A. Implementation Details

In this section, we present the benchmarks, evaluation metrics and the relevant contents we used in the main paper to facilitate a comprehensive understanding of our model's performance. This overview will help contextualize our results and provide clarity on how we assessed the effectiveness of our approach.

A.1. Benchmarks

In our main paper, we conduct experiments across three popular text-to-image datasets.

Pick-a-Pic. Pick-a-Pic [23] is an open dataset designed to collect user preferences for images synthesized from text prompts. The dataset is gathered through a user-friendly web application that allows users to synthesize images and select their preferences. Each data sample includes a text prompt, two synthesized images, and a label indicating which the user prefers or a tie if there is no clear preference. The Pick-a-Pic dataset contains over 500,000 examples covering 35,000 unique prompts. Its advantage lies in the fact that the data comes from real users, reflecting their genuine preferences rather than relying on paid crowd workers

DrawBench. DrawBench is a newly introduced benchmark dataset designed for in-depth evaluation of text-to-image synthesis models. It contains 200 carefully crafted prompts categorized into 11 groups, testing the models' abilities across various semantic attributes, including compositionality, quantity, spatial relationships, and handling complex text prompts. The design of DrawBench allows for a multidimensional assessment of model performance, helping researchers identify strengths and weaknesses in image synthesis. By comparing with other models, DrawBench provides a comprehensive evaluation tool for the text-to-image synthesis field, facilitating a deeper understanding of synthesis quality and image-text alignment.

HPD v2. The Human Preference Dataset v2 [53] is a large-scale, cleanly annotated dataset focused on user preferences for images synthesized from text prompts. It contains 798,090 binary preference choices involving 433,760 pairs of images, aiming to address the limitations of existing evaluation metrics that fail to accurately reflect human preferences. HPD v2 eliminates potential biases and provides a more comprehensive evaluation capability, with data sourced from multiple text-to-image synthesis models and real images.

GenEval. GenEval, an object-focused T2I benchmark to evaluate compositional image properties such as object co-occurrence, position, count, and color, contains 553 prompts in total. This dataset plays a pivotal role in advancing research in text-to-image synthesis by providing a structured means to assess how well images align with the descriptive content of accompanying texts.

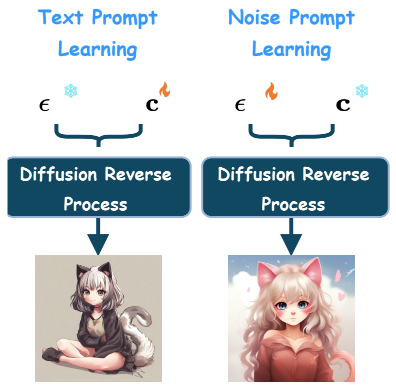


Figure 6. Paradigms of *text prompt learning* (left) and *noise prompt learning* (right).

T2I-CompBench. T2I-CompBench is a comprehensive benchmark designed for evaluating the compositional capabilities of text-to-image (T2I) models, focusing on aspects such as object arrangement, relationships, and attributes within synthesized images. This benchmark consists of a diverse set of prompts that encompass various scenarios and contexts, facilitating a thorough assessment of how well T2I systems can interpret and visualize complex textual descriptions. By providing a structured framework for evaluation, T2I-CompBench significantly contributes to the advancement of research in T2I synthesis, offering insights into the strengths and limitations of current models in generating coherent and contextually accurate images.

For testing, we use these four popular T2I datasets, including the first 100 prompts subset from the Pick-a-Pic web application, 100 prompts from HPD v2 test set, all 200 prompts from DrawBench, all 553 prompts from GenEval and all test prompts in T2I-CompBench. The detailed information of the test sets is shown in Fig. 18.

A.2. Evaluation Metrics

In our main paper, we mainly include 6 evaluation metrics to validate the effectiveness of our NPNet.

PickScore. PickScore is a CLIP-based scoring function trained from the Pick-a-Pic dataset, which collects user preferences for synthesized images. It achieves superhuman performance when predicting user preferences. PickScore aligns well with human judgments, and together with Pick-a-Pic's natural distribution prompts, enables much more relevant text-to-image model evaluation than evaluation standards, such as FID [13] over MS-COCO [27].

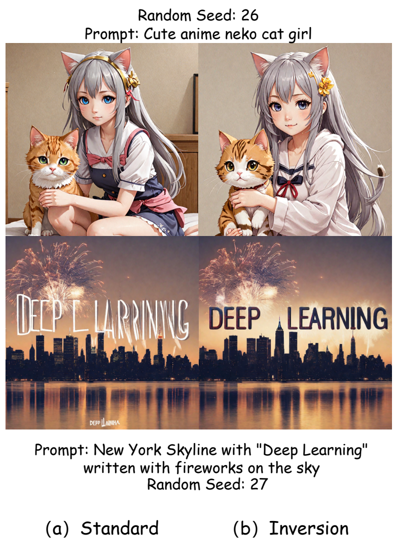

Random Seed: 26
Prompt: Cute anime neko cat girl
Prompt: New York Skyline with "Deep Learning" written with fireworks on the sky
Random Seed: 27
(a) Standard
(b) Inversion

Figure 7. Visualization results about *re-denoise sampling*. *Re-denoise sampling* can help to inject semantic information of the text prompt into the original Gaussian noise.

HPSv2. Human Preference Score v2 (HPSv2) is an advanced preference prediction model by fine-tuning CLIP [39] on Human Preference Dataset v2 (HPD v2). This model aims to align text-to-image synthesis with human preferences by predicting the likelihood of a synthesized image being preferred by users, making it a reliable tool for evaluating the performance of text-to-image models across diverse image distribution.

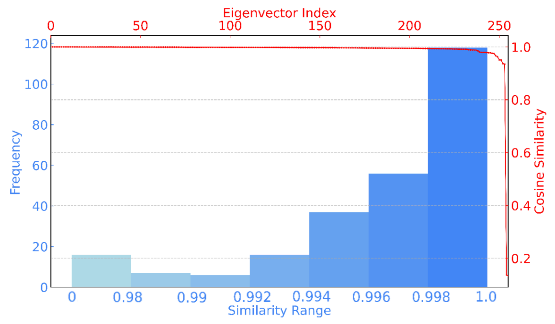


Figure 8. Visualization about the similarities between the singular vectors of $\mathbf{x}_T$ and $\mathbf{x}'_T$. Note that we take the absolute values of the cosine similarity scores, and sort them reversely (the horizontal axis represents the indexes of the singular vectors).

AES. Aesthetic Score (AES) [46] are derived from a model trained on the top of CLIP embeddings with several extra multilayer perceptron (MLP) layers to reflect the visual appeal of images. This metric can be used to evaluate the aesthetic quality of synthesized images, providing insights into how well they align with human aesthetic preferences.

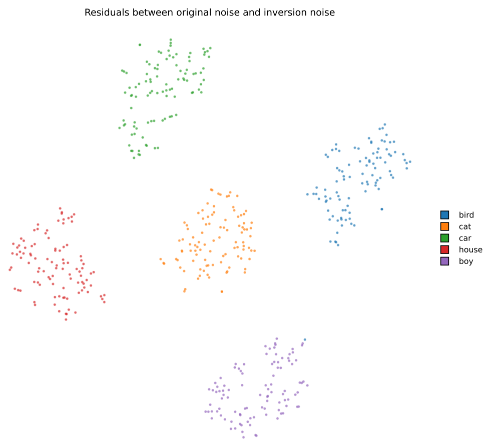


Figure 9. Using re-denoising sampling, we actively enhance the semantic information of the given prompt within the initial noise to enhance the semantic alignment of the generated images. This process is validated by the tight clustering of the residuals between the original noises and the inversion noises, each prompt with 100 noises, demonstrating the semantic information within the inversion noises.

ImageReward. ImageReward [55] is a human preference reward model specifically designed for evaluating text-to-image synthesis. It is trained on a large dataset of human comparisons, allowing it to effectively encode human preferences. The model assesses synthesized images based on various criteria, including alignment with the text prompt and overall aesthetic quality. ImageReward has been shown to outperform traditional metrics like Inception Score (IS) [3] and Fréchet Inception Distance (FID) in correlating with human judgments, making it a promising automatic evaluation metric for text-to-image synthesis.

CLIPScore. CLIPScore [12] leverages the capabilities of the CLIP model, which aligns images and text in a shared embedding space. By calculating the cosine similarity between the image and text embeddings, CLIPScore provides a mearsure of how well a synthesized image corresponds to its textual description. While CLIPScore is effective in assessing text-image alignment, it may not fully capture the nuances of human preferences, particularly in terms of aesthetic quality and detail[2].

MPS. Multi-dimensional Preference Score (MPS) [59], the first multi-dimensional preference scoring model for the evaluation of text-to-image models. The MPS intro-

[2]we follow https://github.com/jmhessel/clipscore

duces the preference condition module upon CLIP model to learn these diverse preferences. It is trained based on the Multi-dimensional Human Preference (MHP) Dataset, which comprises 918,315 human preference choices across four dimensions, including aesthetics, semantic alignment, detail quality and overall assessment on 607,541 images, providing a more comprehensive evaluation of synthesized images. MPS calculates the preference scores between two images, and the sum of the two preference scores equals 1.

A.3. T2I Diffusion Models

In the main paper, we totally use 3 T2I diffusion models, including StableDiffusion-xl (SDXL) [37], DreamShaper-xl-v2-turbo (DreamShaper), and Hunyuan-DiT (DiT) [25].

StableDiffusion-xl. StableDiffusion-xl (SDXL) is an advanced generative model, building upon the original Stable Diffusion architecture. This model leverages a three times larger UNet backbone, and utilizes a refinement model, which is used to improve the visual fidelity of samples synthesized by SDXL using a post-hoc image-to-image technique. SDXL is designed to synthesize high-resolution images from text prompts, demonstrating significant improvements in detail, coherence, and the ability to represent complex scenes compared to its predecessors.

DreamShaper-xl-v2-turbo. DreamShaper-xl-v2-turbo, a fine-tuned version on SDXL, is a text-to-image model designed for high-quality image synthesis, focusing on faster inference time and enhanced image synthesis capabilities. DreamShaper-xl-v2-turbo maintains the high-quality image output characteristic of its predecessor, while its turbo enhancement allows for quicker synthesis cycles. The overall style of the synthesized images leans towards fantasy, while it achieves a high level of authenticity when realism is required.

Hunyuan-DiT. Hunyuan-DiT is a text-to-image diffusion transformer with fine-grained understanding of both English and Chinese. With careful design of the model architecture, it can perform multi-turn multimodal dialogue with users to synthesized high-fidelity images, under the refinement of the Multimodal Large Language Model.

A.4. Model Architecture

Our NPNet consists of two branches, one is *singular value prediction*, and another is *residual prediction*.

For the *singular value prediction* branch, we processes the decomposed SVD components through three parallel MLP branches: one for the orthogonal matrix U, one for V, and another for singular values s. Each branch contains two linear layers with ReLU activation. We combine them with a self-attention module to model global relationships, followed by a residual MLP that adjusts the singular values while retaining their original characteristics through skip connections. The refined components are reconstructed via matrix operations to produce the output. The detail of the self-attention is illustrated in Fig. 11.

For the *residual prediction* branch, we employs a hybrid architecture combining a pretrained Swin-Tiny transformer with learnable up/down-sampling layers. The input is first upsampled to 224×224 resolution, projected to 3 channels via 1×1 convolution, then processed by the transformer's feature extractor. Features are downsampled to the original resolution and projected back through another 1×1 convolution. An optional residual connection adds the input to the refined output.

A.5. Hyper-parameter Settings

Our method is straightforward and intuitive, and the parameter settings for the entire experiment are also very simple, with epoch 30, and batch size 64 for all experiments. During training, we surprise the obverse that randomly assigning one prompt to the whole samples in one batch can yield a better model, where we speculate that such an operation could be considered as a strong regularizer as in Chen et al. [6]. We conduct experiments on three T2I diffusion models, including SDXL, DreamShaper-xl-v2-turbo and Hunyuan-DiT, with CFG ω_l 5.5, 3.5, and 5.0 respectively. The inverse CFG ω_w is 1.0 for all three models. To collect training data, the inference steps are 10, 4, and 10 for SDXL with DDIM inverse scheduler, Dreamshaper-xl-v2-turbo with DPMSolver inverse scheduler, and Hunyuan-DiT with DDIM inverse scheduler, respectively. The human preference model we use to filter the data is the HPSv2, and the filtering threshold k equals 0. Unless otherwise specified, all quantitative experiments and synthesized images in this paper are conducted and synthesize with inference steps 50, respectively. All experiments are conducted using $1\times$ RTX 4090 GPUs, and all these noise pairs are collected with inference step 10 to construct NPDs.

B. Related Work

Synthesizing images that are precisely aligned with given text prompts remains a significant challenge for Text-to-Image (T2I) diffusion models. To deal with this problem, several works explore training-free improvement strategies, by optimizing the noises during the diffusion reverse process.

Lugmayr et al. [32] utilizes a pre-trained unconditional diffusion model as a generative prior and alters the reverse diffusion iterations based on the unmasked regions of the input image. Hu et al. [19] proposes to denoise one more step before the standard denoising process to eliminate the SNR noise discrepancy between training and inference. Meng et al. [34] observe that denoising the noise with inversion

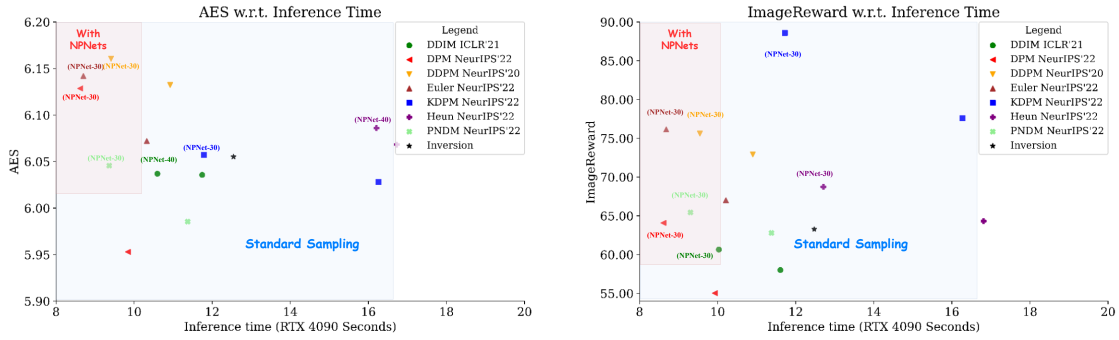


Figure 10. We evaluate our NPNet with 7 samplers on SDXL in Pick-a-Pic dataset, including both the deterministic sampler and stochastic sampler (with a default inference step 50). "NPNet-30" means the inference step is 30 with NPNet. The red area in the top left corner of the image represents the results of efficient high-performance methods, while the experimental results of NPNet are nearly in that same region. It highlights that NPNet is capable of synthesizing higher-quality images with fewer steps and consuming less time. Moreover, the results demonstrate the generalization ability of our NPNet across different samplers.

Table 6. Generalization on different diffusion models. We train our NPNet with NPD collected from SDXL. We apply it directly to DreamShaper-xl-v2-turbo on Pick-a-Pic dataset. Our results show promising performance, highlighting the model's capability for cross-model generalization.

Inference Steps		PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
4	Standard	21.57	29.02	5.9172	53.12
	NPNet (ours)	21.62	29.20	5.9159	58.46
10	Standard	22.39	32.16	6.0296	96.67
	NPNet (ours)	22.41	32.24	6.0320	97.92
30	Standard	22.42	32.33	6.0116	98.97
	NPNet (ours)	22.44	32.48	6.0054	100.18
50	Standard	22.41	32.12	6.0161	98.09
	NPNet (ours)	22.47	32.25	6.0033	99.86

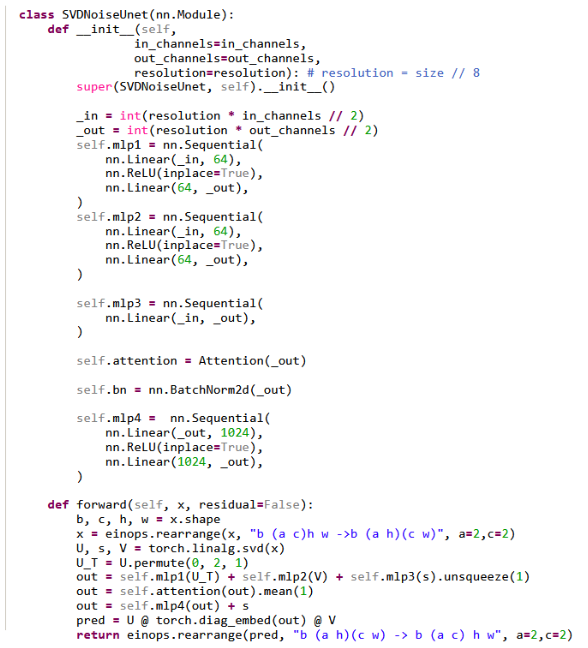
steps can generate better images compared with the original denoising process. Based on that, Qi et al. [38] aims to reduce the truncate errors during the denoising process, by increase the cosine similarity between the initial noise and the inversed noise in an end-to-end way. It introduces significant time costs, and the synthesized images may be over-rendered, making it difficult to use in practical scenarios.

Another research direction introduces extra modules to help optimize the noises during the reverse process. Chefer et al. [5] introduce the concept of Generative Semantic Nursing (GSN), and slightly shifts the noisy image at each timestep of the denoising process, where the semantic information from the text prompt is better considered. InitNO [11] consists of the initial latent space partioning and the noise optimization pipeline, responsible for defining valid regions and steering noise navigation, respectively. Such methods are not universally applicable, we discuss this in Appendix C

Unlike previous approaches, we are the first to reframe this task as a learning problem. we directly learn to prompt the initial noise into the winning ticket noise to address this issue, by training a universal Noise prompt network (NPNet) with our noise prompt dataset (NPD). Our NPNet operates as a plug-and-play module, with very limited memory cost and negligible inference time cost, produce images with higher preference scores and better alignment with the input text prompts effectively.

Table 7. Comparison results with InitNO on StableDiffusion-v1-4 [41] on Pick-a-Pic dataset. We directly apply NPNet trained for SDXL, and remove the embedding $\mathbf{e}$ to StableDiffusion-v1-4.

	PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
Standard	19.17	19.49	5.4575	-122.73
InitNO	16.50	14.47	5.3116	-205.66
NPNet (ours)	19.25	19.54	5.5151	-95.89



```
class SVDNoiseUnet(nn.Module):
    def __init__(self,
                in_channels=in_channels,
                out_channels=out_channels,
                resolution=resolution): # resolution = size // 8
        super(SVDNoiseUnet, self).__init__()

        _in = int(resolution * in_channels // 2)
        _out = int(resolution * out_channels // 2)
        self.mlp1 = nn.Sequential(
            nn.Linear(_in, 64),
            nn.ReLU(inplace=True),
            nn.Linear(64, _out),
        )
        self.mlp2 = nn.Sequential(
            nn.Linear(_in, 64),
            nn.ReLU(inplace=True),
            nn.Linear(64, _out),
        )

        self.mlp3 = nn.Sequential(
            nn.Linear(_in, _out),
        )

        self.attention = Attention(_out)

        self.bn = nn.BatchNorm2d(_out)

        self.mlp4 =  nn.Sequential(
            nn.Linear(_out, 1024),
            nn.ReLU(inplace=True),
            nn.Linear(1024, _out),
        )

    def forward(self, x, residual=False):
        b, c, h, w = x.shape
        x = einops.rearrange(x, "b (a c)h w ->b (a h)(c w)", a=2,c=2)
        U, s, V = torch.linalg.svd(x)
        U_T = U.permute(0, 2, 1)
        out = self.mlp1(U_T) + self.mlp2(V) + self.mlp3(s).unsqueeze(1)
        out = self.attention(out).mean(1)
        out = self.mlp4(out) + s
        pred = U @ torch.diag_embed(out) @ V
        return einops.rearrange(pred, "b (a h)(c w) -> b (a c) h w", a=2,c=2)
```

Figure 11. The implementation details of the self-attention module in *singular value prediction* branch.

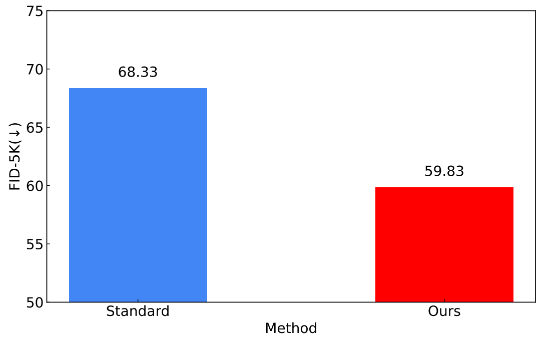


Figure 12. The FID comparison with 5000 images in class-conditional ImageNet with the resolution 512 × 512. The results validate the effectiveness of our NPNet on improving the conventional image quality metric.

C. Discussion with the Previous Works

We previously mentioned that these methods [5, 11, 16, 17, 47], optimize the noise during the reverse process by incorporating additional modules. These methods have shown promising results in tasks involving compositional generalization. However, these methods often struggle to transfer to other datasets and models, making them not universally applicable. These approaches require the manual interest subject tokens, necessitating extensive test to identify the optimal tokens for a given sentence, which complicates their application across different datasets. Furthermore, modifying the model pipeline usually requires in-depth code changes, making it difficult to achieve straightforward plug-and-play integration with other models. Moreover, these methods demand multiple rounds of noise optimization during the reverse process, resulting in significant time consumption.

In contrast, our approach addresses these challenges from multiple perspectives, offering a more flexible and universal solution. It is capable of cross-model and cross-dataset applications, provides plug-and-play functionality, and incurs minimal time overhead. We first follow the code in Guo et al. [11], and manually provide the subject tokens following Chefer et al. [5]. We conduct the experiments on StableDiffusion-v1-4 [41] on Pick-a-Pic dataset. Note that The authors did not provide the dataset or the necessary subject tokens for the algorithm, so we are limited to using our own dataset. Moreover, we had to manually select and trim the objects in the prompts. Additionally, the author's code requires extensive modifications to the pipeline, making it difficult to adapt for use with other diffusion models. We directly apply the NPNet trained for SDXL to StableDiffusion-v1-4. The experimental results are shown in Table. 7, demonstrating the superiority of our NPNet. In addition, to prove that our method can be directly used in conjunction with other noise optimization methods, we directly use the NPNet trained on SDXL on these mainstream noise optimization methods, and the experimental results in Table. 20 and Fig. 19 prove that our NPNet can further improve the performance of other methods, which validates the generalizability of our NPNet.

D. Additional Experiment Results

D.1. Motivation

To the best of our knowledge, we are the first to propose the *noise prompt learning* framework, as illustrated in Fig. 6. In order to transform the random Gaussian noise into the golden noise, we utilize the *re-denoise sampling*, a straightforward method to boost the semantic faithfulness in the synthesized images, illustrated in Fig. 7, to obtain the noise pairs to construct our *noise prompt dataset*. Notably, in Fig. 9, we validate that re-denoising sampling can enhance the semantic information in the initial noise. When designing the architecture of *noise prompt network*, we discover the singular vectors between the source noise $\mathbf{x}_T$ and target noise $\mathbf{x}'_T$ exhibit remarkable similarity, albeit possibly in opposite directions, as shown in Fig. 8. Building upon this, we design the *singular value prediction* branch.

D.2. Exploration of Data Selection Strategies

Since the target noise collected through *re-denoise sampling* is not always of high quality, it is crucial to choose an appropriate method for data filtering. Effective selection ensures that only high-quality noise pairs are used, which

Table 8. We evaluate the effectiveness of our NPNet on T2I-CompBench benchmark. The results validate the effectiveness of our method.

Method	Attribute Binding			Object Relationship				Complex (↑)
	Color (↑)	Shape (↑)	Texture (↑)	2D-Spatial (↑)	3D-Spatial (↑)	Non-Spatial (↑)	numeracy (↑)	
Standard	0.5850	0.5028	0.5083	0.2097	0.3667	0.3113	0.5013	0.3130
Ours	0.5940	0.5094	0.5207	0.2146	0.3713	0.3151	0.5201	0.3174

is essential for training the NPNet, affecting the model's performance and reliability. For this reason, we conduct experiments on the choice of human preference model to filter our data, shown in Appendix Table 9, here the filtering threshold $m = 0$. The results demonstrate that using HPSv2 ensures data diversity, allowing the filtered data to enhance the model's performance effectively. This approach helps maintain a rich variety of training samples, which contributes to the model's generalization ability and overall effectiveness.

We also explore the influence under difference filtering thresholds m, the results are shown in Appendix Table 10. Our findings reveal that while increasing the filtering threshold m can improve the quality of the training data, it also results in the exclusion of a substantial amount of data, ultimately diminishing the synthesizing diversity of the final NPNet.

D.3. Evaluate the Quality of Synthesized Images

To validate the quality of the synthesized images with our NPNet, we calculate the FID[3] of 5000 images in class-conditional ImageNet with the resolution 512 × 512 on SDXL, shown in Appendix Fig. 12. Note that we just synthesize the "fish" class in the ImageNet dataset, whose directory ids are [n01440764, n01443537, n01484850, n01491361, n01494475, n01496331, n01498041]. The main fish class contains sub-class labels, including "tench", "Tinca tinca", "goldfish", "Carassius auratus", "great white shark", "white shark", "man-eater", "man-eating shark", "Carcharodon carcharias", "tiger shark", "Galeocerdo cuvieri", "hammerhead", "hammerhead shark", "electric ray", "crampfish", "numbfish", "torpedo", "stingray". Each time, we randomly choose one prompt with postfix "a class in ImageNet", in order to synthesize ImageNet-like images. The results reveal that with our NPNet, the T2I diffusion models can synthesize images with higher quality than the standard ones.

D.4. Evaluate the probability density of synthesized images with NPNet

We visualize image distributions before and af- ter NPNet by generating 50 images under identical conditions, using ImageNet images (1 class) as ground truth and text-to- image SDXL for generation in Fig. 13. The experimental results show the use of NPNet, which enables the generated images to more closely resemble the real distribution, and more inclined to high-density reigons.

[3]We follow the code in https://github.com/GaParmar/clean-fid

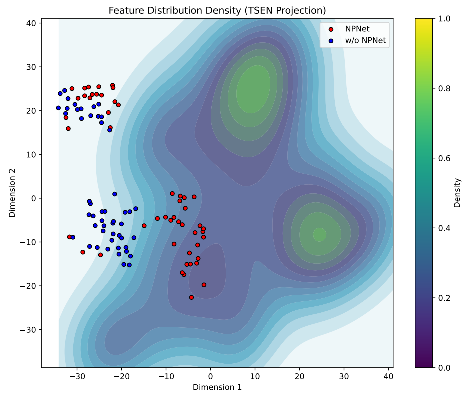


Figure 13. We visualize image distributions before and after NPNet by generating 50 images under identical conditions, using ImageNet images (1 class) as ground truth and text-to-image SDXL for generation. While text-to-image (v.s. class-conditioned) generation may induce distribution shifts, **NPNet shifts generation toward high-density regions.**

D.5. Generalization and Robustness

In this subsection, we provide more experiments to validate the generalization ability and robustness of our NPNet.

Generalization to Models, Datasets and Inference Steps. In Appendix Table 6, we directly apply the NPNet for SDXL to DreamShaper-xl-v2-turbo without fine-tuning on the corresponding data samples. Even so, our NPNet achieves nearly the best performance across arbitrary inference steps, demonstrating the strong generalization capability of our model. Besides, we also present the winning rate of DreamShaper-xl-v2-turbo and Hunyuan-DiT across 3 different datasets, as presented in Appendix Fig. 14. These experimental results indicate that our method has a high success rate in transforming random Gaussian noise into winning noise, highlighting the effectiveness of our approach.

Table 9. To collect valuable samples, we explore the data selection with different human preference models on SDXL with inference steps 10 on Pick-a-Pic dataset."Standard*" here means no human preference model is applied.

Method	Filter Rate	PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
Standard*	-	21.21	25.95	5.9608	40.47
PickScore	34.28%	21.23	25.90	5.9750	32.56
HPSv2	23.88%	21.24	26.01	5.9675	42.47
AES	47.62%	21.22	25.83	5.9636	43.23
ImageReward	35.02%	21.2139	25.70	5.9580	32.39
PickScore+HPSv2	41.93%	21.23	25.75	5.9514	38.59
PickScore+ImageReward	52.57%	21.14	25.97	5.9936	42.19
All	74.41%	21.21	25.95	5.9608	40.47

Table 10. Experiments on the HPSv2 filtering threshold. We conducted experiments on SDXL on Pick-a-Pic dataset to investigate the impact of adding a threshold during the filtering process, like $s_0 + m < s_0'$, where s_0 and s_0' are the human preference scores of denoising images $\mathbf{x}_0$ and $\mathbf{x}_0'$.

Threshold	Filter Rate	PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
$m = 0$	23.88%	21.86	28.68	6.0540	66.21
$m = 0.005$	41.21%	21.78	28.79	6.0703	60.30
$m = 0.01$	50.19%	21.72	28.64	6.0766	61.60
$m = 0.02$	83.47%	21.82	28.78	6.0447	65.46

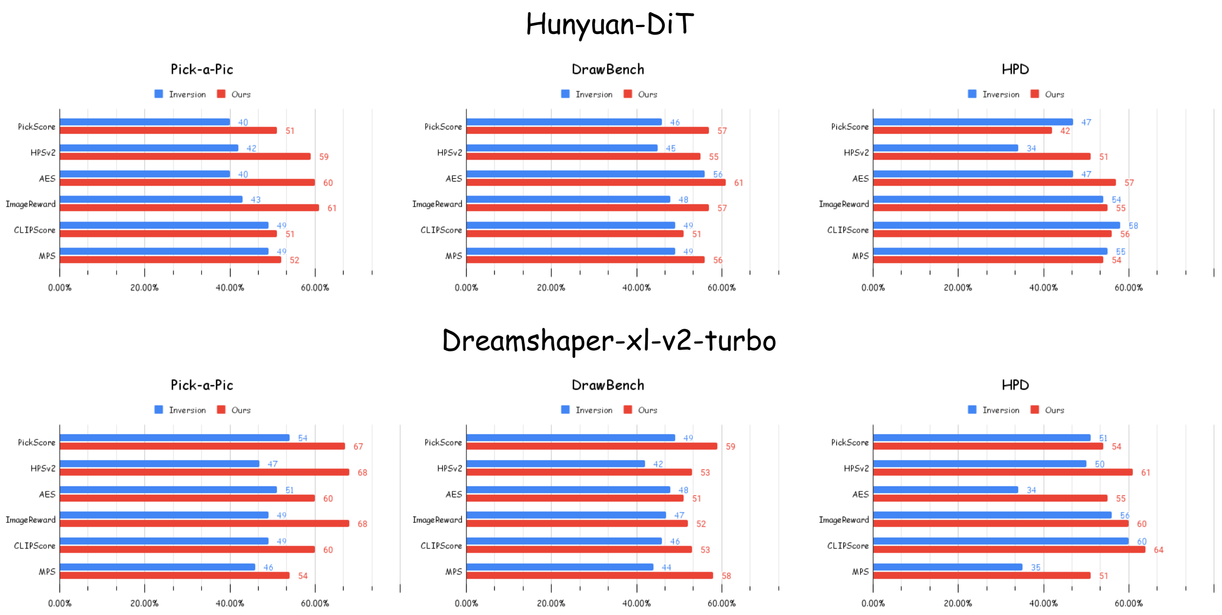


Figure 14. The winning rate comparison on DreamShaper-xl-v2-turbo and Hunyun-DiT across 3 datasets, including Pick-a-Pic, DrawBench and HPD v2 (HPD) with inference steps 50. The results demonstrate the superiority of our NPNet.

Generalization to Random Seeds. As we mentioned in Sec. 3.2, the random seed range for the training set is $[0, 1024]$, while the random seed range for the test set is $[0, 100]$. This discrepancy may lead to our NPNet potentially overfitting on specific random seeds. To evaluate the performance of our NPNet under arbitrary random seeds, we artificially modified the seeds in the test set. The experimental results on the Pick-a-Pic dataset are presented in Appendix Table 19, demonstrating that our NPNet maintains strong performance across a variety of random seed conditions, making it suitable for diverse scenarios in real-world applications. The results demonstrate that our NPNet exhibits strong generalization capabilities across the out-of-distribution random seed ranges.

Robustness to Inference Steps and Hyper-parameters. In Appendix Fig. 16, we conduct the experiments on DreamShaper-xl-v2-turbo and Hunyuan-DiT under various inference steps. The curve representing our method consistently remains at the top, demonstrating that our model achieves the best performance across various inference steps, further validating the robustness of our approach. To further support our claims, we present the winning rate of SDXL, DreamShaper-xl-v2-turbo and Hunyuan-DiT under various inference in two different datasets, shown in Appendix Fig. 17. These promising results validate the effectiveness of our NPNet.

Robustness to Hyper-parameters. We also conduct the experiments on different hyper-parameter settings, including the CFG value, batch size and training epochs, shown in Appendix Table 12. It reveals that the optimal setting of these parameters are CFG 5.5, batch size 64, and training epoch 30. For all the experiments in the paper, we all use this setting. Moveover, we explore the influence of the number of training samples, shown in Appendix Table 17, we believe that a large dataset can ensure data diversity and improve the model's robustness and generalization ability.

D.6. Efficiency Analysis and Ablation Studies

Efficiency Analysis. As a plug-and-play module, it raises concerns about potential increases in inference latency and memory consumption, which can significantly impact its practical value. In addition to Fig. 5 presented in the main paper, we also measure the time required to synthesize each image under the same inference step conditions, shown in Appendix Table 11. Our model achieves a significant improvement in image quality with only a 0.4-second inference delay. Additionally, as shown in Appendix Fig. 15, our model requires just 500 MB of extra memory. These factors highlight the lightweight and efficient nature of our model, underscoring its broad application potential.

Ablation Studies. We explore the influence of the text embedding term $\mathbf{e}$. Although in Appendix Table 15, the value of α is very small, the results in Appendix Table 16 still demonstrate the importance of this term. It can facilitate a refined adjustment of how much semantic information influences the model's predictions, enabling the semantic relevance between the text prompt and synthesized images, and the diversity of the synthesized images.

D.7. Experiments on Large-scale Dataset

To evaluate the effectiveness of our NPNet, we conduct the experiments on a large-scale dataset, GenEval, across different T2I models. The results are shown in Appendix Table 13 and Table 14. The all improved metrics reveal the superiority of our NPNet, suggesting that our NPNet can improve the compositional image properties of the synthesized images.

E. Theoretical Understanding of Re-denoise Sampling

In main paper Sec. 3.1, we utilize *re-denoise sampling* to produce noise pairs. we propose to utilize $\text{DDIM-Inversion}(\cdot)$ to obtain the noise from the previous step. Specifically, the joint action of DDIM-Inversion and CFG can induce the initial noise to attach semantic information. The mechanism behind this method is that $\text{DDIM-Inversion}(\cdot)$ injects semantic information by leveraging the guidance scale in classifier-free guidance (CFG) inconsistency:

Theorem E.1. *Given the initial Gaussian noise* $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ *and the operators* $\text{DDIM-Inversion}(\cdot)$ *and* $\text{DDIM}(\cdot)$. *Using re-denoise sampling, we can obtain that:*

$$\begin{aligned}\mathbf{x}'_T = \mathbf{x}_T + \frac{\alpha_T \sigma_{T-k} - \alpha_{T-k}\sigma_T}{\alpha_{T-k}} \Big[&(\omega_l - \omega_w)(\epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T - \frac{k}{2}|\mathbf{c}) \\ &- \epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T - \frac{k}{2}|\varnothing))\Big].\end{aligned} \tag{10}$$

where k stands for the DDIM sampling step, $\mathbf{c}$ *is the text prompt, and ω_l and ω_w are CFG at the timestep T and CFG at timestep $T-k$, respectively.*

Proof. One step *re-denoise sampling* represents one additional step forward sampling and one step reverse sampling against the initial Gaussian noise, which can be denoted as

$$\mathbf{x}'_T = \text{DDIM-Inversion}(\text{DDIM}(\mathbf{x}_T)), \tag{11}$$

where $\text{DDIM-Inversion}(\cdot)$ refers to the sampling algorithm in Eqn. 1 when $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are interchanged. We can

Table 11. Experiments on different samplers $w.r.t.$ inference time cost on SDXL. The NPNet trained on noise samples produced by the deterministic sampler DDIM, demonstrates impressive generalization to non-deterministic samplers, incurring only minimal additional time costs.

Methods		PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)	Time Cost(second per image)
DDIMScheduler [48]	Standard	21.69	28.48	6.0373	58.01	11.69
	NPNet (ours)	21.86	28.68	6.0540	65.01	12.10
DPMSolverMultistepScheduler [31]	Standard	21.66	28.41	5.9513	55.01	9.84
	NPNet (ours)	21.72	28.81	5.9744	67.30	10.43
DDPMScheduler [15]	Standard	21.78	28.72	6.1353	72.91	10.86
	NPNet (ours)	21.91	29.24	6.1505	78.50	11.43
EulerAncestralDiscreteScheduler [22]	Standard	21.72	28.66	6.0740	67.01	10.28
	NPNet (ours)	21.84	28.96	6.0886	85.05	10.86
PNDMScheduler [29]	Standard	21.78	29.35	5.9809	62.81	11.40
	NPNet (ours)	21.81	29.74	6.0256	67.58	11.82
KDPM2AncestralDiscreteScheduler [22]	Standard	21.81	29.22	6.0382	77.59	16.25
	NPNet (ours)	21.93	29.62	6.0951	84.78	16.73
HeunDiscreteScheduler [22]	Standard	21.83	28.71	6.0705	64.33	16.74
	NPNet (ours)	21.86	28.98	6.0892	73.31	17.04

Table 12. Ablation studies of the hyper-parameters on SDXL on Pick-a-Pic dataset.

Hyper-parameters		PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
Epochs	5	21.77	28.62	6.0688	65.84
	10	21.76	28.67	6.0629	60.59
	15	21.69	28.62	6.0721	58.74
	30	21.86	28.68	6.054	65.01
Guidance Scale	$\omega_1 = 1$	20.11	21.80	6.0601	-51.30
	$\omega_1 = 3$	21.53	27.28	6.0880	44.49
	$\omega_1 = 5.5$	21.86	28.68	6.0540	65.01
	$\omega_1 = 7$	21.81	29.12	6.0529	70.31
Batch Size	$bs = 16$	21.76	28.74	6.0677	60.80
	$bs = 32$	21.68	28.68	6.0483	65.47
	$bs = 64$	21.86	28.68	6.0540	65.01

Table 13. We evaluate the effectiveness of our NPNet on larger T2I benchmark, GenEval. The results validate the superiority of our method.

Model		PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
SDXL	Standard	22.58	28.00	5.4291	55.44
	Inversion	22.64	28.24	5.4266	58.37
	NPNet(ours)	22.73	28.41	5.4506	65.56
DreamShaper-xl-v2-turbo	Standard	23.51	31.20	5.5234	97.52
	Inversion	23.47	30.85	5.5157	95.91
	NPNet(ours)	23.59	31.24	5.5577	100.06
Hunyuan-DiT	Standard	23.15	30.46	5.6233	111.73
	Inversion	23.16	30.63	5.6210	111.98
	NPNet(ours)	23.15	30.69	5.7040	115.57

Table 14. We evaluate the effectiveness of our NPNet on Genval benchmark.

	Single (↑)	Two (↑)	Counting (↑)	Colors (↑)	Positions (↑)	Color Attribution (↑)	Overall (↑)
Standard	97.50	63.64	36.25	85.11	8.00	19.00	51.58
NPNet (ours)	98.75	67.68	37.50	82.98	13.00	21.00	53.49

rewrite it in forms of linear transformation:

$$\begin{aligned}
\mathbf{x}'_T &= \alpha_T \left(\frac{\mathbf{x}_{T-k} - \sigma_{T-k}\epsilon_\theta(\mathbf{x}_{T-k}, T-k)}{\alpha_{T-k}} \right) + \sigma_T \epsilon_\theta(\mathbf{x}_{T-k}, T-k) \\
\mathbf{x}'_T &= \alpha_T \\
&\left(\frac{\alpha_{T-k}\left(\frac{\mathbf{x}_T - \sigma_T\epsilon_\theta(\mathbf{x}_T, T)}{\alpha_T}\right) + \sigma_{T-k}\epsilon_\theta(\mathbf{x}_T, T) - \sigma_{T-k}\epsilon_\theta(\mathbf{x}_{T-k}, T-k)}{\alpha_{T-k}} \right) \\
&+ \sigma_T \epsilon_\theta(\mathbf{x}_{T-k}, T-k) \\
\mathbf{x}'_T &= \mathbf{x}_T - \sigma_T\epsilon_\theta(\mathbf{x}_T, T) + \frac{\alpha_T\sigma_{T-k}}{\alpha_{T-k}}\epsilon_\theta(\mathbf{x}_T, T) - \frac{\alpha_T\sigma_{T-k}}{\alpha_{T-k}}\epsilon_\theta(\mathbf{x}_{T-k}, T-k) \\
&+ \sigma_T \epsilon_\theta(\mathbf{x}_{T-k}, T-k) \\
\mathbf{x}'_T &= \mathbf{x}_T + \frac{\alpha_T\sigma_{T-k} - \alpha_{T-k}\sigma_T}{\alpha_{T-k}}\left[\epsilon_\theta(\mathbf{x}_T, T) - \epsilon_\theta(\mathbf{x}_{T-k}, T-k)\right],
\end{aligned} \tag{12}$$

where k stands for the DDIM sampling step. Substitute $\epsilon_\theta(\mathbf{x}_t, t) = (\omega+1)\epsilon_\theta(\mathbf{x}_t, t|\mathbf{c}) - \omega\epsilon_\theta(\mathbf{x}_t, t|\varnothing)$ into Eq. 12, we can obtain

$$\begin{aligned}
\mathbf{x}'_T &= \mathbf{x}_T + \frac{\alpha_T\sigma_{T-k} - \alpha_{T-k}\sigma_T}{\alpha_{T-k}}\Big[(\omega_l+1)\epsilon_\theta(\mathbf{x}_T, T|\mathbf{c}) - \omega_l\epsilon_\theta(\mathbf{x}_T, T|\varnothing)) \\
&- (\omega_w+1)\epsilon_\theta(\mathbf{x}_{T-k}, T-k|\mathbf{c}) + \omega_w\epsilon_\theta(\mathbf{x}_{T-k}, T-k|\varnothing))\Big].
\end{aligned} \tag{13}$$

Where ω_l and ω_w refer to the classifier-free guidance scale at the timestep T and the classifier-free guidance scale at timestep $T-k$, respectively. $\mathbf{c}$ stands for the text prompt (*i.e.*, condition). Consider the first-order Taylor expansion $\epsilon_\theta(\mathbf{x}_{T-k}, T-k|\mathbf{c}) = \epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\mathbf{c}) + \frac{\mathbf{x}_{T-k}-\mathbf{x}_{T-\frac{k}{2}}}{2}\frac{\partial\epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\mathbf{c})}{\partial\mathbf{x}_{T-\frac{k}{2}}} + \frac{k}{2}\frac{\partial\epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\mathbf{c})}{\partial T-\frac{k}{2}} + \mathcal{O}((\frac{k}{2})^2)$ and $\epsilon_\theta(\mathbf{x}_T, T|\mathbf{c}) = \epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\mathbf{c}) + \frac{\mathbf{x}_T-\mathbf{x}_{T-\frac{k}{2}}}{2}\frac{\partial\epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\mathbf{c})}{\partial\mathbf{x}_{T-\frac{k}{2}}} - \frac{k}{2}\frac{\partial\epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\mathbf{c})}{\partial T-\frac{k}{2}} + \mathcal{O}((\frac{k}{2})^2)$, when $\mathbf{x}_T$ satisfies the condition $\left\|\frac{\|\|\mathbf{x}_T-\mathbf{x}_{T-k}\|\|}{k}\right\| \leq L$, where $L < +\infty$, Eq. 13 can be transformed into:

$$\begin{aligned}
\mathbf{x}'_T &= \mathbf{x}_T + \frac{\alpha_T\sigma_{T-k} - \alpha_{T-k}\sigma_T}{\alpha_{T-k}}\Big[(\omega_l - \omega_w)(\epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\mathbf{c}) \\
&- \epsilon_\theta(\mathbf{x}_{T-\frac{k}{2}}, T-\frac{k}{2}|\varnothing))\Big].
\end{aligned} \tag{14}$$

The proof is complete. □

By using Eqn. 14, when there is a gap between ω_l and ω_w, *re-denoise sampline* can be considered as a technique to inject semantic information under the guidance of future timestep ($t = T - \frac{k}{2}$) CFG into the initial Gaussian noise.

F. Discussion

Limitations. Although our experimental results have demonstrated the superiority of our method, the limitations still exist. As a machine learning framework, our method also faces classic challenges from training data quality and model architecture design. First, noise prompt data quality sets the performance limit of our method. The data quality is heavily constrained by *re-denoise sampling* and data selection, but lack comprehensive understanding. For example, there exists the potential risk that the proposed data collection pipeline could introduce extra bias due to the AI-feedback-based selection. Second, the design of NPNet is still somewhat rudimentary. While ablation studies support each component of NPNet, it is highly possible that more elegant and efficient architectures may exist and work well for the novel noise prompt learning task. Optimizing model architectures for this task still lacks principled understanding and remain to be a challenge.

Future Directions. Our work has various interesting future directions. First, it will be highly interesting to investigate improved data collection methods in terms of both performance and trustworthiness. Second, we will design more streamlined structures rather than relying on a parallel approach with higher performance or higher efficiency. For example, we may directly utilize a pre-trained diffusion model to synthesize golden noise more precisely. Third, we will further analyze and improve the generalization of our method, particularly in the presence of out-of-distribution prompts or even beyond the scope of T2I tasks.

Table 15. The values of the two trainable parameters α and β.

Model	α	β
SDXL	1.00E-04	-0.0189
DreamShaper-xl-v2-turbo	7.00E-05	0.0432
Hunyuan-DiT	2.00E-04	0.0018

G. More Visualization Results

We present more visualization results on different dataset, different diffusion models and with different noise optimization methods on Fig. 22, 23, 24, 25, 26, 27, 28, 29, 31, 30, 32 and 33.

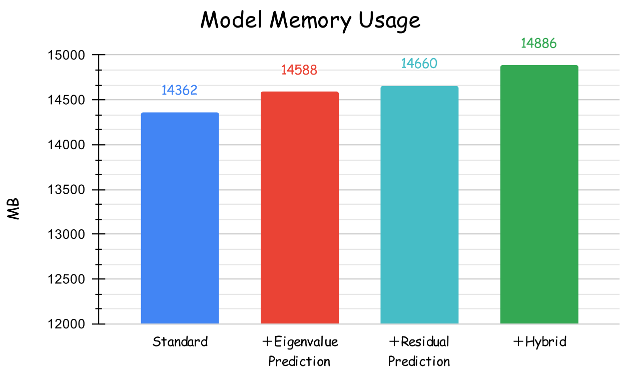


Figure 15. Our NPNet requires only about 500MB, illustrating the light-weight and efficiency of our model.

Table 16. We explore the influence of text embedding $\mathbf{e}$. The results reveal that text embedding $\mathbf{e}$ is crucial in *noise prompt learning*, which aims to inject the semantic information into the noise.

Method	PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
NPNet *w/o text embedding* $\mathbf{e}$	21.72	28.70	6.0513	62.14
NPNet	21.86	28.68	6.0540	65.01

Algorithm 1: Noise Prompt Dataset Collection

```
1: Input: Timestep $t \in [0, \cdots, T]$, random Gaussian noise $\mathbf{x}_T$, text prompt $\mathbf{c}$, DDIM operateor $\textbf{DDIM}(\cdot)$, DDIM inversion operator $\textbf{DDIM-Inversion}(\cdot)$, human preference model $\Phi$ and filtering threshold $m$.
2: Output: Source noise $\mathbf{x}_T$, target noise $\mathbf{x}'_T$ and text prompt $\mathbf{c}$.
3: Sample Gaussian noise $x_T$
4: # re-denoise sampling, see Sec. 3.1 in the main paper
5: $\mathbf{x}_{T-1} = \textbf{DDIM}(\mathbf{x}_T)$
6: $\mathbf{x}'_T = \textbf{DDIM-Inversion}(\mathbf{x}_{T-1})$
7: # standard diffusion reverse process
8: $\mathbf{x}_0 = \textbf{DDIM}(\mathbf{x}_T)$
9: $\mathbf{x}'_0 = \textbf{DDIM}(\mathbf{x}'_T)$
10: # data filtering via the human preference model, see Sec. 3.1
11: if $\Phi(\mathbf{x}_0, \mathbf{c}) + m < \Phi(\mathbf{x}'_0, \mathbf{c})$ then
12:     store $(\mathbf{x}_T, \mathbf{x}'_T, \mathbf{c})$
13: end if
```

Algorithm 2: Noise Prompt Network Training

```
1: Input: Noise prompt dataset $\mathcal{D} := \{\mathbf{x}_{T_i}, \mathbf{x}'_{T_i}, \mathbf{c}_i\}_{i=1}^{|\mathcal{D}|}$, noise prompt model $\phi$ parameterized by singular value predictor $f(\cdot)$ and residual predictor $g(\cdot,\cdot)$, the frozen pre-trained text encoder $\mathcal{E}(\cdot)$ from diffusion model, normalization layer $\sigma(\cdot,\cdot)$, MSE loss function $\ell$, and two trainable parameters $\alpha$ and $\beta$.
2: Output: The optimal noise prompt model $\phi^*$ trained on the training set $\mathcal{D}$.
3: # singular value prediction, see (6)
   $\tilde{\mathbf{x}}'_{T_i} = f(\mathbf{x}_{T_i})$
4: # residual prediction, see (7)
   $\hat{\mathbf{x}}_{T_i} = g(\mathbf{x}_{T_i}, \mathbf{c}_i)$
5: # see (9)
   $\mathbf{x}'_{T_{pred_i}} = \alpha\sigma(\mathbf{x}_{T_i}, \mathcal{E}(\mathbf{c}_i)) + \tilde{\mathbf{x}}'_{T_i} + \beta\hat{\mathbf{x}}_{T_i}$
6: $\mathcal{L}_i = \ell(\mathbf{x}'_{T_{pred_i}}, \mathbf{x}'_{T_i})$
7: update $\phi$
8: return $\phi^*$
```

Algorithm 3: Inference with Noise Prompt Network

```
1: Input: Text prompt $\mathbf{c}$, the trained noise prompt network $\phi^*(\cdot,\cdot)$ and the diffusion model $f(\cdot,\cdot)$.
2: Output: The golden clean image $\mathbf{x}'_0$.
3: Sample Gaussian noise $x_T$
4: # get the golden noise
   $\mathbf{x}'_{T_{pred}} = \phi^*(\mathbf{x}_T, \mathbf{c})$
5: # standard inference pipeline
   $\mathbf{x}'_0 = f(\mathbf{x}'_{T_{pred}}, \mathbf{c})$
6: return $\mathbf{x}'_0$
```

Table 17. In order to explore the scaling law [21] in NPNet, we train our NPNet with different numbers of training samples on SDXL on Pick-a-Pic dataset.

Numbers of Training Samples	Method	PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)	CLIPScore(%) (↑)	MPS(%) (↑)
	Standard	21.69	28.48	6.0373	58.01	82.04	-
3W	NPNet (ours)	21.82	28.78	6.0750	67.26	82.17	50.63
6W	NPNet (ours)	21.75	28.65	6.0392	64.56	81.98	51.60
10W	NPNet (ours)	21.86	28.68	6.0540	65.01	84.08	52.15

Table 18. We evaluate NPNet on few steps T2I diffusion models, like LCM [33], PCM [51] and SDXL-Lightning [26] on GenEval dataset. Here we use the NPNet from SDXL, and the results demonstrate NPNet can generalize well to different kinds of T2I diffusion models, boosting their performance directly.

Model		PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)	CLIPScore (↑)
SDXL-Lightning	Standard	22.85	29.12	5.6521	59.02	0.8093
(4 steps)	NPNet(ours)	23.03	29.71	5.7178	72.67	0.8150
LCM	Standard	22.30	26.52	5.4932	33.21	0.8050
(4 steps)	NPNet(ours)	22.38	26.83	5.5598	37.08	0.8123
PCM	Standard	22.05	26.98	5.5245	23.28	0.8031
(8 steps)	NPNet(ours)	22.22	27.59	5.5667	35.01	0.8175

Table 19. Random seed generalization experiments on SDXL with difference inference steps on Pick-a-Pic dataset. The random seeds of our training set range from $[0, 1024]$, containing the random seeds of our test set. To explore the generalization ability of NPNet on out-of-distribution random seeds, we manually adjust the random seed range of the test set.

Inference Steps	Random Seed Range		PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)
50	[0, 1024] (Original)	Standard	21.69	28.48	6.0373	58.01
		Inversion	21.71	28.57	6.0503	63.27
		NPNet (ours)	21.86	28.68	6.0540	65.01
	[2500, 3524]	Standard	21.63	28.57	5.9748	67.09
		Inversion	21.71	28.75	5.9875	70.92
		NPNet (ours)	21.81	29.02	5.9917	80.83
	[5000, 6024]	Standard	21.74	28.82	6.0534	78.02
		Inversion	21.78	29.04	6.0418	76.05
		NPNet (ours)	21.83	29.09	6.4220	79.84
	[7500, 7524]	Standard	21.70	29.12	6.0251	78.71
		Inversion	21.78	29.18	6.0541	82.69
		NPNet (ours)	21.81	29.02	6.0641	89.53
100	[0, 1024] (Original)	Standard	21.71	28.70	6.0041	61.76
		Inversion	21.72	28.70	6.0061	61.73
		NPNet (ours)	21.86	29.10	6.0761	74.57
	[2500, 3524]	Standard	21.69	28.73	5.9946	69.22
		Inversion	21.74	28.92	5.9863	71.86
		NPNet (ours)	21.81	29.04	5.9977	78.75
	[5000, 6024]	Standard	21.81	28.40	6.0489	79.57
		Inversion	21.85	28.66	6.0374	79.94
		NPNet (ours)	21.88	29.16	6.0576	83.87
	[7500, 7524]	Standard	21.73	28.67	6.0347	77.66
		Inversion	21.80	28.75	6.0600	82.33
		NPNet (ours)	21.86	29.12	6.0502	89.79

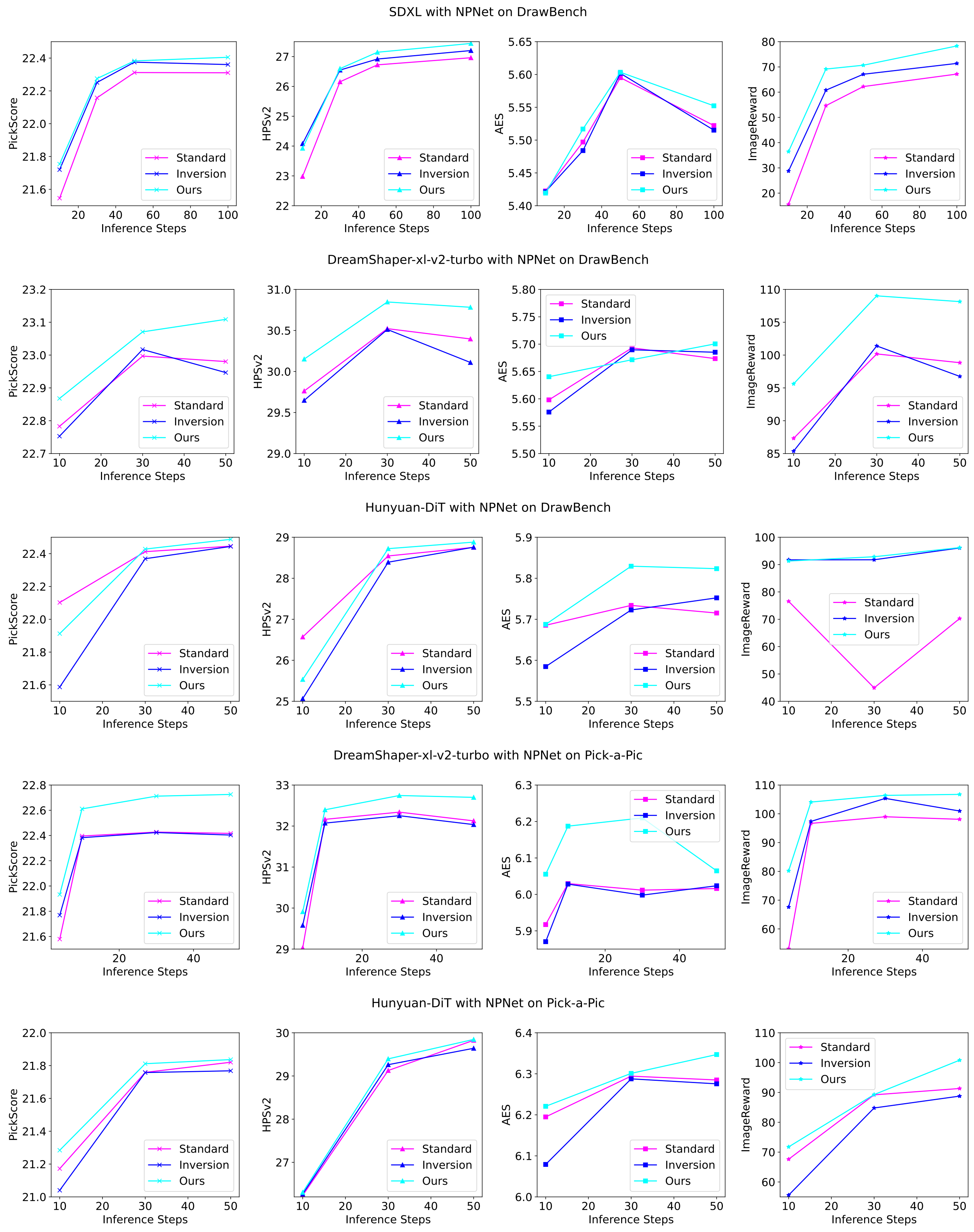


Figure 16. Visualization of performance w.r.t inference steps on SDXL, DreamShaper-xl-v2-turbo and Hunyuan-DiT on Pick-a-Pic dataset and DrawBench dataset. The results demonstrate the strong generalization ability of our NPNet.

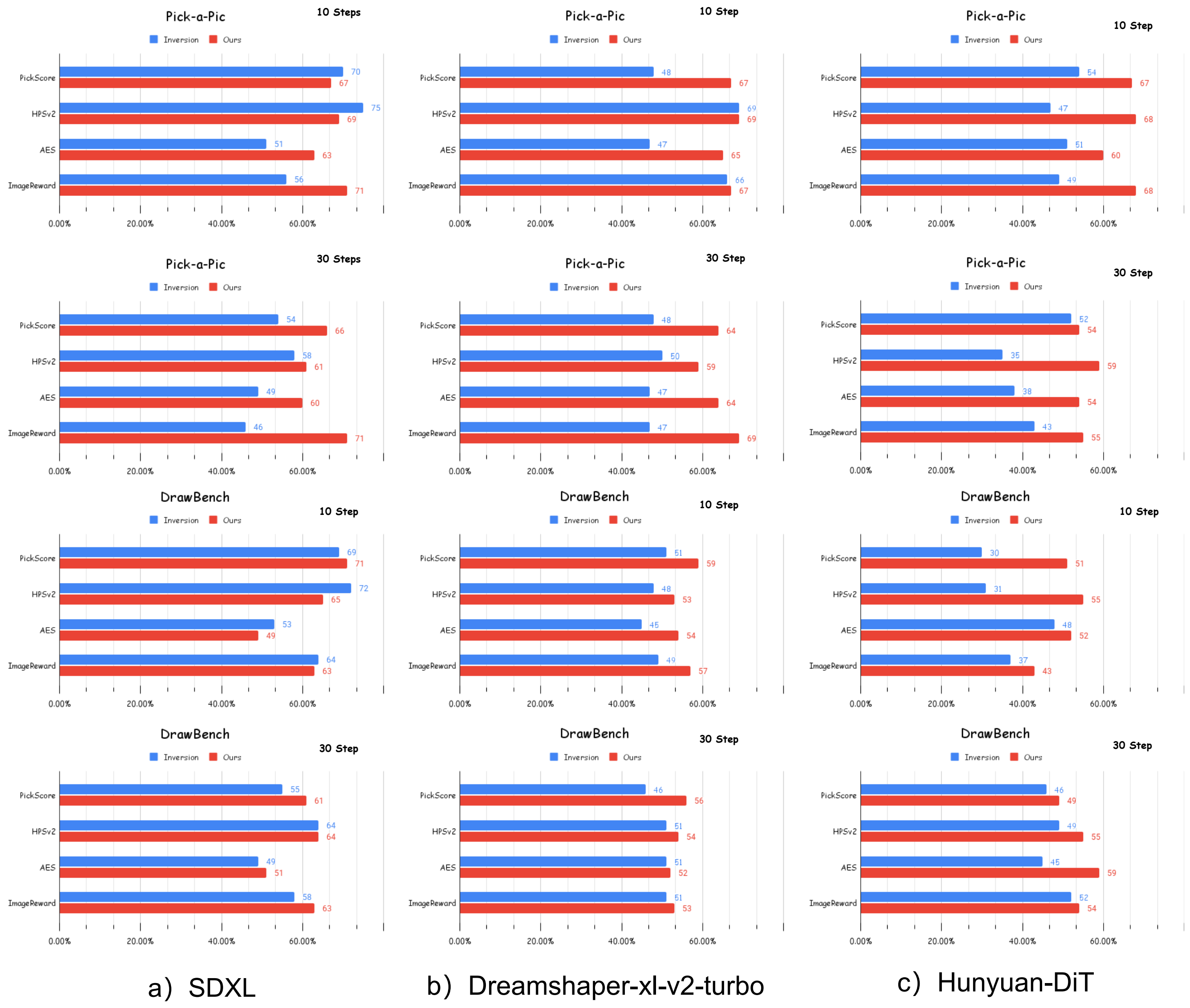


Figure 17. The winning rate comparison on SDXL, DreamShaper-xl-v2-turbo and Hunyuan-DiT across **2** datasets, including DrawBench and HPD v2 (HPD). The results reveal that our NPNet is more effective in transforming random Gaussian noise into golden noises in different inference steps across different datasets.

Pick-a-Pic

A curious cat exploring a haunted mansion.

A spanish water dog breed as arthur morgan from red dead redemption.

Movie Still of The Joker wielding a red Lightsaber, Darth Joker a sinister evil clown prince of crime, HD Photograph.

A closeup portrait of a playful maid, undercut hair, apron, amazing body, pronounced feminine feature, kitchen, , freckles, flirting with camera.

Art nouveau style, a crystal emperor moth with iridescent wings at the center of the universe with 7 Cosmic Rays emanating from it, futuristic, astrological, metaphysical, mystical, golden mean, HD 4K, sharp detail, photo-realistic.

Anime style. Girl at a wood table.

Ancient prayer figures pond sitter bouldartmoor carra, Christian Krohg, melancholia

Black and white 1905 year futuristic portrait of professional photographer with camera in hand in a desert sadly covered by mushrooms.

• • •

DrawBench

Three cats and three dogs sitting on the grass.

A real life photography of super mario, 8k Ultra HD.

An IT-guy trying to fix hardware of a PC tower is being tangled by the PC cables like Laokoon. Marble, copy after Hellenistic original from ca. 200 BC. Found in the Baths of Trajan, 1506.

New York Skyline with 'Deep Learning' written with fireworks on the sky.

A realistic photo of a Pomeranian dressed up like a 1980s professional wrestler with neon green and neon orange face paint and bright green wrestling tights with bright orange boots.

A pear cut into seven pieces arranged in a ring.

A triangular orange picture frame. An orange picture frame in the shape of a triangle.

An American multinational technology company that focuses on artificial intelligence, search engine, online advertising, cloud computing, computer software, quantum computing, e-commerce, and consumer electronics.

• • •

HPD v2

An anime man in flight uniform with hyper detailed digital artwork and an art style inspired by Klimt

A hand-drawn cute gnome holding a pumpkin in an autumn disguise.

Gnomes are playing music during Independence Day festivities in a forest near Lake George.

A small green dinosaur toy with orange spots standing on its hind legs and roaring with its mouth open.

A cute rainbow kitten with different colored eyes in the chibi-style of Studio Ghibli is featured on a postcard.

A slime monster.

A digital anime portrait of tatsumaki with green curly hair and green eyes wearing a jacket.

Portrait of young Jerry Lewis in comic style.

• • •

GenEval

A photo of a stop sign and a toaster.

A photo of a red train and a purple bear.

A photo of a purple tennis racket and a black sink.

A photo of a blue vase and a black banana.

A photo of a bear above a spoon.

A photo of a bed right of a frisbee.

A photo of a zebra below a computer keyboard.

photo of a wine glass right of a hot dog.

A photo of a potted plant and a backpack.

A photo of four computer keyboards.

• • •

Figure 18. Part of our test datasets. All of the training and test datasets will be released.

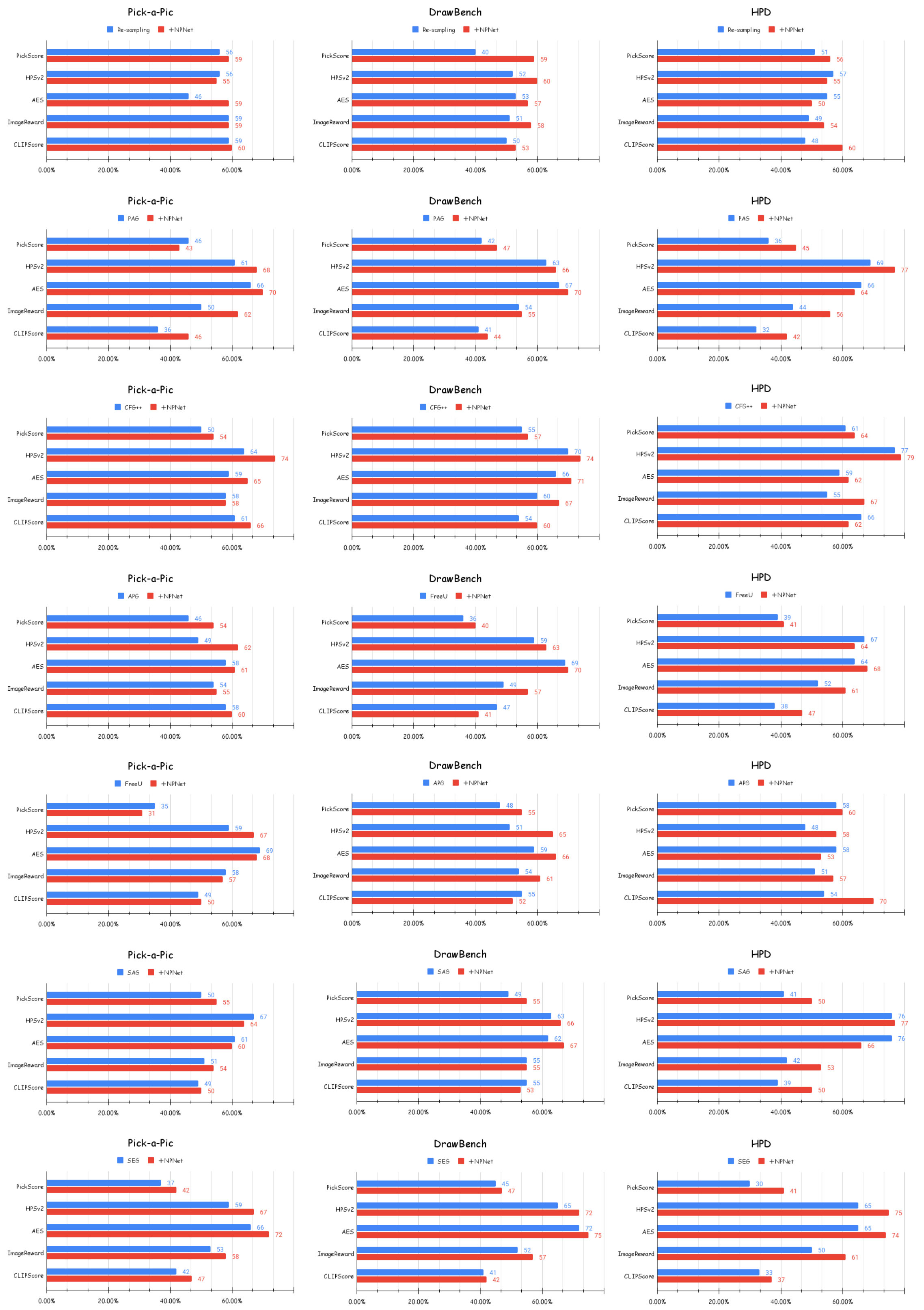


Figure 19. Winning rate of noise optimization methods on three different datasets, using SDXL. The results demonstrate that when NPNet is used in conjunction with other noise optimization methods, it can enhance the winning rate of existing approaches to some extent.

Table 20. We combine other popular noise optimization methods with our NPNet, evaluating on three different datasets on SDXL. The experimental results indicate that when NPNet is used alongside other methods, it significantly enhances their performance, further validating the effectiveness and generalizability of our approach.

Dataset	Method	PickScore (↑)	HPSv2 (↑)	AES (↑)	ImageReward (↑)	CLIPscore (↑)
Pick-a-Pic	Standard	21.69	28.48	6.0373	58.01	0.8204
	Re-sampling [32]	21.77	28.63	5.9875	64.94	0.8327
	+ NPNet (ours)	21.90	29.29	6.1491	71.09	0.8386
	PAG [1]	21.64	29.45	6.2246	55.91	0.7966
	+ NPNet (ours)	21.70	29.80	6.2411	62.03	0.8079
	CFG++ [9]	21.67	29.54	6.1239	75.16	0.8322
	+ NPNet (ours)	21.82	29.84	6.1703	81.60	0.8374
	APG [44]	21.69	28.48	6.1472	65.86	0.8295
	+ NPNet (ours)	21.86	29.13	6.1629	76.50	0.8322
	FreeU [47]	21.39	29.12	6.2134	79.74	0.8094
	+ NPNet (ours)	21.42	29.44	6.2194	77.42	0.8059
	SAG [18]	21.70	29.42	6.1507	59.61	0.8162
	+ NPNet (ours)	21.79	29.63	6.1535	65.75	0.8193
	SEG [16]	21.47	29.23	6.2118	61.60	0.8060
	+ NPNet (ours)	21.60	29.72	6.2253	72.51	0.8077
DrawBench	Standard	22.31	26.72	5.5952	62.21	0.8077
	Re-sampling	22.30	26.96	5.5104	64.07	0.8106
	+ NPNet (ours)	22.47	27.55	5.6487	75.36	0.8136
	PAG	22.25	27.81	5.7082	67.25	0.7965
	+ NPNet (ours)	22.34	28.17	5.7646	72.10	0.7989
	CFG++	22.35	28.28	5.6720	81.69	0.8230
	+ NPNet (ours)	22.45	28.67	5.7035	86.40	0.8263
	APG	22.29	26.94	5.6180	70.15	0.8194
	+ NPNet (ours)	22.46	27.73	5.6445	81.30	0.8182
	FreeU	21.97	27.31	5.7379	63.18	0.7973
	+ NPNet (ours)	22.04	27.87	5.7464	77.53	0.7998
	SAG	22.30	27.64	5.6651	64.56	0.8144
	+ NPNet (ours)	22.37	27.98	5.6998	72.18	0.8104
	SEG	22.16	28.08	5.7867	64.37	0.7979
	+ NPNet (ours)	22.30	28.51	5.8093	74.12	0.7985
HPD	Standard	22.88	29.71	5.9985	96.63	0.8734
	Re-sampling	22.91	29.78	5.9948	97.39	0.8775
	+ NPNet (ours)	22.96	30.16	6.0098	98.34	0.8787
	PAG	22.80	30.54	6.1180	90.94	0.8536
	+ NPNet (ours)	22.92	30.97	6.1091	106.24	0.8584
	CFG++	23.03	30.94	6.0269	107.70	0.8875
	+ NPNet (ours)	23.08	31.07	6.0533	109.06	0.8920
	APG	22.94	29.64	6.0572	97.90	0.8836
	+ NPNet (ours)	23.06	30.40	6.0232	111.55	0.8945
	FreeU	22.72	30.52	6.0907	97.03	0.8580
	+ NPNet (ours)	22.73	30.83	6.1148	102.77	0.8631
	SAG	22.86	30.37	6.0680	93.62	0.8670
	+ NPNet (ours)	22.92	30.71	6.0177	97.20	0.8748
	SEG	22.74	30.47	6.0841	91.86	0.8569
	+ NPNet (ours)	22.80	30.74	6.1160	104.16	0.8591

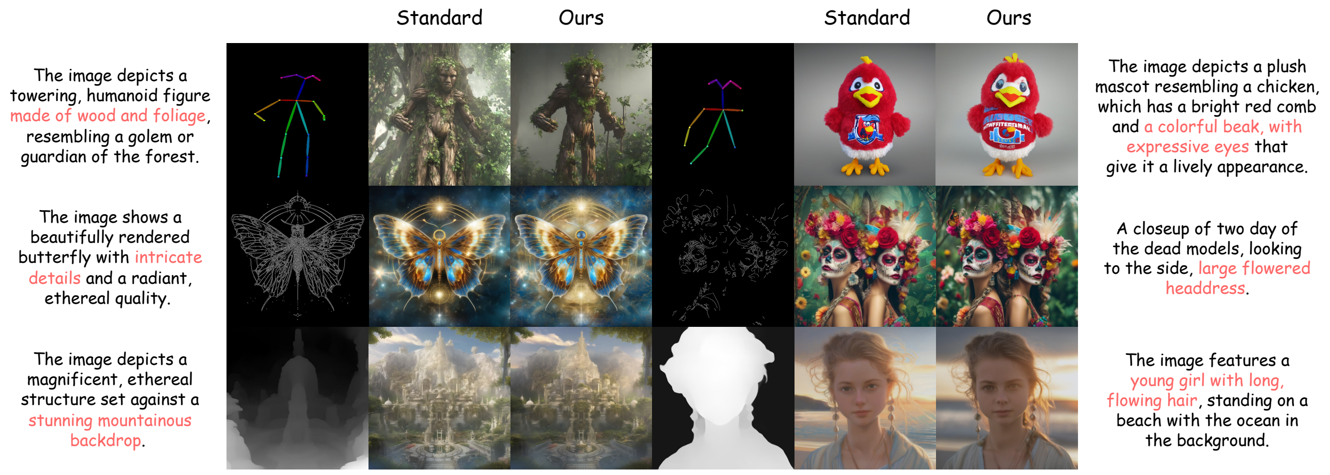
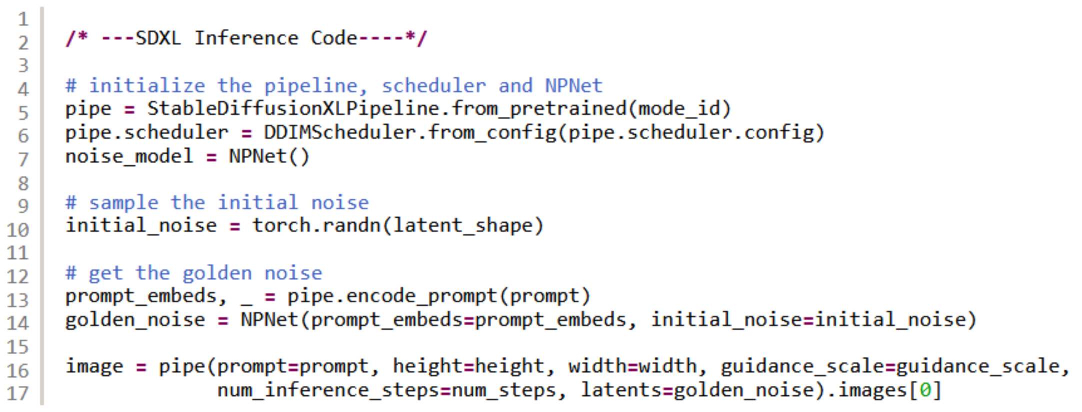

Standard
Ours
Standard
Ours
The image depicts a towering, humanoid figure made of wood and foliage, resembling a golem or guardian of the forest.
The image depicts a plush mascot resembling a chicken, which has a bright red comb and a colorful beak, with expressive eyes that give it a lively appearance.
The image shows a beautifully rendered butterfly with intricate details and a radiant, ethereal quality.
A closeup of two day of the dead models, looking to the side, large flowered headdress.
The image depicts a magnificent, ethereal structure set against a stunning mountainous backdrop.
The image features a young girl with long, flowing hair, standing on a beach with the ocean in the background.

Figure 20. ControlNet visualization with our NPNet on SDXL, including conditions like openpose, canny and depth. *Middle* is the standard method, and *right* is our result. Our NPNet can be directly applied to the corresponding downstream tasks without requiring any modifications to the pipeline of the T2I diffusion model.


```

/* ---SDXL Inference Code----*/

# initialize the pipeline, scheduler and NPNet
pipe = StableDiffusionXLPipeline.from_pretrained(mode_id)
pipe.scheduler = DDIMScheduler.from_config(pipe.scheduler.config)
noise_model = NPNet()

# sample the initial noise
initial_noise = torch.randn(latent_shape)

# get the golden noise
prompt_embeds, _ = pipe.encode_prompt(prompt)
golden_noise = NPNet(prompt_embeds=prompt_embeds, initial_noise=initial_noise)

image = pipe(prompt=prompt, height=height, width=width, guidance_scale=guidance_scale,
             num_inference_steps=num_steps, latents=golden_noise).images[0]
```

Figure 21. Example inference code with NPNet on SDXL. Our NPNet operates as a plug-and-play module, which can be easily implemented.

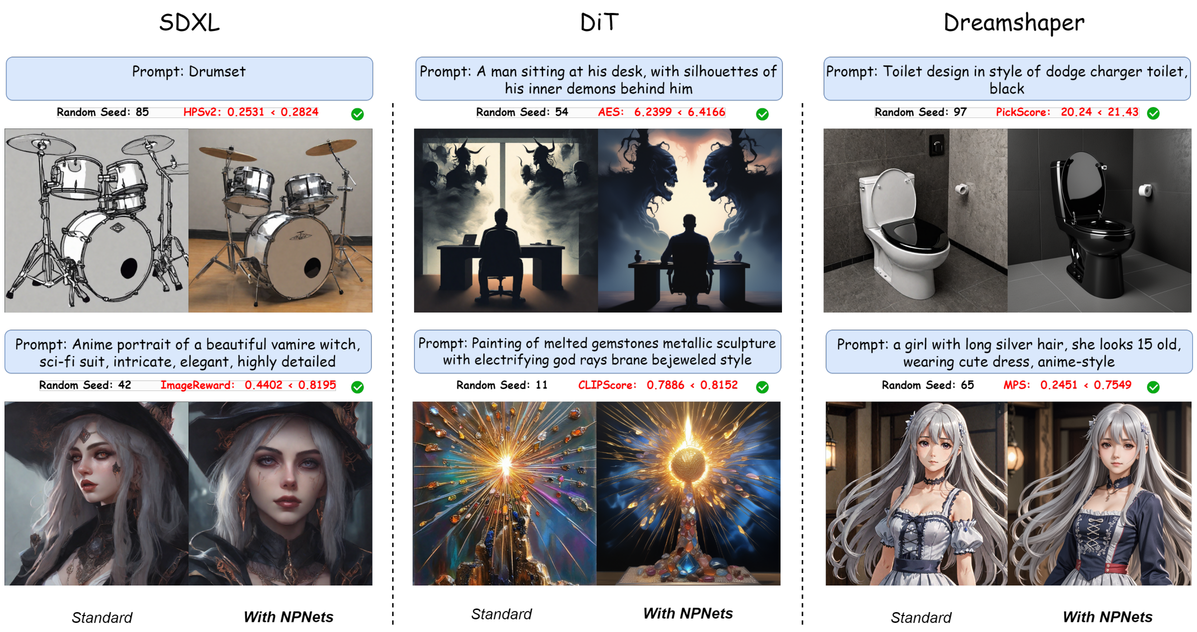

SDXL
DiT
Dreamshaper
Prompt: Drumset
Random Seed: 85 HPSv2: 0.2531 < 0.2824
Prompt: A man sitting at his desk, with silhouettes of his inner demons behind him
Random Seed: 54 AES: 6.2399 < 6.4166
Prompt: Toilet design in style of dodge charger toilet, black
Random Seed: 97 PickScore: 20.24 < 21.43
Prompt: Anime portrait of a beautiful vamire witch, sci-fi suit, intricate, elegant, highly detailed
Random Seed: 42 ImageReward: 0.4402 < 0.8195
Prompt: Painting of melted gemstones metallic sculpture with electrifying god rays brane bejeweled style
Random Seed: 11 CLIPScore: 0.7886 < 0.8152
Prompt: a girl with long silver hair, she looks 15 old, wearing cute dress, anime-style
Random Seed: 65 MPS: 0.2451 < 0.7549
Standard
With NPNets
Standard
With NPNets
Standard
With NPNets

Figure 22. We visualized images synthesized by 3 different diffusion models and evaluated them using 6 human preference metrics. Images for each prompt are synthesized using the same random seed. These images with NPNet demonstrated a noticeable improvement in overall quality, aesthetic style, and semantic faithfulness, along with numerical improvements across all six metrics. More importantly, our NPNet is applicable to various diffusion models, showcasing strong generalization performance with broad application potential.

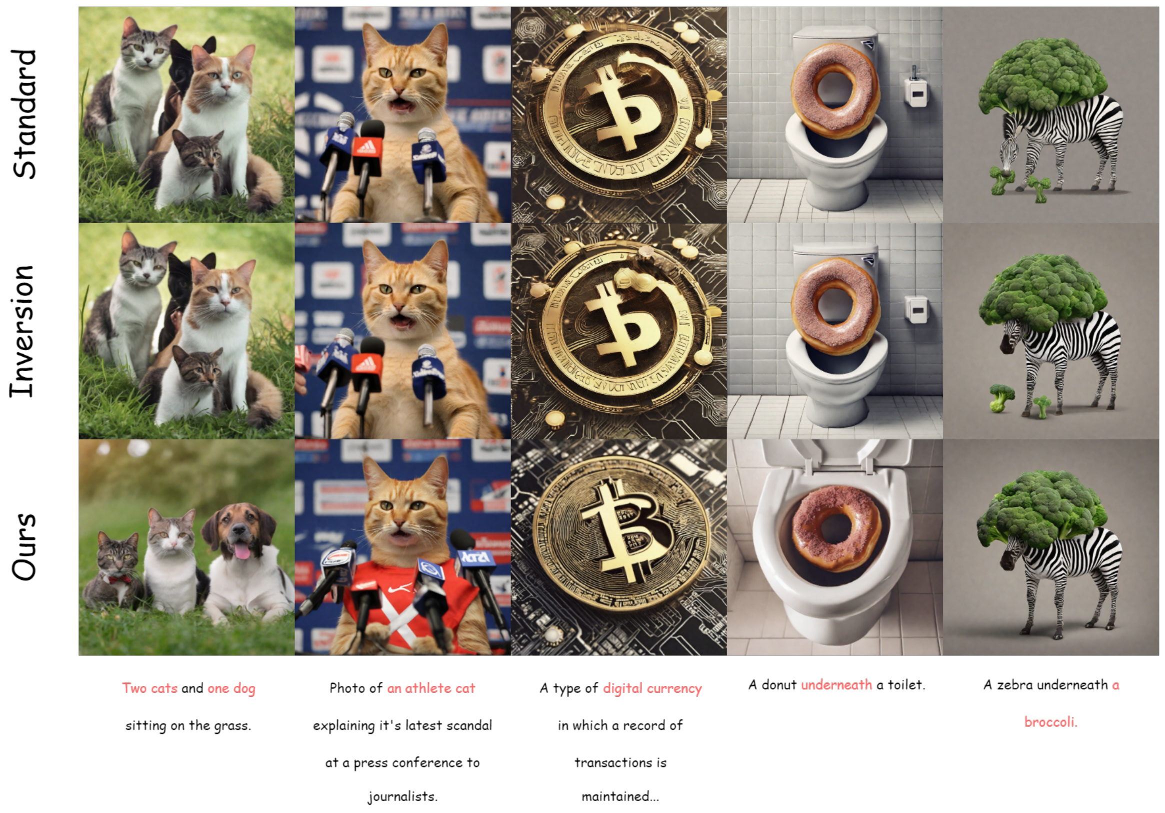
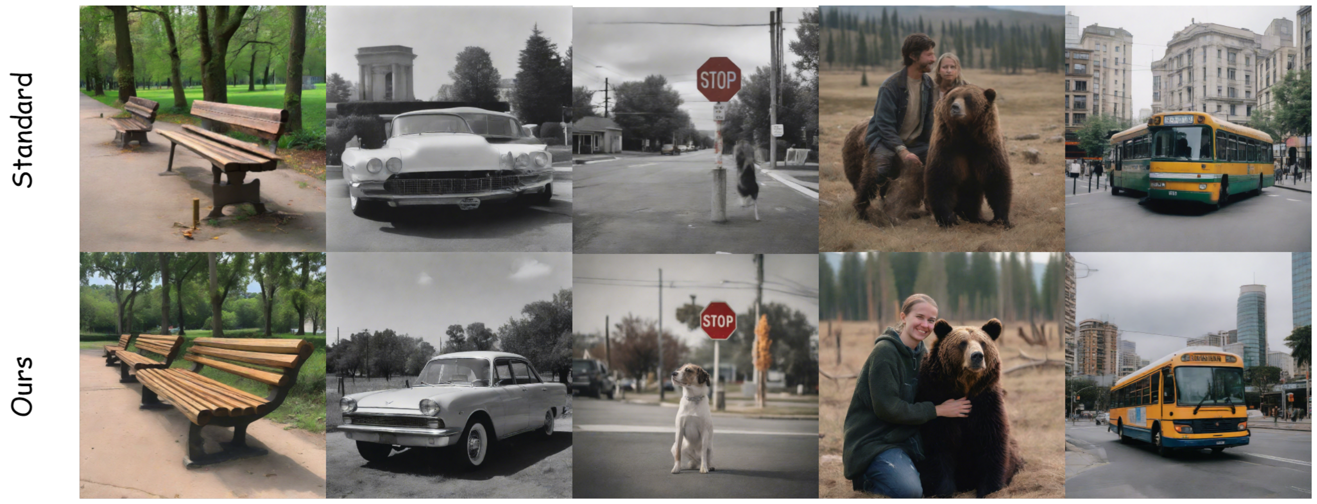
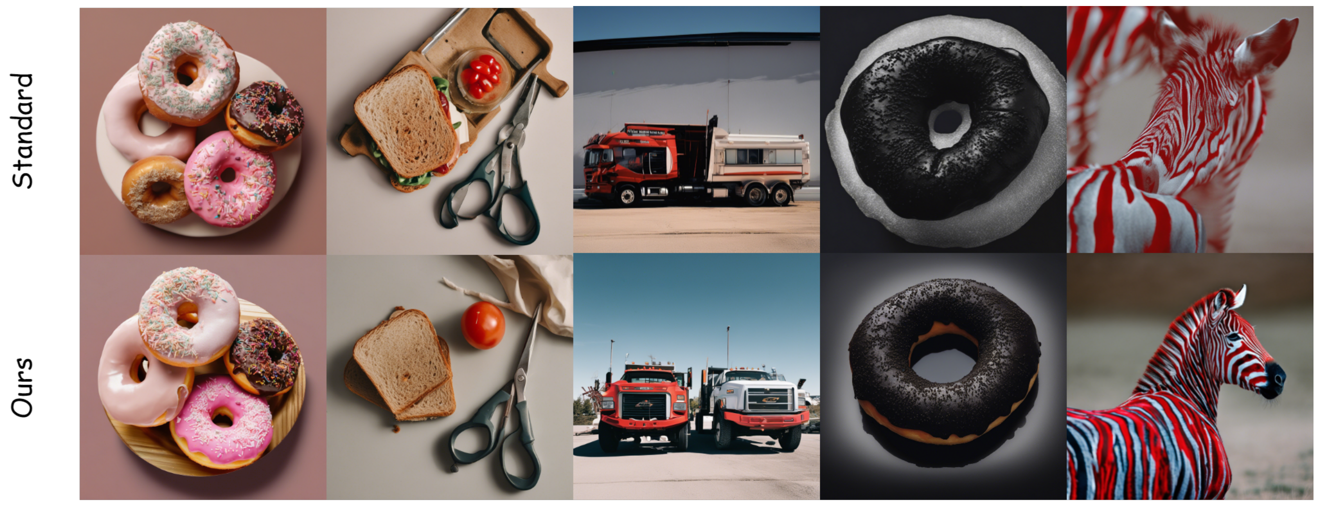

Standard
Inversion
Ours
Two cats and one dog sitting on the grass.
Photo of an athlete cat explaining it's latest scandal at a press conference to journalists.
A type of digital currency in which a record of transactions is maintained...
A donut underneath a toilet.
A zebra underneath a broccoli.

Figure 23. Visualization results about different methods on SDXL.



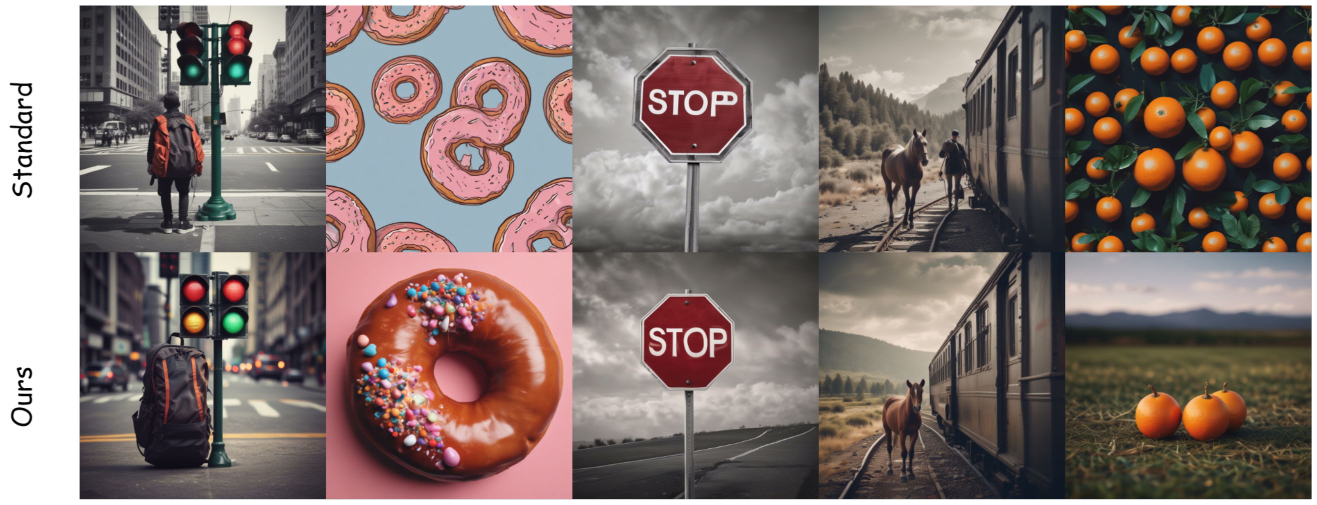
A photo of three benchs. A photo of a car. A photo of a stop sign and a dog. A photo of a person and a bear. A photo of a bus.

Figure 24. Qualitative comparison on LCM.



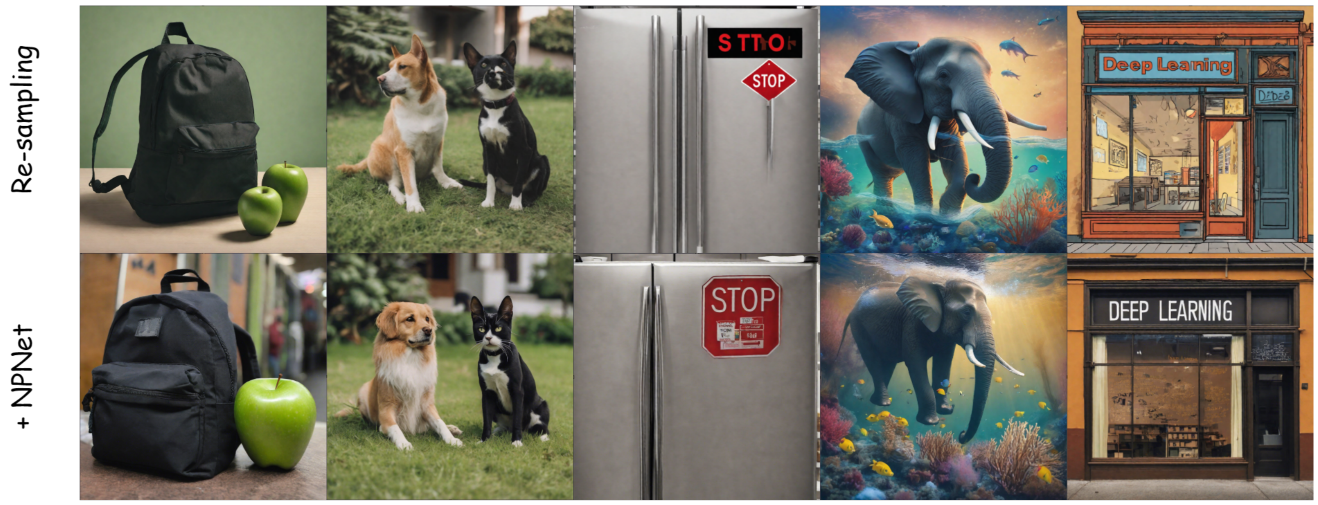
A photo of four donuts. A photo of a scissors, tomato and a sandwich. A photo of two trucks. A photo of a black donut. A photo of a red zebra.

Figure 25. Qualitative comparison on PCM.



A photo of a traffic light and a backpack. | A photo of a donut. | A photo of a stop sign. | A photo of a horse and a train. | A photo of three oranges.

Figure 26. Qualitative comparison on SDXL-Lightning.



A green apple and a black backpack | One cat and one dog sitting on the grass. | A stop sign on the right of a refrigerator. | An elephant under the sea. | A storefront with "Deep Learning" written on it.

Figure 27. Qualitative comparison on Re-sampling.

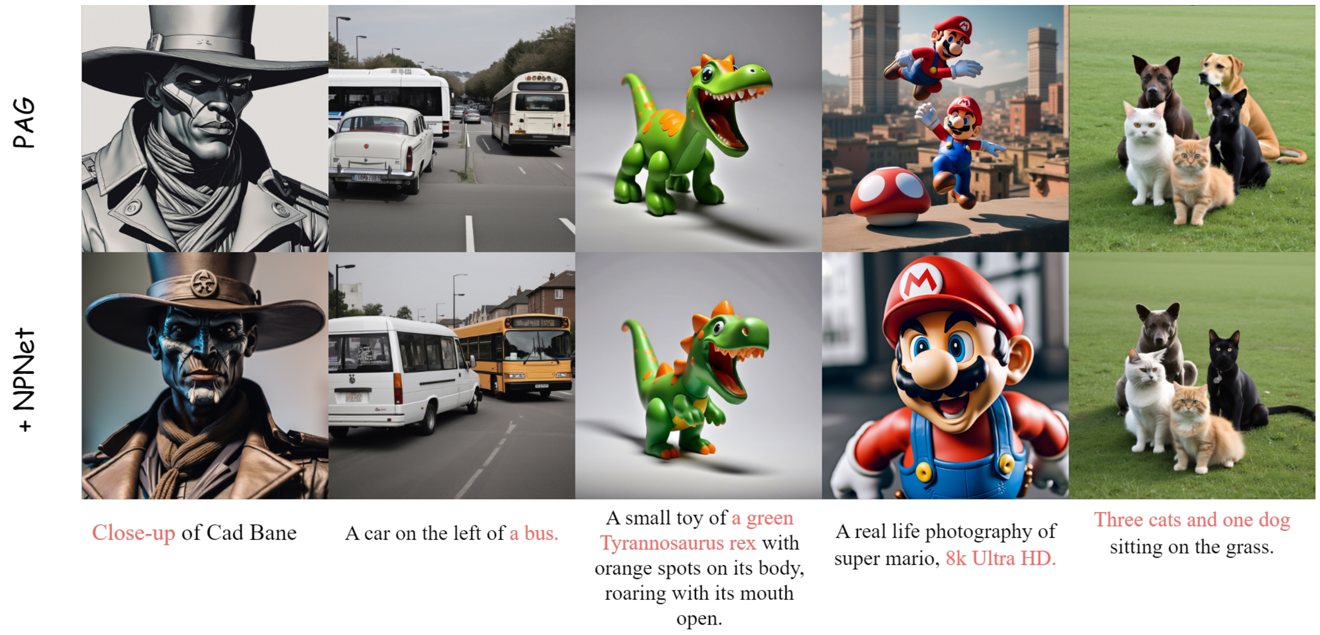

PAG
+ NPNet
Close-up of Cad Bane
A car on the left of a bus.
A small toy of a green Tyrannosaurus rex with orange spots on its body, roaring with its mouth open.
A real life photography of super mario, 8k Ultra HD.
Three cats and one dog sitting on the grass.

Figure 28. Qualitative comparison on PAG.

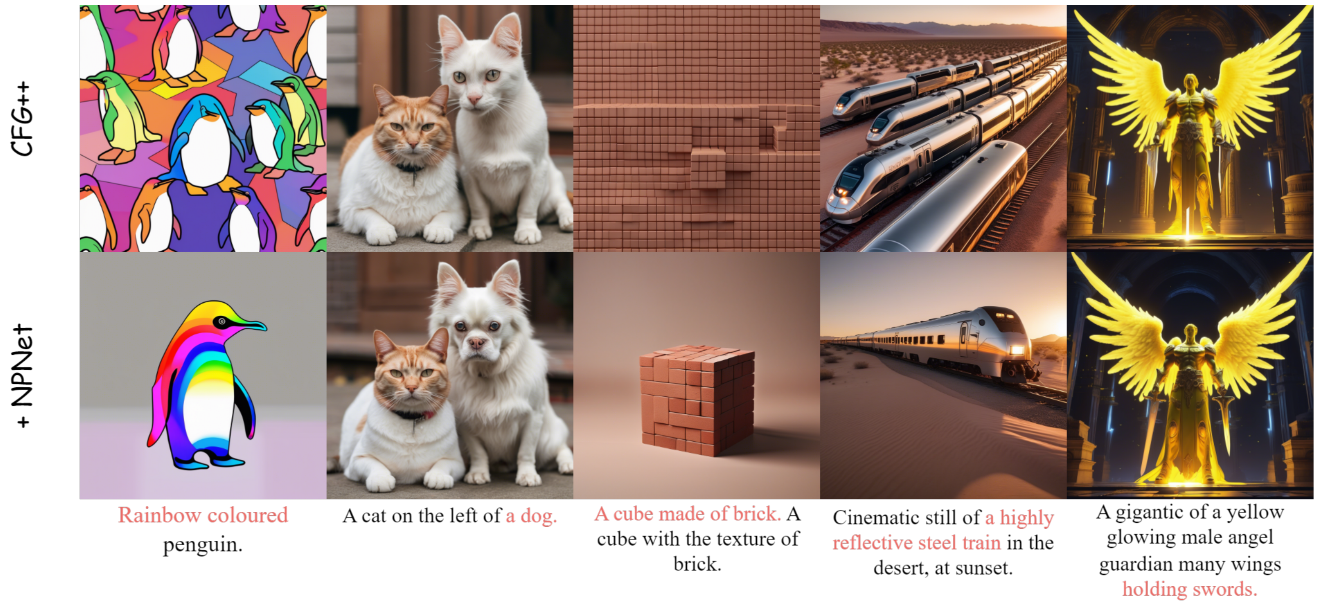
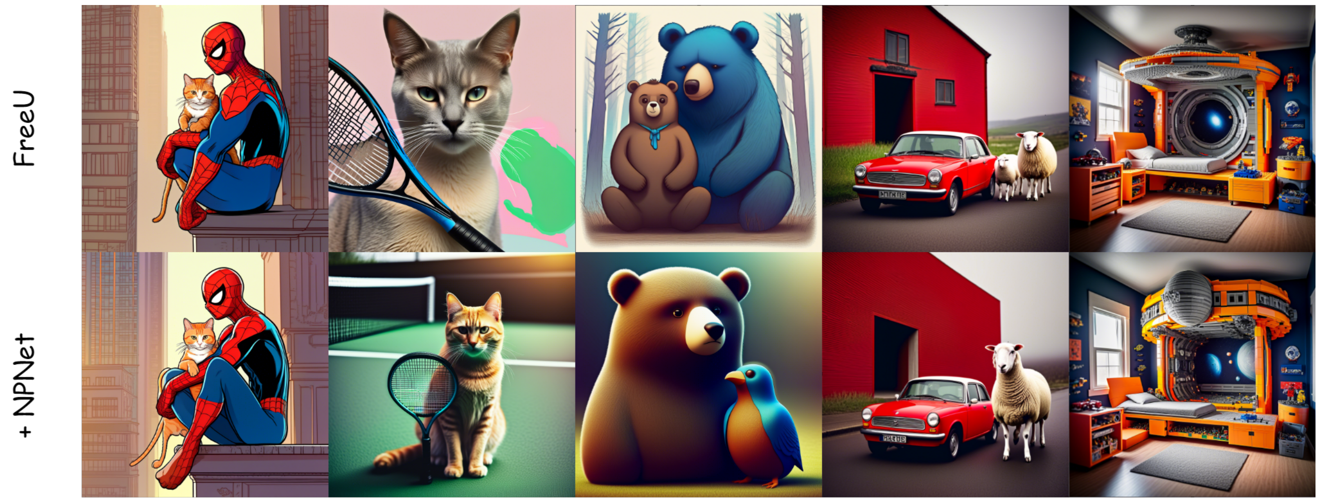
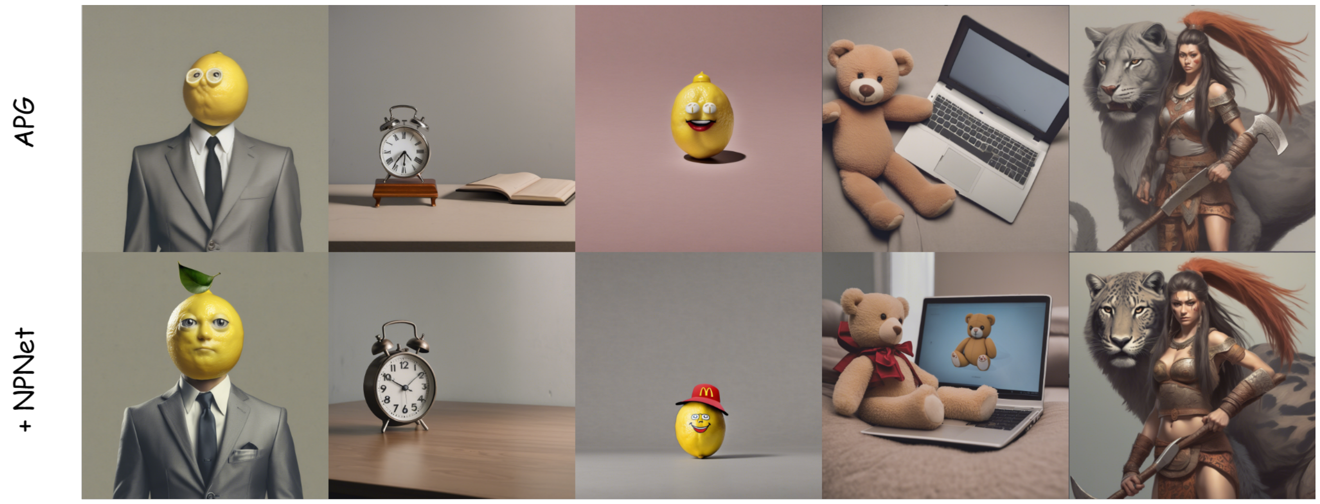

CFG++
+ NPNet
Rainbow coloured penguin.
A cat on the left of a dog.
A cube made of brick. A cube with the texture of brick.
Cinematic still of a highly reflective steel train in the desert, at sunset.
A gigantic of a yellow glowing male angel guardian many wings holding swords.

Figure 29. Qualitative comparison on CFG++.



Spider-Man holding a ginger cat. | A cat on the right of a tennis racket. | A blue bird and a brown bear. | A red car and a white sheep. | Photo of a mega Lego space station inside a kid's bedroom.

Figure 30. Qualitative comparison on FreeU.



A lemon wearing a suit and tie, | A single clock is sitting on a table. | A lemon with a McDonald's hat. | A teddy bear on top of a laptop. | Warrior holds an axe with a exotic animal with long hair style loisel.

Figure 31. Qualitative comparison on APG.

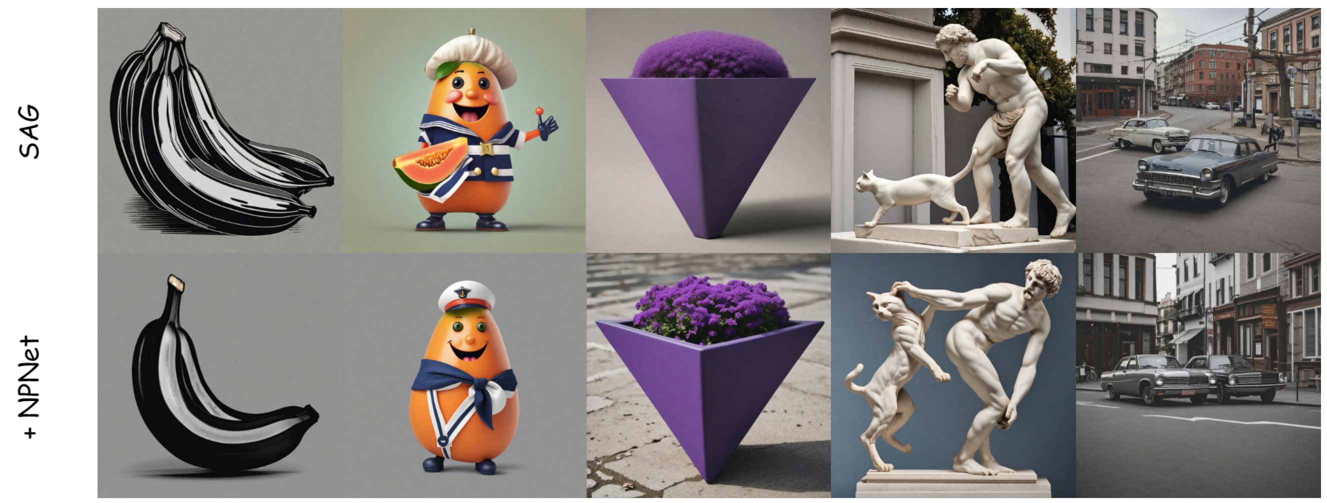

SAG
+ NPNet
A black colored banana.
A papaya fruit dressed as a sailor.
A triangular purple flower pot. A purple flower pot in the shape of a triangle.
Greek statue of a man with a cat.
Two cars on the street.

Figure 32. Qualitative comparison on SAG.

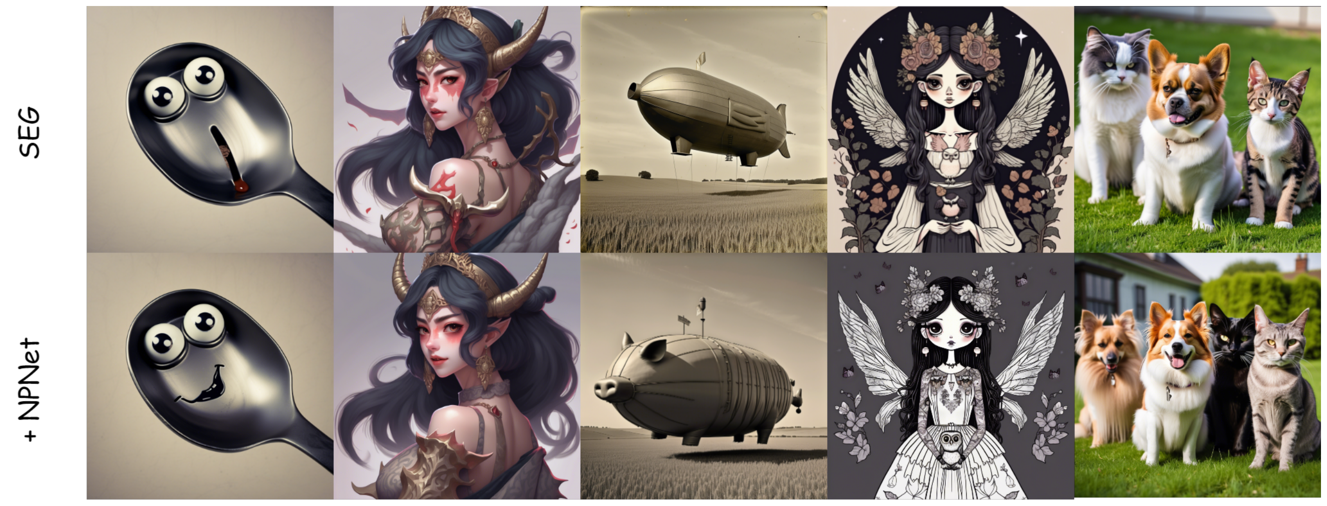

SEG
+ NPNet
A spoon dressed up with eyes and a smile.
An anime-style demon princess is depicted in a digital painting.
An old photograph of a 1920s airship shaped like a pig, floating over a wheat field.
Dark fairy with an owl in Cottagecore style
Two cats and two dogs sitting on the grass.

Figure 33. Qualitative comparison on SEG.