The Silent Assistant: *NoiseQuery* as Implicit Guidance for Goal-Driven Image Generation

Supplementary Material

Contents

Overview

In Appendix A, we introduce the generative posterior concept, emphasizing its consistency across different models and samplers. Moreover, Appendix B provides a detailed analysis of high-level semantic generation results, exploring our zero-shot transferability, complex image generation, and the effects of CFG scales. Lastly, implementation details for experiments on low-level visual properties are outlined in Appendix C, along with the noise offset method for color tendency control.

A. Analysis of Generative Posteriors

In this work, we define the generative posterior as the deterministic output of DDIM sampling from a specific initial noise with a NULL text prompt. As analyzed in Sec. 3.3, generative posteriors reveal the hidden features of the initial noise and exhibit model-agnostic behavior.

To further investigate the role of samplers, we compare generative posteriors from the same initial noise across multiple models (Stable Diffusion [42] v1.4, v1.5, v2.0, v2.1, SD-turbo [48] and PixArt-α [6]) and samplers (DDIM [49], LMS, Heun, Euler, PNDM [34], UniPC [67], DPM2 [26], DPM++ 2M [35], and DPM++ 2M Karras[26, 35]). As illustrated in Fig. S2, the generated images display remarkable consistency across different models and sampling strategies. This universal consistency enables the creation of a model-agnostic noise library, where each noise sample is associated with specific latent features and can be seamlessly reused across models and samplers. These findings underline the versatility of leveraging initial noise in a wide range of generative tasks.

Model	Noise	PickScore	CLIPScore
SD v2.1 → SD v1.4	Random	21.40	31.16
	NoiseQuery	**21.47**	**31.26**
SD v2.1 → SD v1.5	Random	21.41	31.08
	NoiseQuery	**21.49**	**31.29**
SD v2.1 → SD v2.0	Random	21.63	31.60
	NoiseQuery	**21.68**	**31.75**
SD v2.1 → PixArt-α	Random	22.24	31.48
	NoiseQuery	**22.32**	**31.67**
SD v2.1 → SD-Turbo	Random	22.07	31.51
	NoiseQuery	**22.19**	**31.70**

Table S1. Zero-shot transferability from SD v2.1 [42] to various generative models on MSCOCO [33] using BLIP features as semantic query. *NoiseQuery* outperforms random noise consistently, proving the initial noise is a universal implicit assistant.

Scheduler	Method	ImageReward↑	PickScore↑	HPS v2↑	CLIPScore↑
PDNM	SD v2.1 / +NoiseQuery	0.10 / **0.26**	21.31 / **21.42**	24.71 / **25.23**	30.91 / **31.67**
DDIM	SD v2.1 / +NoiseQuery	0.12 / **0.26**	21.30 / **21.45**	24.72 / **25.17**	31.18 / **31.71**
LMS	SD v2.1 / +NoiseQuery	0.07 / **0.27**	21.31 / **21.48**	24.66 / **25.37**	30.83 / **31.76**
Heun	SD v2.1 / +NoiseQuery	0.09 / **0.27**	21.34 / **21.48**	24.66 / **25.29**	30.93 / **31.23**
Euler	SD v2.1 / +NoiseQuery	0.12 / **0.22**	21.35 / **21.38**	24.61 / **25.12**	31.23 / **31.54**
DPM2	SD v2.1 / +NoiseQuery	0.09 / **0.27**	21.31 / **21.42**	24.66 / **25.37**	30.81 / **31.79**
DPM++ 2M Karras	SD v2.1 / +NoiseQuery	0.10 / **0.27**	21.36 / **21.44**	24.77 / **25.32**	30.83 / **31.79**
DPM++ 2M SDE Karras	SD v2.1 / +NoiseQuery	0.23 / **0.28**	21.42 / **21.59**	23.32 / **25.68**	31.20 / **31.23**

Table S2. Zero-shot transferability from DDIM to various samplers on DrawBench [46].

B. Extended Analysis of High-Level Semantics

B.1. Zero-Shot Transferability

Our findings show that implicit information in initial noise remains consistent across different models, regardless of their architectures, as discussed in Sec. 3.3. To validate this, we directly apply our noise library built on Stable Diffusion v2.1 to various different models for generation in a zero-shot manner. As shown in Tab. S1, our approach consistently improves performance across all models, demonstrating its robustness and broad applicability. This highlights the generalizability of our method, enabling seamless integration with any diffusion model.

Additionally, NoiseQuery provides a goal-aligned initial noise while users can freely choose various schedulers for generation. As shown in Tab. S2, a library built with DDIM generalizes well to other schedulers.

B.2. Complex Image Generation.

As shown in Fig. S1, our method improves the model's performance in challenging scenarios like visual text generation, object interaction, object composition, and spatial

Figure S1. Results on challenging scenarios, showcasing how our method enhances performance in complex image generation tasks. The example shown uses CLIP features for semantic queries.

relationship, while reducing attribute leakage. For example, when generating a "blue dog", a random noise might cause the model to adaptively infuse the blue into the entire scene. This is because the model tries to enforce semantic consistency globally, which can destabilize the generation process, leading to unintended attribute spillover. In contrast, our approach selects noise that already contains target features (e.g., objects, layouts), minimizing the need for global adjustments. This preserves object attributes, enhances semantic consistency, and prevents attribute leakage.

B.3. Different CFG Scale Analysis

We provide a comprehensive analysis of our method's performance across various classifier-free guidance (CFG) scales on 10k MSCOCO prompts in Fig. 7. Remarkably, even at low CFG scales, our approach achieves comparable or superior results compared to the baseline at much higher scales. This demonstrates that the selected noise samples effectively align with text prompts, reducing generation difficulty and eliminating the need for excessively high guidance scales that often cause over-saturation and instability.

To complement the quantitative analysis, Fig. S3 visualizes the generated images across different CFG scales. Our method consistently produces semantically accurate and visually coherent outputs even at low scales, while random noise frequently results in failed generations. Even at higher CFG scales, the baseline struggles to maintain stable semantic alignment, whereas our approach achieves robust and reliable performance across most scales.

C. Implementation Details of Low-Level Visual Properties

C.1. Texture

Texture refers to the small-scale spatial patterns of intensity variation in an image, which can be used to characterize the surface properties of objects. To control texture in generated images, we utilize the Gray-Level Co-occurrence Matrix (GLCM) [19], a commonly used technique for quantifying texture in terms of statistical measures. The GLCM features capture the spatial arrangement and frequency of pixel intensity changes, providing a texture profile for each image.

For texture-based noise querying, we compare the GLCM features of the reference image with those stored in the noise library. The noise with the closest texture profile (using a distance metric like Euclidean distance) is selected to generate the output image.

C.2. Shape

Shape refers to the geometric form or structure of objects in an image, independent of texture or color. To control shape in generated images, we use Hu Moments [22], which are invariant shape descriptors that capture the overall geometry of an object, regardless of its size, position, or orientation. These moments are derived from the image's spatial distribution of intensity and provide a compact, rotation-invariant representation of the object's shape.

For shape-based noise querying, the similarity between the reference and each noise sample is measured using Euclidean distance between their Hu Moments vectors. The noise sample with the closest similarity to the reference shape is selected to ensure the generated image maintains the desired shape.

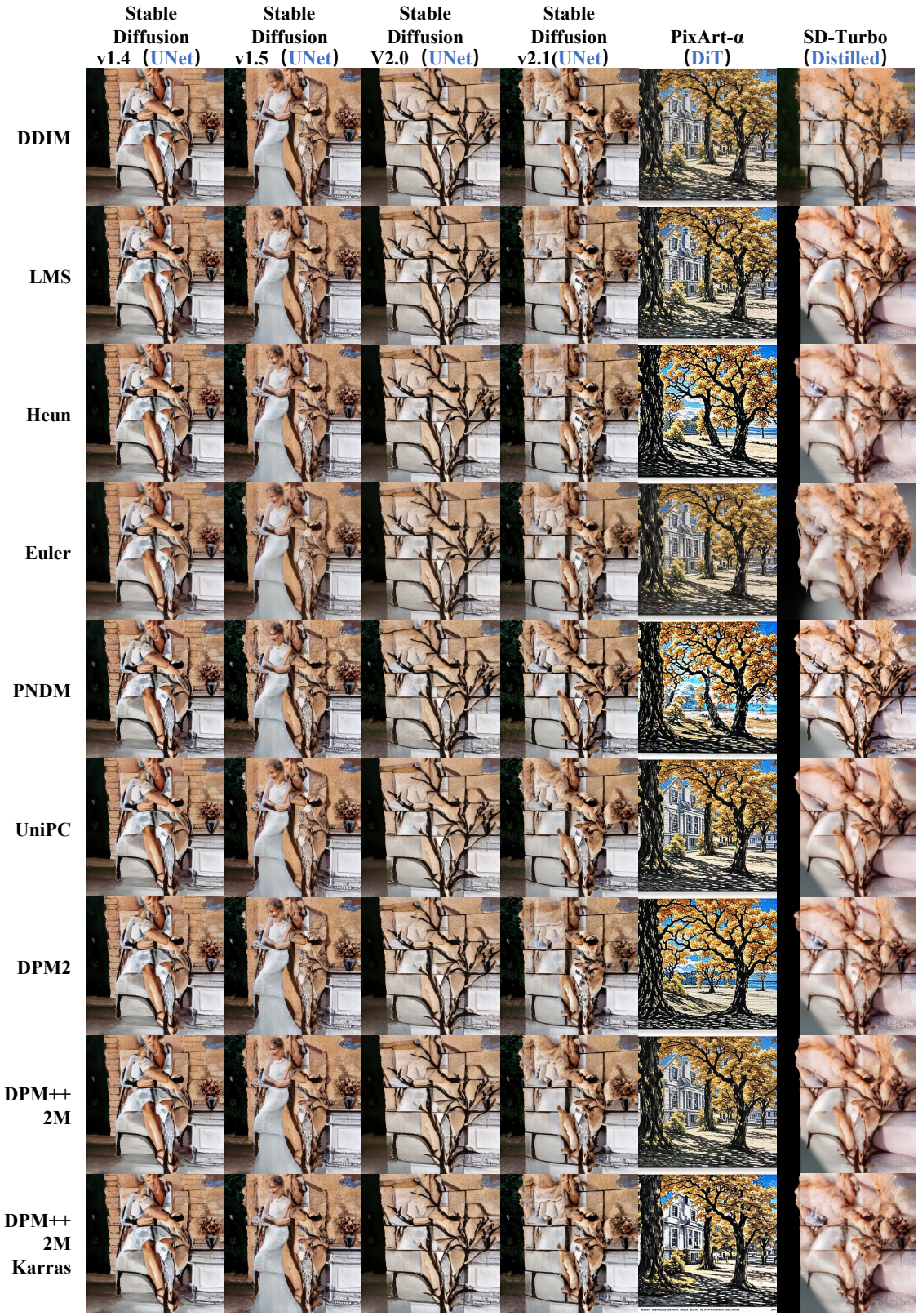

Figure S2. Generative posteriors obtained from the same initial noise using different models (columns) and samplers (rows).

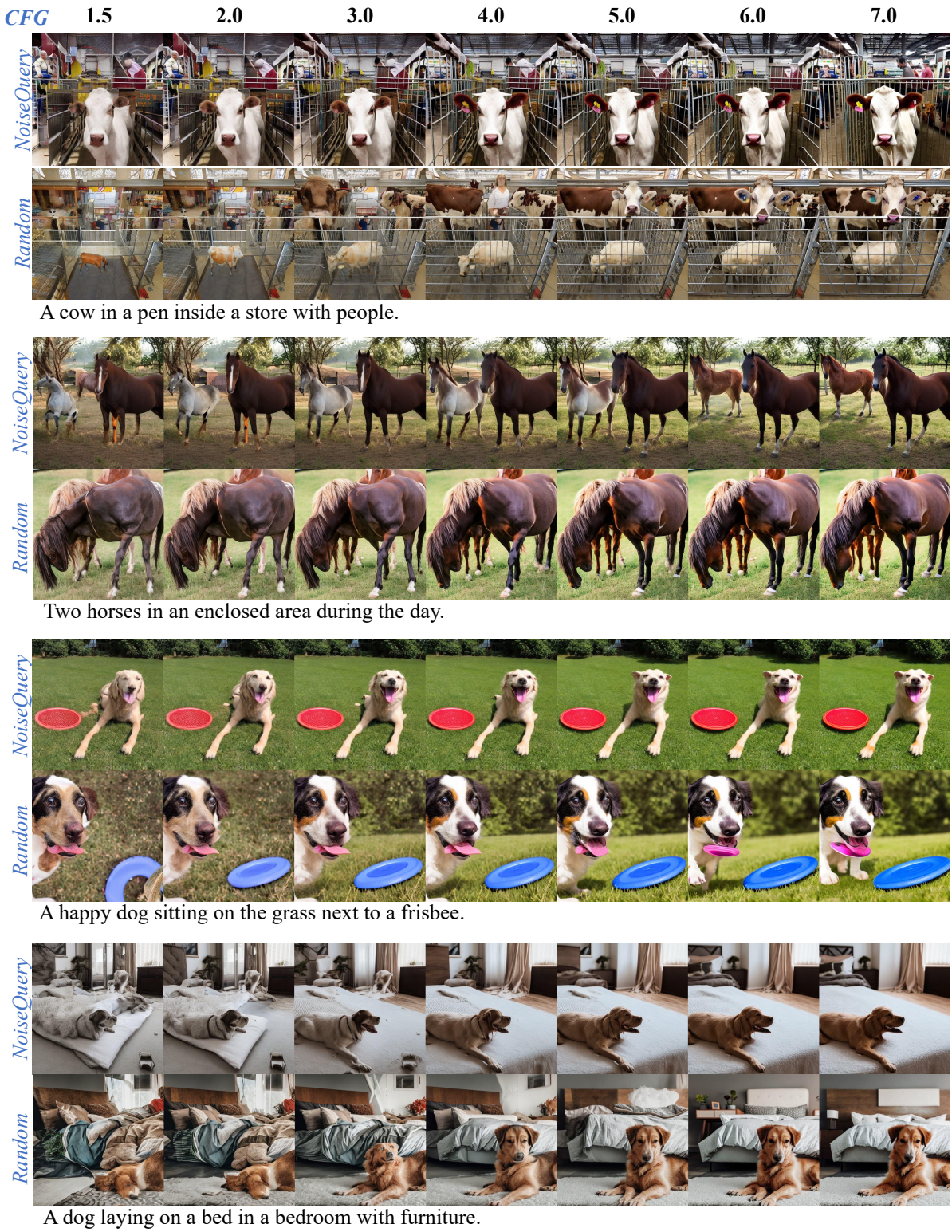

Figure S3. Visual comparison of generated images at different CFG scales (1.5, 2.0, ..., 7.0). Our method produces semantically accurate and visually coherent images at low scales, while random noise often fails to generate meaningful or semantically consistent outputs. This highlights our method's ability to reduce generation difficulty and improve stability.

Algorithm 1: Noise Offset for Controlled Color Shifting

Input: Diffusion model $\mathcal{M}$; Initial noise ϵ; Adjustment parameters δ (e.g., brightness, saturation); Color adjustment function $S(\cdot, \delta)$; DDIM inversion process DDIM-Inverse$(\cdot, T)$;

Output: Modified noise ϵ^*

begin

$\mathcal{I}_{\text{uncond}} \leftarrow \mathcal{M}(\epsilon, T, c = \emptyset)$;

$\mathcal{I}_{\text{adjusted}} \leftarrow S(\mathcal{I}_{\text{uncond}}, \delta)$;

$\epsilon^* \leftarrow$ DDIM-Inverse$(\mathcal{I}_{\text{adjusted}}, T)$;

return ϵ^* ;

end

Figure S4. Unlike the original Stable Diffusion, which produces images with medium brightness, our *NoiseQuery* (offset) expands the range to include both very bright and very dark samples.

C.3. Sharpness

Sharpness is a key low-level visual property that relates to the level of detail and clarity in an image, particularly the prominence of high-frequency components, such as edges and fine textures. To control sharpness in the generated images, we focus on measuring High-Frequency Energy (HFE). HFE quantifies the amount of high-frequency content (i.e., fine details and sharp edges) in an image. Higher HFE corresponds to sharper, more detailed images, while lower HFE indicates softer, blurrier images. The noise sample with the higher HFE (based on sorting) is chosen, ensuring the generated image has the desired sharpness.

C.4. Noise Offset for Controlled Color Shifting

Beyond directly selecting noise, we also propose a method to introduce subtle color shifts by adjusting generative posteriors, preserving semantic content while allowing controlled color variations. We show the pseudo-code in Algorithm 1 Specifically, we first generate an unconditional image from the initial noise and then apply subtle color adjustments, such as changes in brightness or saturation. After the color changes are made, we reverse the adjusted image back into the noise space via DDIM inversion. This results in a modified noise that consistently carries the desired color shifts, ensuring uniform color variations across different prompts while maintaining the underlying semantic content.

This helps address the limitation of SD [42] in generating very bright or dark images, caused by noise distribution discrepancies during training and inference [32]. As shown in Fig. S4, subtly adjusting the brightness of generative posteriors can shift the inherent tendency of the initial noise, enabling the generation of images with a wider range of brightness during inference. shifts the noise distribution, enabling a wider range of brightness in generated images.