# A Implementation Details

## A.1 Proximal Policy Optimization (PPO) Algorithm

Proximal Policy Optimization (PPO) algorithm offers a balance between sample efficiency and ease of implementation. In this section, we elaborate in more detail of the PPO algorithm adopted in our paper. First, consider a surrogate objective:

$$L^{\mathrm{CPI}}(\boldsymbol{\phi}) = \mathbb{E}_t \left[ \frac{\boldsymbol{\pi}_{\boldsymbol{\phi}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\boldsymbol{\pi}_{\boldsymbol{\phi}_{\mathrm{old}}}(\boldsymbol{a}_t|\boldsymbol{s}_t)} \hat{A}_t \right],$$

where $\boldsymbol{\pi}_{\boldsymbol{\phi}}$ and $\boldsymbol{\pi}_{\boldsymbol{\phi}_{\mathrm{old}}}$ are the policy network before and after the update, respectively. The advantage estimator $\hat{A}_t$ is computed by:

$$\hat{A}_t = -V(\boldsymbol{s}_t) + R(\boldsymbol{s}_T, \boldsymbol{a}_T),$$

where $V(\boldsymbol{s}_t)$ is a learned state-value function. This objective function effectively maximizes the probability ratio $\rho_t(\boldsymbol{\phi}) = \frac{\boldsymbol{\pi}_{\boldsymbol{\phi}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\boldsymbol{\pi}_{\boldsymbol{\phi}_{\mathrm{old}}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}$ when considering the advantage of taking action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$. However, directly maximizing $L^{\mathrm{CPI}}(\boldsymbol{\phi})$ usually leads to an excessively large policy update, hence, we consider how to modify the objective, to penalize changes to the policy that move $\rho_t(\boldsymbol{\phi})$ away from 1. This gives rise to the clipped surrogate objective:

$$L^{\mathrm{CLIP}}(\boldsymbol{\phi}) = \mathbb{E}_t \left[ \min \left( \rho_t(\boldsymbol{\phi})\hat{A}_t, \mathrm{clip}\left(\rho_t(\boldsymbol{\phi}), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right]$$

where $\epsilon$ is a hyper-parameter that controls the range of the probability ratio. Finally, to learn the state-value function, we additionally add a term for value function estimation error to the objective following [20], which results in the objective in Eq. 11 in our main paper.

## A.2 Architecture

Our policy network consists of a depth-wise convolution layer, a point-wise convolution layer and a multi-layer perceptron (MLP). We take the NAT model's output feature as the generation status input, and additionally use adaptive layernorm (AdaLN) [15, 16] layers to incorporate timestep information into the policy network. The architecture of the adversarial reward model follows the discriminator in StyleGAN-T [18]. Notably, the policy network is highly lightweight, incurring *negligible* additional inference cost:

| model | $T$ | infer. cost (all) | infer. cost (policy) | proportion (policy/all) |
|---|---|---|---|---|
| AdaNAT-S | 4 | 184.5 GFLOPs | 0.064 GFLOPs | 0.03% |

### A.3 Hyperparameter Details

We perform the optimization loop of AdaNAT in Algorithm 1 for 1000 iterations. In practice, we perform 5 gradient updates within each loop of Algorithm 1 for both the policy network and the adversarial reward model to facilitate a more stable and efficient optimization process in the minimax game. For the optimization of the policy network, we adopt $\epsilon = 0.2, c = 0.5$ for the PPO objective, and use Adam [9] optimizer with a learning rate of $1 \times 10^{-5}$, $\beta_1 = 0.9, \beta_2 = 0.999$. The batch size is set to 4096. The $\sigma$ hyperparameter in Eq. 9, which balances exploration and exploitation, is set to 0.6 and reduced to 0.3 after 500 iterations. For the adversarial reward model, we use Adam [9] optimizer with a learning rate of $1 \times 10^{-4}$ and $\beta_1 = 0.5, \beta_2 = 0.999$. The batch size of updating the adversarial reward model is set to 1024 by default. For experiments on ImageNet 512×512 [17] and CC3M [21], we reduce the batch size to 512 to fit the memory constraints. For the pre-training of our NAT models, we generally follow the training settings used in previous work [1], with modifications on learning rate to 4e-4 and a larger batch size of 2048 on ImageNet dataset. The results on CC3M is based on a publicly available Muse model from github[1].

## B Experiment Details of FID-based Reward

When implementing FID-based reward design as described in Section 4.3, we find that the FID-based reward model is not able to provide a stable and effective reward signal for the adaptive policy network, leading to divergence:

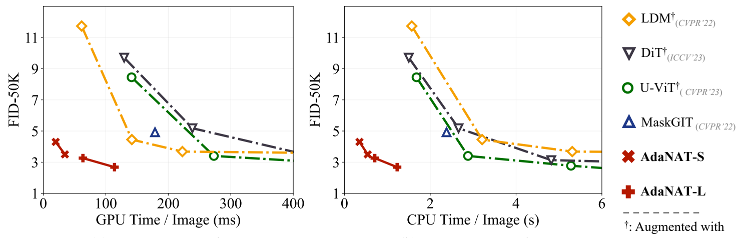
| dataset | model | $T$ | FID-50K↓ adaptive | FID-50K↓ non-adaptive |
|---|---|---|---|---|
| ImageNet 256 × 256 | AdaNAT-L | 8 | 55.4 (Fail) | **2.56** |

As a result, we adopted a non-adaptive version of policy network, where all samples share the same generation configuration. Empirically, the non-adaptive policy network can also be optimized effectively with a low FID. However, as discussed in Section 5.2, this numerical superiority does not translate to a practical advantage in terms of sample quality. The FID reward-based policy network fails to produce images of satisfactory quality.

## C Practical Latency

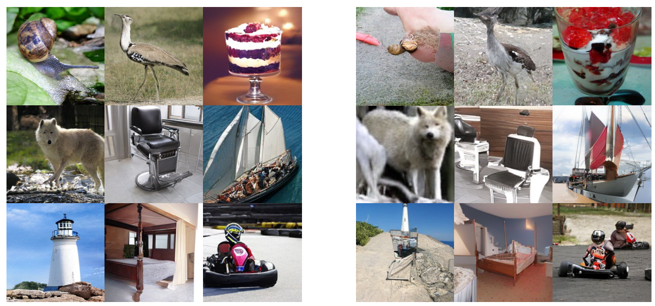
Figure 1 illustrates the comparison of the practical latency of AdaNAT against several competitive baselines on ImageNet 256×256. This comparison includes the latency on both GPU and CPU for generating a single image. The results present a more comprehensive comparison on the efficiency & efficacy tradeoff of AdaNAT and other methods in practical scenarios.

[1] Due to the concern that including the link runs a risk of violating anonymity, we will provide the specific reference in the camera-ready version of this paper.




(a) FID *vs.* GPU Time (b) FID *vs.* CPU Time

**Fig. 1: Practical latency of AdaNAT on ImageNet 256×256**. GPU time is measured on an A100 GPU with batch size 50. CPU time is measured on Xeon 8358 CPU with batch size 1. † : DPM-Solver [11] augmented diffusion models.



(a) AdaNAT (b) AutoNAT

**Fig. 2: Qualitative comparisons between AdaNAT and AutoNAT [14] on ImageNet 256×256**. AdaNAT generates images with superior visual quality.

## D Comparisons with AutoNAT

Similar to AdaNAT, AutoNAT [14] aims to enhance the policies in non-autoregressive Transformers. It achieves this by optimizing a FID-based objective. The table below provides quantitative comparisons between AdaNAT and AutoNAT on ImageNet 256×256 with $T = 8$.

| method | TFLOPs↓ | FID-50K↓ |
|---|---|---|
| AutoNAT-L [14] | 0.9 | 2.68 |
| AdaNAT-L (FID-based) | 0.9 | **2.56** |
| AdaNAT-L (Adv-based) | 0.9 | 2.86 |

The results demonstrate that the FID-based approach achieves better quantitative metrics, with our FID-based AdaNAT model outperforming AutoNAT. However, as discussed in Section 5.2, this optimization often leads to overfitting, resulting in subpar image quality. Consequently, we opted for an adversarial-based approach, which offers a more robust and favorable solution. Qualitative comparisons between AutoNAT and AdaNAT in Figure 2 illustrate that AdaNAT generates images with superior visual quality.

## E Potential Impact, Limitation, and Future Work

As with any AI-generated content technology, there are potential ethical considerations and risks of misuse, such as creating misleading content, deepfakes, or spreading misinformation. Additionally, like other data-driven approaches, the model may inadvertently reinforce biases present in the training data. In terms of limitations and future work, it is essential to investigate the efficacy of AdaNAT on larger-scale datasets like laion-5B [19] and explore the performance of NAT models exceeding 1B parameters to understand scalability and robustness. Additionally, applying AdaNAT across more diverse generative tasks and domains [4–7] could broaden its impact. Integrating more advanced adaptive inference methods [8,23–25,27–31] and learning techniques [7,13,22,26] can further enhance the capabilities and applicability of non-autoregressive Transformers. Finally, better interpreting the decisions made by the policy network and translating them into insights for designing improved non-autoregressive transformer generation paradigms presents a valuable direction for future research.

## F Scheduling Functions of Existing Works

The scheduling functions of existing works [2, 3, 10, 12] are shown in the table below:

| generation policy | scheduling functions |
|---|---|
| re-masking ratio $m^{(t)}$ | $m^{(t)} = \cos \frac{\pi(t+1)}{2T}$ |
| sampling temp. $\tau_1^{(t)}$ | ${\tau_1}^{(t)} = 1.0$ |
| re-masking temp. $\tau_2^{(t)}$ | ${\tau_2}^{(t)} = \frac{\lambda(T-t)}{T}$ |
| guidance scale $w^{(t)}$ | $w^{(t)} = \frac{k(t+1)}{T}$ |

## References


1. Bao, F., Li, C., Cao, Y., Zhu, J.: All are worth words: a vit backbone for score-based diffusion models. In: CVPR (2023)
2. Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.H., Murphy, K., Freeman, W.T., Rubinstein, M., et al.: Muse: Text-to-image generation via masked generative transformers. In: ICML (2023)
3. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: CVPR (2022)
4. Gal, R., Patashnik, O., Maron, H., Bermano, A.H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators (2022)
5. Guo, J., Wang, C., Wu, Y., Zhang, E., Wang, K., Xu, X., Shi, H., Huang, G., Song, S.: Zero-shot generative model adaptation via image-specific prompt learning. In: CVPR (2023)
6. Guo, J., Xu, X., Pu, Y., Ni, Z., Wang, C., Vasu, M., Song, S., Huang, G., Shi, H.: Smooth diffusion: Crafting smooth latent spaces in diffusion models. CVPR (2024)
7. Guo, J., Zhao, J., Ge, C., Du, C., Ni, Z., Song, S., Shi, H., Huang, G.: Everything to the synthetic: Diffusion-driven test-time adaptation via synthetic-domain alignment. arXiv preprint arXiv:2406.04295 (2024)

8. Huang, G., Wang, Y., Lv, K., Jiang, H., Huang, W., Qi, P., Song, S.: Glance and focus networks for dynamic visual recognition. TPAMI (2022)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
10. Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., Krishnan, D.: Mage: Masked generative encoder to unify representation learning and image synthesis. In: CVPR (2023)
11. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In: NeurIPS (2022)
12. Mentzer, F., Minnen, D., Agustsson, E., Tschannen, M.: Finite scalar quantization: Vq-vae made simple. In: ICLR (2023)
13. Ni, Z., Wang, Y., Yu, J., Jiang, H., Cao, Y., Huang, G.: Deep incubation: Training large models by divide-and-conquering. In: ICCV (2023)
14. Ni, Z., Wang, Y., Zhou, R., Guo, J., Hu, J., Liu, Z., Song, S., Yao, Y., Huang, G.: Revisiting non-autoregressive transformers for efficient image synthesis. In: CVPR (2024)
15. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
16. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: AAAI (2018)
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. In: IJCV (2015)
18. Sauer, A., Karras, T., Laine, S., Geiger, A., Aila, T.: Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In: International conference on machine learning. pp. 30105–30118. PMLR (2023)
19. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. In: NeurIPS (2022)
20. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
21. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018)
22. Tang, Y., Ni, Z., Zhou, J., Zhang, D., Lu, J., Wu, Y., Zhou, J.: Uncertainty-aware score distribution learning for action quality assessment. In: CVPR (2020)
23. Wang, Y., Chen, Z., Jiang, H., Song, S., Han, Y., Huang, G.: Adaptive focus for efficient video recognition. In: ICCV (2021)
24. Wang, Y., Huang, R., Song, S., Huang, Z., Huang, G.: Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition (2021)
25. Wang, Y., Lv, K., Huang, R., Song, S., Yang, L., Huang, G.: Glance and focus: a dynamic approach to reducing spatial redundancy in image classification. NeurIPS (2020)
26. Wang, Y., Ni, Z., Song, S., Yang, L., Huang, G.: Revisiting locally supervised learning: an alternative to end-to-end training. In: ICLR (2021)
27. Wang, Y., Yue, Y., Lin, Y., Jiang, H., Lai, Z., Kulikov, V., Orlov, N., Shi, H., Huang, G.: Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. In: CVPR (2022)
28. Wang, Y., Yue, Y., Xu, X., Hassani, A., Kulikov, V., Orlov, N., Song, S., Shi, H., Huang, G.: Adafocusv3: On unified spatial-temporal dynamic video recognition. In: ECCV (2022)

29. Yang, L., Han, Y., Chen, X., Song, S., Dai, J., Huang, G.: Resolution adaptive networks for efficient inference. In: CVPR (2020)
30. Yang, L., Jiang, H., Cai, R., Wang, Y., Song, S., Huang, G., Tian, Q.: Condensenet v2: Sparse feature reactivation for deep networks. In: CVPR (2021)
31. Zheng, Z., Yang, L., Wang, Y., Zhang, M., He, L., Huang, G., Li, F.: Dynamic spatial focus for efficient compressed video action recognition. TCSVT (2023)