou & Song, 2024; Cardoso et al., 2023) as well as reward-guided alignment (Wu et al., 2023a; Singhal et al., 2025), all without requiring additional model training. Beyond diffusion models, SMC has also been explored as a tool for improving large language models. In particular, Zhao et al. (2024) propose SMC as a principled inference framework to mitigate capability and safety challenges in LLMs, a direction that has since been extended to test-time scaling for mathematical reasoning (Feng et al., 2024; Puri et al., 2025). Our work is most closely related to recent efforts that learn optimal proposal distributions for SMC in discrete diffusion models (Ou et al., 2025a). While it also adopts variance minimisation as a learning objective, VMPO offers a more comprehensive study and understanding of variance minimisation, revealing connections to various previous alignment methods through different choices of potential functions and variance minimisation objectives.

F Limitation and Future Work

As established in Appendix C, VMPO provides a holistic framework that unifies various diffusion alignment methods through different potential functions and variance minimisation objectives. This discloses several avenues for further exploration. One limitation of this paper is that we do not exhaustively discuss which specific variant serves as the optimal choice across different scenarios. While VMPO-Diff demonstrated strong empirical results, the design space for potentials is vast, and future research could systematically benchmark these variants against a broader range of baselines to identify their respective strengths in different alignment scenarios.

Furthermore, our empirical evaluation primarily focuses on text-to-image models like Stable Diffusion v1.5 and v3.5-M. Expanding this framework to other modalities, such as video generation (Kong et al., 2024; Wan et al., 2025), and to more recently advanced models, like Qwen-Image (Wu et al., 2025a), Z-image (Cai et al., 2025), presents a promising direction for future work. Additionally, our results indicate that VMPO, like other alignment methods, remains susceptible to reward hacking, as gains in preference scores often coincide with a reduction in visual diversity. Future research could investigate more robust regularisation strategies beyond standard KL penalties to better preserve the original data distribution while achieving effective reward alignment. Finally, applying VMPO to align one-step generative models can be an interesting direction for future research.

Figure 6: Illustration of the generated samples of different models.

Figure 7: Illustration of the generated samples of different models in text rendering.