arXiv:2508.10382v2 [cs.CV] 26 Nov 2025

Towards Spatially Consistent Image Generation: On Incorporating Intrinsic Scene Properties into Diffusion Models

Hyundo Lee[1], **Suhyung Choi**[1], **Inwoo Hwang**[2†], **Byoung-Tak Zhang**[1†]
[1]AI Institute, Seoul National University
[2]Columbia University
{hdlee, schoi}@bi.snu.ac.kr, inwoo.hwang@columbia.edu, btzhang@bi.snu.ac.kr

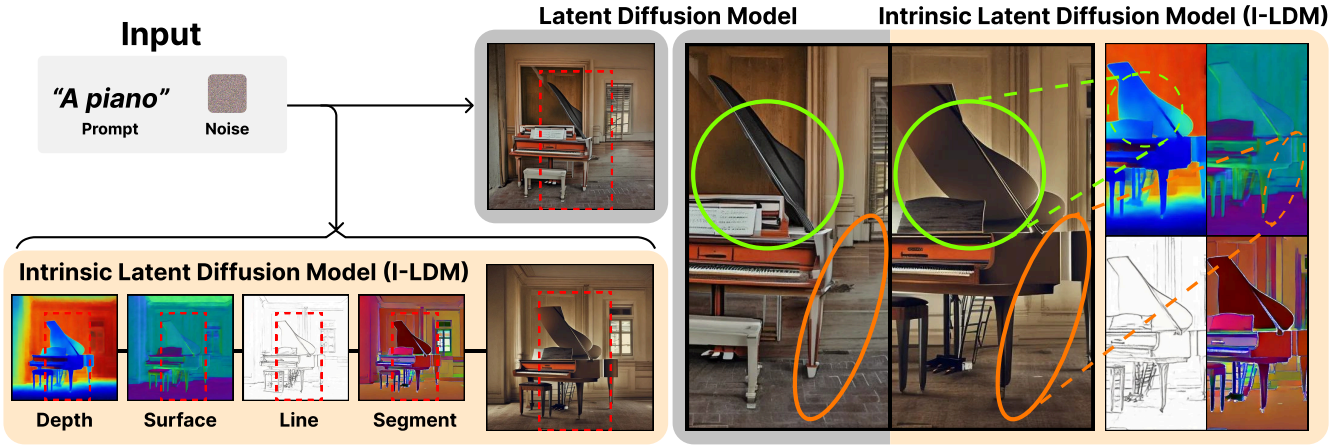


Figure 1: By co-generating images and aligned intrinsic scene properties, we aim to address the problem of *spatial inconsistency* prevalent in existing text-to-image models. **(Gray)** Paradoxical image generated by Stable Diffusion 2.1, including inconsistent wall, object (green circle), and floor (orange circle). **(Beige)** Our approach generates an image and intrinsic scene properties representing the scene from diverse perspectives, thereby producing a more natural and realistic image.

Abstract

Image generation models trained on large datasets can synthesize high-quality images but often produce spatially inconsistent and distorted images due to limited information about the underlying structures and spatial layouts. In this work, we leverage intrinsic scene properties (e.g., depth, segmentation maps) that provide rich information about the underlying scene, unlike prior approaches that solely rely on image-text pairs or use intrinsics as conditional inputs. Our approach aims to co-generate both images and their corresponding intrinsics, enabling the model to implicitly capture the underlying scene structure and generate more spatially consistent and realistic images. Specifically, we first extract rich intrinsic scene properties from a large image dataset with pre-trained estimators, eliminating the need for additional scene information or explicit 3D representations. We then aggregate various intrinsic scene properties into a single latent variable using an autoencoder. Building upon pre-trained large-scale Latent Diffusion Models (LDMs), our method simultaneously denoises the image and intrinsic domains by carefully sharing mutual information so that the image and intrinsic reflect each other without degrading image quality. Experimental results demonstrate that our method corrects spatial inconsistencies and produces a more natural layout of scenes while maintaining the fidelity and textual alignment of the base model (e.g., Stable Diffusion).

1 Introduction

Recent text-to-image (T2I) generation models, notably diffusion models (Ho, Jain, and Abbeel 2020; Dhariwal and Nichol 2021; Saharia et al. 2022; Rombach et al. 2022), have shown remarkable success in producing diverse and realistic images. However, these models often produce images with *spatial inconsistencies*—such as distorted object geometry or implausible scene layouts as in Fig. 1—due to their reliance on pixel-level supervision and lack of structured scene understanding (Podell et al. 2023). To overcome this, richer and more structured representations beyond pixels are needed to guide the generation process to generate more spatially coherent images.

A natural remedy is to incorporate explicit 3D scene information (e.g. point clouds, meshes) into the generative process (Luo and Hu 2021; Jun and Nichol 2023). While effective in grounding geometry, the high cost of data acquisition

†Corresponding authors.

and rendering limits its scalability and diversity. Additionally, the reliance on explicit 3D structure makes them ill-suited for abstract, stylized, or artistic scenes. An alternative approach is to leverage intrinsic scene properties (or simply *intrinsics*)—such as depth, surface normals, or segmentation maps—which offer structured, complementary views of the same scene. Here, we use the term *intrinsics* in the sense of Barrow et al. (1978), referring to image-inherent scene attributes rather than camera intrinsics in multi-view geometry. However, prior work leveraging intrinsics has not aimed to improve spatial consistency, but rather focused on conditional generation given external intrinsic inputs (Zhang, Rao, and Agrawala 2023). This reliance on additional signals limits their applicability to general T2I generation and fails to model the underlying scene structure.

To address this issue, we propose a new approach that improves spatial consistency by jointly modeling both images and their intrinsic scene properties. As illustrated in Fig. 1, intrinsics provide complementary geometric and semantic cues that help delineate object boundaries and spatial layouts, especially in perceptually ambiguous regions. By learning a joint distribution over images and intrinsics, our model captures multiple structured views of a scene, thereby significantly reducing spatial inconsistencies in the generation process. In contrast to conditional approaches that model a unidirectional relationship, our joint modeling allows intrinsics and images to regularize each other during training. Incorporating intrinsics also offers practical advantages: they can be efficiently extracted from large-scale 2D datasets using pre-trained models, without the need for 3D data or manual labeling. Moreover, since large T2I models implicitly contain intrinsic knowledge (Du et al. 2023; Zhan et al. 2023; El Banani et al. 2024), we leverage this capacity with only minimal adjustments, without compromising the quality and capabilities of the original model.

To this end, we propose Intrinsic Latent Diffusion Models (I-LDM), which extend existing T2I models (Rombach et al. 2022; Podell et al. 2023; Chen et al. 2024b) to jointly generate images and their corresponding intrinsics. To enable efficient joint generation, we first train an autoencoder that encodes multiple intrinsic channels into a single latent space. Since naive fine-tuning might compromise the performance of the base model, we retain the architecture and weights for generating images while adopting LoRA weights (Hu et al. 2021) to generate intrinsics separately. Furthermore, we introduce a cross-domain weight scheduling mechanism that shares self-attention between the image and intrinsic domains during the denoising process, promoting alignment without introducing visual artifacts.

By learning from both perceptual and structural views of a scene, I-LDM produces more faithful and structured images without compromising the visual fidelity of strong base models. We evaluate our model under general text prompts and challenging multi-object arrangements, showing that I-LDM produces clearer separation and more realistic spatial layouts. We show that I-LDM generates more realistic images and achieves higher scores in human preference estimators while maintaining the fidelity and text alignment of the base model. Furthermore, we demonstrate that learning to co-generate intrinsics helps to model complex structures by providing consistent structural guidance in dynamic hand pose generation tasks.

2 Related work

2.1 Large-scale Diffusion Models

Recently, image synthesis has achieved remarkable results based on diffusion models (Esser et al. 2024; Sauer et al. 2024; Chen et al. 2024b), which generate images by denoising random noise (Sohl-Dickstein et al. 2015; Song and Ermon 2019; Ho, Jain, and Abbeel 2020). Diffusion models have evolved to enable natural language-based control (Saharia et al. 2022; Nichol et al. 2021; Ho and Salimans 2022) using language models such as CLIP (Radford et al. 2021). Latent diffusion models (LDMs) (Rombach et al. 2022) apply perceptual compression via VQ-VAE (Razavi, Van den Oord, and Vinyals 2019) to learn the denoising process in perceptually compressed latent space, facilitating high-resolution image synthesis. Models such as DALL-E (Shi et al. 2020) or Stable Diffusion (Rombach et al. 2022; Podell et al. 2023) have been trained on large-scale text-image pair datasets containing billions of examples, allowing them to generate images conditioned on general natural language without restricting domains (Schuhmann et al. 2022). Our work is based on production-ready Stable Diffusion models, using fine-tuning techniques with LoRA (Hu et al. 2021) to maintain their quality and capability.

2.2 Reflecting Physical Structure in LDMs

Diffusion models trained on pairs of text images struggle to accurately capture the physical structures and spatial layout of scenes, occasionally producing distorted or inconsistent images (Podell et al. 2023; Shen et al. 2024). One solution is to incorporate 3D data (Luo and Hu 2021; Zhou, Du, and Wu 2021; Jun and Nichol 2023), such as point clouds (Chang et al. 2015), CAD models (Wu et al. 2015), or polygonal meshes (Deitke et al. 2023, 2024). Despite effectively capturing geometry, this approach is limited to specific objects and struggles with general or abstract text prompts due to the high cost and inflexibility of 3D data. Instead, we leverage intrinsic scene properties to capture underlying structures while maintaining flexibility for diverse textual inputs.

In image generation, intrinsic scene properties (Barrow et al. 1978) have been widely used as conditional controls, including SDEdit (Meng et al. 2021), sketch-guided generation (Voynov, Aberman, and Cohen-Or 2023), and ControlNet (Zhang, Rao, and Agrawala 2023), focusing on aligning outputs to given intrinsic conditions. In contrast, we aim to co-generate intrinsics alongside images, enabling the model to learn and reflect the distribution of underlying scenes.

Recent studies also explore generating intrinsics from images using diffusion models, including depth maps and surface normals (Fu et al. 2025; Ji et al. 2023; Saxena et al. 2023; Qiu et al. 2024; Ke et al. 2024), segmentation maps (Amit et al. 2021; Li et al. 2023; Nguyen et al. 2024), and other scene properties (Du et al. 2023). However, our goal is not to reconstruct exact geometry, but to guide image

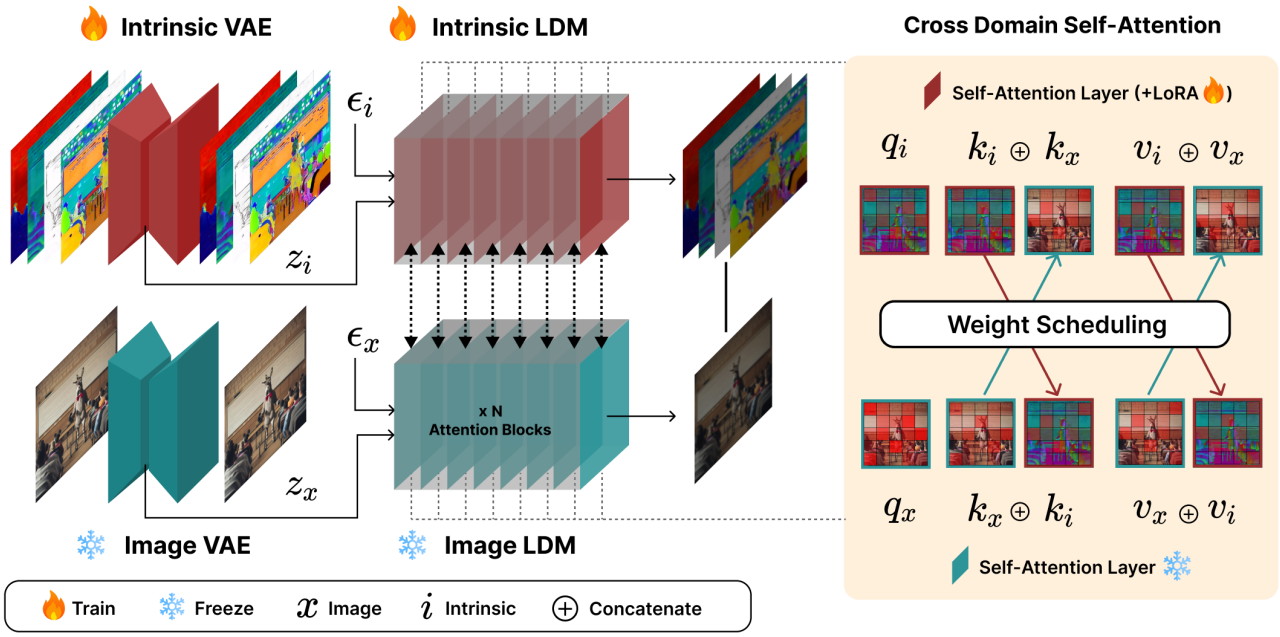


Figure 2: The overall architecture of I-LDM. **(Left)** We first train an intrinsic VAE to encode all intrinsics into a single latent variable. **(Middle)** Then, we train the LoRA weights of the self-attention layers included in the diffusion network of the intrinsic domain to learn the denoising process. **(Right)** We employ a weight scheduling mechanism for exchanging self-attention with the image domain. As a result, I-LDM simultaneously generates spatially consistent images and intrinsics during inference.

generation by jointly modeling intrinsic and image domains, using only text conditions.

A prior approach generates surface normals from noise using GANs and then conditions image synthesis on them (Wang and Gupta 2016). However, generating intrinsics without considering the image domain can lead to spatial inconsistencies, as they lack awareness of the full scene. Another related line of work is LDM3D (Stan et al. 2023), which generates depth maps aligned with images from text. However, their primary goal is to generate RGB-D images and 360° views. In contrast, we focus on enhancing spatial consistency by jointly modeling image and intrinsic representations during generation. We empirically compare our method with these prior approaches to validate the effectiveness in improving spatial consistency and image quality.

3 Method

In this section, we utilize intrinsics to reflect the underlying scene in generated images more faithfully. The theoretical motivation is discussed in Sec. A. We first present our model architecture, which incorporates cross-domain self-attention and a weight scheduling mechanism to co-generate images and aligned intrinsics without compromising image quality. We then introduce a practical approach for handling multiple intrinsics simultaneously.

3.1 Intrinsic Latent Diffusion Model

In this work, we consider four intrinsic scene properties: depth map, surface normal, segmentation map, and line drawing. A straightforward way to generate an image-intrinsic pair (x, i) given a text condition c is to augment pre-trained T2I models with intrinsic features—either by adding extra channels or using spatial addition—and fine-tune them. However, we found that this significantly harms the image quality, as demonstrated in Sec. 4.1.

Instead, we propose a method using a large LDM model consisting of self-attention layers (e.g., Stable Diffusion (Rombach et al. 2022; Podell et al. 2023), PixArt (Chen et al. 2024b)) as a base model that fully exploits its capabilities without compromising its quality. As depicted in Fig. 2, we devise separate models for the image domain and the intrinsic domain, preserving the architecture and weights the same as the base model for the image domain. To train the intrinsic domain, we adapt LoRA (Hu et al. 2021) to fine-tune only the attention layers, leveraging the inherent knowledge of the intrinsics (Du et al. 2023; Zhan et al. 2023).

Second, we employ cross-domain self-attention (Fu et al. 2025) to align the spatial arrangements between the two domains and allow them to reflect each other's information. Specifically, we concatenate the keys and values from the self-attention layers at the same stage as follows:

$$\begin{aligned} q_x &= Q_x \cdot z_x, \quad q_i = Q_i \cdot z_i, \\ k_x &= k_i = (K_x \cdot z_x) \oplus (K_i \cdot z_i), \\ v_x &= v_i = (V_x \cdot z_x) \oplus (V_i \cdot z_i), \end{aligned} \tag{1}$$

where the subscript x, i denotes each domain, z denotes intermediate representations, $\oplus$ denotes concatenation, q, k, v denotes the query, key, and value, and Q, K, V denotes the corresponding weight matrix.

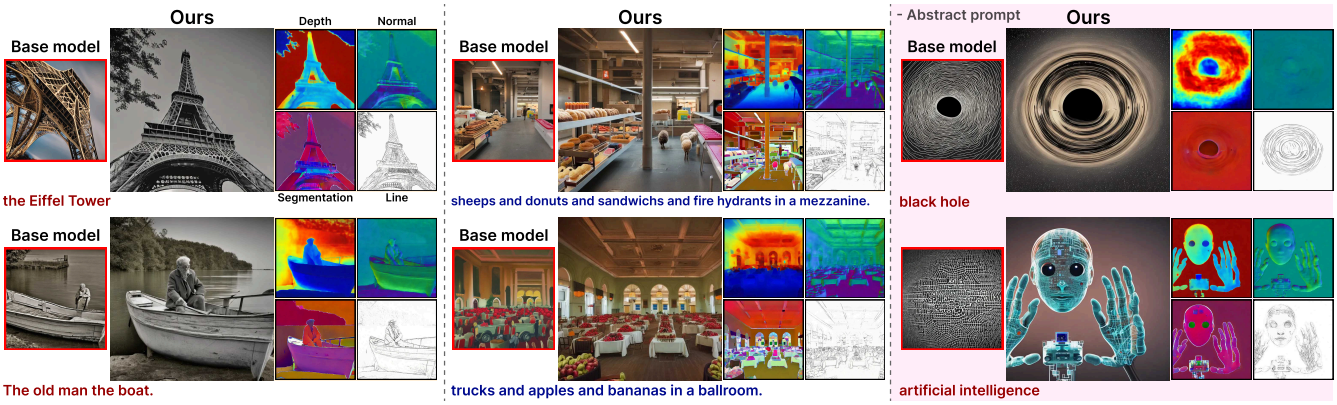


Figure 3: Qualitative analysis of generated images and co-generated intrinsic scene properties from I-LDM (clockwise from top left: depth map, surface normal, line drawing, and segmentation map). The red boxes indicate generated images of the base model. The top and bottom rows visualize samples with Drop and Gaussian weight scheduling, respectively. Red and blue captions denote samples from the Parti and Multi prompts, respectively.

Importantly, we introduce a weight scheduling mechanism for cross-domain self-attention, i.e., $w^{(l,t)}$, in contrast to prior work without weighting (Fu et al. 2025). This is because we found that the received information from the intrinsic model during image sampling often incurs undesirable artifacts in the generated image, leading to a significant degradation in image quality. Therefore, we introduce the scheduling of the weight of the exchanged key and values across domains for each block index l and each timestep t. Specifically, we fix the weight to 1 during training, while introducing two different weight scheduling strategies during sampling, Drop and Gaussian, as follows:

$$\text{(Drop)} \quad w_d^{(l,t)} = \begin{cases} 1 & \text{if} \quad (l \in L) \wedge (t \leq \tau), \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

$$\text{(Gaussian)} \quad w_g^{(l,t)} = \alpha \cdot \exp(-(t-\tau)^2/\sigma^2), \tag{3}$$

where L, τ, σ, α are hyperparameters. We then change the self-attention process as follows:

$$\begin{aligned} Attn_x^{(l,t)} &= \textit{softmax}\left(\frac{q_x \cdot k_x^T}{\sqrt{d_k}} + \log\left(W^{(l,t)}\right)\right) v_x, \\ Attn_i &= \textit{softmax}\left((q_i \cdot k_i^T)/\sqrt{d_k}\right) v_i, \end{aligned} \tag{4}$$

where d_k denotes the inner dimension of q, k, and $W^{(l,t)}$ denotes a $N \times 2N$ weight matrix where the first through N columns are filled with 1, and the $N{+}1$ through $2N$ columns are filled with $w^{(l,t)}$. The difference from the original image denoising process is that the intrinsic model affects the image model with a weight $w^{(l,t)}$ by passing its intermediate key and value vectors. We assign a low weight to the outer blocks of the diffusion network and for low t, where relatively fine-grained detail is denoised. We also assign low weight to very early steps of the denoising process, which greatly impacts the image. This adjustment allows us to mitigate the influence of artifacts on images, thereby preserving the quality of the generated images.

For the loss function, we use the conventional DDPM loss to train LoRA weights θ' in the network $\epsilon_{\theta,\theta'} = \left(\epsilon^{(x)}_{\theta,\theta'}, \epsilon^{(i)}_{\theta,\theta'}\right)$ for the image and intrinsic domains separately, with assigning a balancing factor of λ:

$$\begin{aligned} \mathcal{L} = \; & \mathbb{E}_{x_0, i_0, c, \epsilon_x, \epsilon_i, t}[||\epsilon^{(x)}_{\theta,\theta'}(x_t, i_t, t, c) - \epsilon_x||_2^2 \\ & + \lambda \cdot ||\epsilon^{(i)}_{\theta,\theta'}(x_t, i_t, t, c) - \epsilon_i||_2^2], \end{aligned} \tag{5}$$

where θ is the parameters of the base model, and x_t, i_t denotes encoded x_0, i_0 with mixed noise ϵ_x, ϵ_i respectively. Thus, we train θ' not only to generate intrinsics but also to provide useful information to the image by reducing $||\epsilon^{(x)}_{\theta,\theta'} - \epsilon_x||_2^2$. We independently sample ϵ_x and ϵ_i during the training and use the same ϵ during sampling.

3.2 Intrinsic Space Encoding

While incorporating various intrinsics enriches scene understanding, increasing their number introduces two main challenges: (1) higher training and sampling overhead, and (2) learning difficulty due to distributional mismatch between intrinsics and images. For instance, random noise sampled from a standard normal distribution tends to generate unnatural surface normals, where most pixels are green or blue (Du et al. 2023; Lin et al. 2024). We analyze the latent distribution for images and intrinsics in Sec. D.

To address this, we train an intrinsic VAE that encodes all intrinsics into a shared latent variable with the same dimensionality as the base model. This is achieved by extending the base VAE with additional CNN channels and fine-tuning it. During training, we randomly zero out individual intrinsics to prevent any particular intrinsic from being encoded dominantly, ensuring that the encoded latent representation reflects information from all intrinsics. As a result, our method enables efficient two-pass denoising for images and intrinsics, reducing memory from cross-domain attention. It also aligns intrinsic and image distributions, allowing minimal LoRA weight adjustments and eliminating the need for zero terminal SNR (Lin et al. 2024). Ablations of intrinsic VAE are presented in Tab. 5.

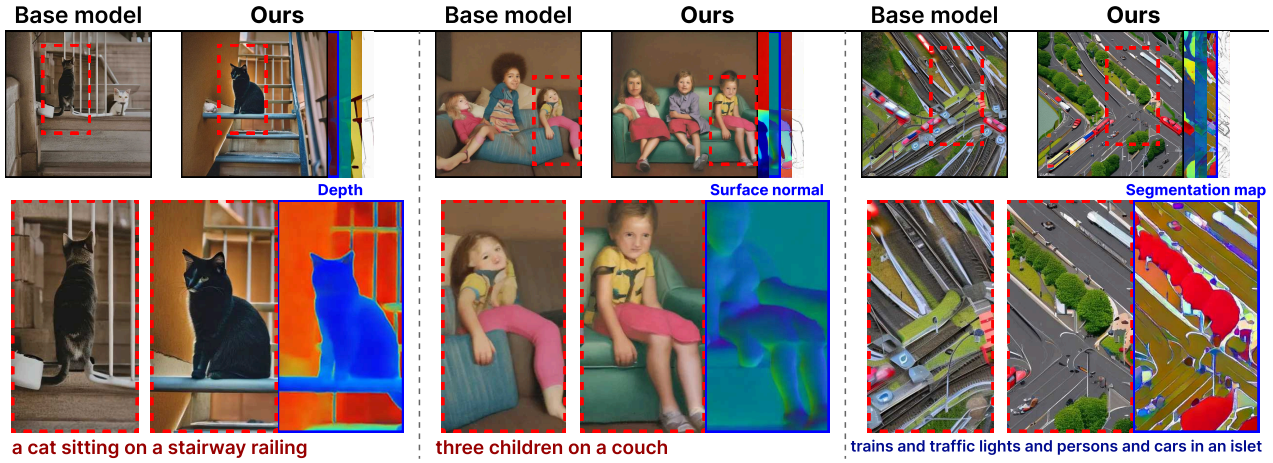


Figure 4: Comparison of base and I-LDM with generated intrinsics, providing key cues for reducing spatial inconsistencies.

4 Experiments

Here, we investigate the effectiveness of intrinsics with I-LDM for generating natural and spatially consistent images. Experimental details are provided in Sec. C.

Metrics. Our goal is to alleviate spatial inconsistencies by accurately reflecting the underlying scene while maintaining the quality and capability of the base model. However, precisely quantifying spatial inconsistency is inherently challenging, as ground-truth structural annotations do not exist for synthesized images. For this, we first utilize metrics based on human preference prediction, including ImageReward (IR) (Xu et al. 2024) and Human Preference Score (HPSv2) (Wu et al. 2023). To further assess the improvement in spatial inconsistency, we evaluate AI preference using GPT-4o, asking *which image better reflects the actual physical scene* between the base model and ours. Finally, to evaluate whether our method maintains the capability of the base diffusion model, we employ conventional CMMD (Jayasumana et al. 2024) and CLIP score (Hessel et al. 2021) to measure fidelity and text alignment, respectively. We measure the distribution discrepancy for CMMD in Sec. 4.1 by comparing with images from the base model, as real images are not available in this setting. This allows us to assess how well each method preserves the base model's quality, rather than indicating improvements over it.

Baselines. We employ Stable Diffusion 2.1 (SD2.1) as a pre-trained base model (Rombach et al. 2022). Comparison of its configuration with ours in terms of model size, inference time, and memory usage is in Tab. 6. For baselines, we first implement LDM3D (Stan et al. 2023) based on SD2.1, training a VAE to encode both images and depth into the existing latent space, and then fine-tune UNet. We also extend this method with four intrinsics that we use, denoted as LDM3D+. We also consider a naive method that adds extra CNN channels for intrinsics to the UNet and fine-tunes it (+CNN), and a variant that injects intrinsic features via spatial addition using the ControlNet architecture (+Spatial-Add). To compare with a two-stage generation approach, we first generate single or multiple intrinsics from text using our intrinsic LDM trained without images, and then use them to condition image generation via ControlNet, denoted as ControlNet-(Intrinsic). Finally, we evaluate base models with doubled sampling steps (denoted as $\times 2$).

Implementation details. For the training dataset, we use a subset of 542k images with aesthetic scores of 6.5 or higher from LAION-5B dataset (Schuhmann et al. 2022) in which the base model was trained. To estimate intrinsics, we use Metric3D (Hu et al. 2024) for depth maps and surface normals, SAM (Ravi et al. 2024) for segmentation maps, and a publicly available image-to-line drawing model. For training intrinsic VAE, we zero-mask each intrinsic with a probability of 0.1. To train the denoising process in the intrinsic domain, we adapt LoRA to self-attention weights with rank 32. The resolution of the images for training is set to 512×512, and 768×768 for sampling. All images were sampled with the CFG scale 7.5 and the sampling steps 25 (50 steps for $\times 2$). For the hyperparameters of our model in Eq. (2), we set $\tau = 900, L = \{3, 4, 5, 6, 7\}$ for w_d, $\alpha = 1, \tau = 800, \sigma = 100$ for w_g.

4.1 Learning Intrinsics on T2I Generation

We perform two tasks to evaluate the quality and spatial consistency of T2I generation with and without learning intrinsics. First, we assess general text conditions using images generated from Parti prompts (Parti) (Yu et al. 2022), designed to test capability across various categories and challenges. However, some prompts are too abstract or simple to evaluate spatial inconsistency (e.g., "energy" or "an orange"). Therefore, we design a second task, the multiple objects prompts (Multi), which constructs complex scenes where structural distortion can be more pronounced by placing many objects. We randomly sample 3 to 5 classes from the MS-COCO dataset (Lin et al. 2014) and one place from Places365 (Zhou et al. 2017), and construct 1,000 prompts in the format **"{class1}s and {class2}s and ... in a {place}"**.

Tabs. 1 and 2 and Fig. 3 summarize both quantitative metrics and qualitative results on the Parti and Multi prompts.

Table 1: Quantitative results on generated images for each prompt in Parti and Multi prompts. The evaluation is conducted using images sampled from three seeds for each prompt. The best result is bolded, and the second-best result is underlined.

	Parti prompt				Multiple objects			
Method	CMMD(↓)	CLIP(↑)	ImageReward(↑)	HPSv2(↑)	CMMD(↓)	CLIP(↑)	ImageReward(↑)	HPSv2(↑)
Base	-	0.2725	0.2998	0.2497	-	0.2671	-0.7389	0.2355
Base ×2	**0.00488**	<u>0.2726</u>	0.3647	0.2548	**0.00763**	<u>0.2676</u>	-0.7342	0.2395
Base +CNN	1.07265	0.2164	-1.5378	0.1713	2.51353	0.1986	-1.8801	0.1650
Base +SpatialAdd	0.39911	0.2413	-0.7374	0.2058	1.12689	0.2214	-1.5315	0.1807
LDM3D	0.57590	0.2693	0.1254	0.2349	1.20401	0.2472	-1.0290	0.2096
LDM3D+	0.11885	0.2644	0.0499	0.2357	0.33486	0.2575	-0.6855	0.2298
ControlNet-Norm	0.04578	0.2697	0.3444	0.2548	0.11182	0.2599	-0.6445	0.2388
ControlNet-Seg	0.08571	0.2672	0.3144	0.2550	0.36383	0.2521	-0.6504	0.2340
ControlNet-All	0.20289	0.2671	0.2900	0.2549	0.47350	0.2540	-0.6877	0.2342
I-LDM (Drop)	0.02074	0.2717	<u>0.3843</u>	<u>0.2560</u>	<u>0.03707</u>	0.2664	<u>-0.6148</u>	**0.2461**
I-LDM (Gaussian)	<u>0.01538</u>	**0.2739**	**0.4416**	**0.2582**	0.08228	**0.2687**	**-0.5739**	<u>0.2450</u>

Table 2: LLM evaluation of spatial consistency between images generated by the base model and ours. Ours, Tie, Base denote GPT-4o preference percentages.

	Parti prompt			Multiple objects		
Method	Ours(↑)	Tie	Base(↓)	Ours(↑)	Tie	Base(↓)
I-LDM (D)	41.9%	32.8%	25.2%	47.7%	36.6%	15.7%
I-LDM (G)	41.0%	34.9%	24.1%	44.3%	39.4%	16.3%

Table 3: Quantitative results on downscaling the intrinsic resolution provided by the intrinsic estimators for training.

Method	Resolution	CMMD(↓)	CLIP(↑)	ImageReward(↑)	HPSv2(↑)
I-LDM (D)	256	0.02027	0.2732	0.4572	0.2590
	384	0.02122	0.2739	0.5129	0.2606
	512	0.02730	0.2717	0.4719	0.2622
I-LDM (G)	256	0.01431	0.2742	0.4960	0.2606
	384	0.01466	0.2756	0.5311	0.2618
	512	0.01633	0.2741	0.4803	0.2611

Compared to the base model and other intrinsic-based approaches, learning to co-generate multiple intrinsics leads to better alignment with the physical structure of the scene without compromising image quality (see Sec. G.2 for detailed baseline analysis). Fig. 4 emphasizes that the base model may misrepresent the physical relationship between each object and incorrectly represent the boundaries of objects. In contrast, the intrinsics generated by our method clearly distinguish and reflect them in the image, resulting in a more accurate representation.

While our method reflects valuable information from the intrinsic, we emphasize that it also protects from the effects of ambiguities or inaccuracies from the intrinsic domain.

Preserving diversity. Although abstract scenes can be difficult to represent using geometric intrinsics, such as the "black hole" example in Fig. 3, we demonstrate that our approach does not constrain the diversity of the base model. This is quantitatively supported by maintaining CMMD and CLIP scores across various textual conditions provided by Parti prompts. Analysis on various categories and challenges is reported in Sec. G.1. This is achieved by disentangling the latent representations of images and intrinsics, enabling diversity of images beyond geometric constraints.



Figure 5: Comparison of the base model and I-LDM with Drop, Gaussian, and no weight scheduling.

Robustness to inaccurate intrinsics. An intriguing property of our framework is that we utilize existing intrinsic estimators without the additional cost of data collection. To investigate how our model performs given an imperfect intrinsic estimation, we downscale the resolution of the intrinsic for training. Tab. 3 shows that our model is robust to such errors or noises from intrinsic estimators. Moreover, Fig. 9 in Sec. F shows that meaningful information can still be exchanged despite misalignment across domains. This is due to the design of cross-domain self-attention process (Eq. (4)), which estimates cross-domain attention within the entire set of patches rather than just the same region.

Drop vs. Gaussian. Analysis of different weight scheduling are visualized in Fig. 5. With drop scheduling, there is a tendency to improve fine-grained details while maintaining the overall composition of the image. In contrast, Gaussian scheduling typically adjusts the overall image composition. On the other hand, completely removing the weight scheduling from the inference process degrades the quality of the images. However, fine-tuning approaches (+CNN, +SpatialAdd, LDM3D) suffer from severe image degradation and do not provide meaningful improvements in spatial inconsistency. See Sec. G.3 for further analysis.

4.2 Complex Hand Structure Generation

The human hand, with its intricate articulations that result in highly complex structures and self-occlusions, is a well-

Table 4: Quantitative results on 1000 generated images trained with InterHand samples. The best result is bolded.

Method	CMMD(↓)	HPSv2(↑)	Detection(↑)	Confidence(↑)
Base (Rombach et al. 2022)	0.7222	0.1475	29.82%	0.9310
Base ×2	**0.6906**	0.1472	30.22%	0.9271
I-LDM (Drop)	0.7750	0.1487	28.20%	0.9242
I-LDM (Gauss)	0.7051	**0.1506**	**39.40%**	**0.9384**

Table 5: Results on ablating each intrinsic, model settings.

	CMMD(↓)	CLIP(↑)	ImageReward(↑)	HPSv2(↑)
I-LDM	0.01633	0.2741	0.4803	0.2611
- Depth	0.01144	0.2720	0.3914	0.2578
- Normal	0.01299	0.2742	0.4655	0.2592
- Line	0.01919	0.2746	0.5074	0.2615
- Segment	0.01872	0.2727	0.4302	0.2590
$-w^{(l,t)}$	0.05460	0.2706	0.3852	0.2556
-VAE	0.01836	0.2702	0.3545	0.2551
LoRA-only	0.03314	0.2747	0.3617	0.2493
+resolution	0.08452	0.2640	0.1240	0.2470
$+w_{train}$	0.03827	0.2716	0.3920	0.2552

known example of a particularly challenging object to synthesize. We investigate how learning intrinsics affect the generation of such complex structures. First, since the base model tends to generate hand images with limited poses, we generate diverse and complex poses by training a dataset with various hand poses. Specifically, we extract 3,493 distinct samples from InterHand dataset (Moon et al. 2020). We then train the base model by applying LoRA without any annotations or additional information. The prompt is fixed as "A close-up of a hand with a black background". We use a hand detection model (Zhang et al. 2020) to measure the detection rate and confidence scores for detected cases, assessing whether generated hands follow an accurate structure.

We applied hand LoRA trained on the base model to the image domain of I-LDM to investigate how learned intrinsic information affects the generation of hands with dynamic poses. Tab. 4 and visualizations in Sec. H summarize the experimental results, which show that reflecting intrinsics substantially improves the generation of accurate hand structures. This experiment supports our claim that incorporating intrinsics helps generate complex structures.

4.3 Adapting I-LDM to Other Base Models

To verify the applicability of our method with more scalable or different architectures, we additionally implement I-LDM based on SDXL (Podell et al. 2023) with 2.5B parameters and based on PixArt-α (Chen et al. 2024b) with DiT architecture (Peebles and Xie 2023). Fig. 6 visualizes the qualitative results of our model. Samples with SDXL demonstrate that even with a more scalable base model, there are still spatial inconsistencies that I-LDM alleviates. The example of PixArt-α also shows that our method is generally applicable to Transformer-based architectures. For quantitative analysis and more qualitative results, see Secs. E.2 and G.3.

4.4 Ablation Study and Analysis

We perform ablation studies on each intrinsic and the training and sampling settings. Evaluations are performed on

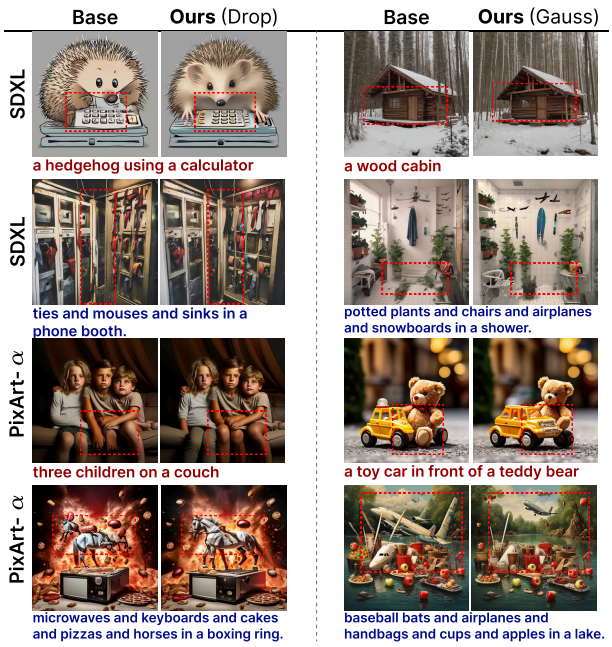


Figure 6: Examples of generated images of I-LDM using SDXL and PixArt-α as base models.

Parti prompts, using $w_g^{(l,t)}$ unless specifically noted. More detailed analysis including visualizations are provided in Sec. I and Sec. J.

Effects of each intrinsic scene properties. First, we perform an ablation study on each intrinsic scene property that composes I-LDM. Tab. 5 summarizes the effect of ablating each intrinsic, indicating that learning for each intrinsic shows meaningful improvement. While line drawing has a small negative effect on metrics, we qualitatively observe that it enhances the generation of object shapes.

Ablations on training settings. Next, we conduct an ablation study on the training and sampling process. As shown in Tab. 5, removing weight scheduling causes noticeable artifacts from the intrinsic domain, and excluding the intrinsic VAE results in inconsistent intrinsic generation. Training LoRA weights without image domain degrades intrinsic quality due to lack of image domain information. Additionally, using scheduling weights or increasing the resolution to 768×768 during training negatively impacts image quality.

5 Conclusion

To mitigate spatial inconsistencies in image generation, we proposed learning intrinsic scene properties aligned with the images. We presented I-LDM, a model that jointly generates images and intrinsics (i.e. depth map, surface normal, line drawing and segmentation map). We carefully designed our architecture to share valuable self-attention information between the image and the intrinsics while preserving the quality and capabilities of the base model. Experimental results on Parti, Multi prompts, and InterHand demonstrated the effectiveness of our approach.

References

Amit, T.; Shaharbany, T.; Nachmani, E.; and Wolf, L. 2021. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*.

Barrow, H.; Tenenbaum, J.; Hanson, A.; and Riseman, E. 1978. Recovering intrinsic scene characteristics. *Comput. vis. syst*, 2(3-26): 2.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chen, J.; Ge, C.; Xie, E.; Wu, Y.; Yao, L.; Ren, X.; Wang, Z.; Luo, P.; Lu, H.; and Li, Z. 2024a. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, 74–91. Springer.

Chen, J.; YU, J.; GE, C.; Yao, L.; Xie, E.; Wang, Z.; Kwok, J.; Luo, P.; Lu, H.; and Li, Z. 2024b. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. In *ICLR*.

Deitke, M.; Liu, R.; Wallingford, M.; Ngo, H.; Michel, O.; Kusupati, A.; Fan, A.; Laforte, C.; Voleti, V.; Gadre, S. Y.; et al. 2024. Objaverse-xl: A universe of 10m+ 3d objects. *NeurIPS*, 36.

Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsani, K.; Kembhavi, A.; and Farhadi, A. 2023. Objaverse: A universe of annotated 3d objects. In *CVPR*, 13142–13153.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *NeurIPS*, 34: 8780–8794.

Du, X.; Kolkin, N.; Shakhnarovich, G.; and Bhattad, A. 2023. Generative models: What do they know? do they know things? let's find out! *arXiv preprint arXiv:2311.17137*.

El Banani, M.; Raj, A.; Maninis, K.-K.; Kar, A.; Li, Y.; Rubinstein, M.; Sun, D.; Guibas, L.; Johnson, J.; and Jampani, V. 2024. Probing the 3d awareness of visual foundation models. In *CVPR*, 21795–21806.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis.

Fu, X.; Yin, W.; Hu, M.; Wang, K.; Ma, Y.; Tan, P.; Shen, S.; Lin, D.; and Long, X. 2025. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 241–258. Springer.

Hessel, J.; Holtzman, A.; Forbes, M.; Bras, R. L.; and Choi, Y. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *NeurIPS*, 33: 6840–6851.

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

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hu, M.; Yin, W.; Zhang, C.; Cai, Z.; Long, X.; Chen, H.; Wang, K.; Yu, G.; Shen, C.; and Shen, S. 2024. Metric3D v2: A Versatile Monocular Geometric Foundation Model for Zero-shot Metric Depth and Surface Normal Estimation. *arXiv preprint arXiv:2404.15506*.

Hwang, I.; Lee, S.; Kwak, Y.; Oh, S. J.; Teney, D.; Kim, J.-H.; and Zhang, B.-T. 2022. Selecmix: Debiased learning by contradicting-pair sampling. *Advances in Neural Information Processing Systems*, 35: 14345–14357.

Jayasumana, S.; Ramalingam, S.; Veit, A.; Glasner, D.; Chakrabarti, A.; and Kumar, S. 2024. Rethinking fid: Towards a better evaluation metric for image generation. In *CVPR*, 9307–9315.

Ji, Y.; Chen, Z.; Xie, E.; Hong, L.; Liu, X.; Liu, Z.; Lu, T.; Li, Z.; and Luo, P. 2023. Ddp: Diffusion model for dense visual prediction. In *ICCV*, 21741–21752.

Jun, H.; and Nichol, A. 2023. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*.

Ke, B.; Obukhov, A.; Huang, S.; Metzger, N.; Daudt, R. C.; and Schindler, K. 2024. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 9492–9502.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. *arXiv:2304.02643*.

Lee, H.; Hwang, I.; Go, H.; Choi, W.-S.; Kim, K.; and Zhang, B.-T. 2023. Learning geometry-aware representations by sketching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23315–23326.

Lee, S.; Hwang, I.; Kang, G.-C.; and Zhang, B.-T. 2022. Improving Robustness to Texture Bias via Shape-focused Augmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 4322–4330. IEEE Computer Society.

Li, Z.; Zhou, Q.; Zhang, X.; Zhang, Y.; Wang, Y.; and Xie, W. 2023. Open-vocabulary object segmentation with diffusion models. In *ICCV*, 7667–7676.

Lin, S.; Liu, B.; Li, J.; and Yang, X. 2024. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 5404–5411.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Luo, S.; and Hu, W. 2021. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2837–2845.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

Moon, G.; Yu, S.-I.; Wen, H.; Shiratori, T.; and Lee, K. M. 2020. InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image. In *ECCV*.

Nguyen, Q.; Vu, T.; Tran, A.; and Nguyen, K. 2024. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. *NeurIPS*, 36.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4195–4205.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Qiu, L.; Chen, G.; Gu, X.; Zuo, Q.; Xu, M.; Wu, Y.; Yuan, W.; Dong, Z.; Bo, L.; and Han, X. 2024. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *CVPR*, 9914–9925.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. 8748–8763. PMLR.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; Mintun, E.; Pan, J.; Alwala, K. V.; Carion, N.; Wu, C.-Y.; Girshick, R.; Dollár, P.; and Feichtenhofer, C. 2024. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714*.

Razavi, A.; Van den Oord, A.; and Vinyals, O. 2019. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 32.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, 10684–10695.

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. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35: 36479–36494.

Salimans, T.; and Ho, J. 2022. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*.

Sauer, A.; Boesel, F.; Dockhorn, T.; Blattmann, A.; Esser, P.; and Rombach, R. 2024. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*.

Saxena, S.; Herrmann, C.; Hur, J.; Kar, A.; Norouzi, M.; Sun, D.; and Fleet, D. J. 2023. The Surprising Effectiveness of Diffusion Models for Optical Flow and Monocular Depth Estimation. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *NeurIPS*, volume 36, 39443–39469. Curran Associates, Inc.

Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35: 25278–25294.

Shen, D.; Song, G.; Xue, Z.; Wang, F.-Y.; and Liu, Y. 2024. Rethinking the Spatial Inconsistency in Classifier-Free Diffusion Guidance. In *CVPR*, 9370–9379.

Shi, Z.; Zhou, X.; Qiu, X.; and Zhu, X. 2020. Improving image captioning with better use of captions. *arXiv preprint arXiv:2006.11807*.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1): 1–48.

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

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32.

Stan, G. B. M.; Wofk, D.; Fox, S.; Redden, A.; Saxton, W.; Yu, J.; Aflalo, E.; Tseng, S.-Y.; Nonato, F.; Muller, M.; et al. 2023. Ldm3d: Latent diffusion model for 3d. *arXiv preprint arXiv:2305.10853*.

Voynov, A.; Aberman, K.; and Cohen-Or, D. 2023. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 Conference Proceedings*, 1–11.

Wang, X.; and Gupta, A. 2016. Generative image modeling using style and structure adversarial networks. In *European conference on computer vision*, 318–335. Springer.

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

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920.

Xu, J.; Liu, X.; Wu, Y.; Tong, Y.; Li, Q.; Ding, M.; Tang, J.; and Dong, Y. 2024. Imagereward: Learning and evaluating human preferences for text-to-image generation. *NeurIPS*, 36.

Yu, J.; Xu, Y.; Koh, J. Y.; Luong, T.; Baid, G.; Wang, Z.; Vasudevan, V.; Ku, A.; Yang, Y.; Ayan, B. K.; et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3): 5.

Zhan, G.; Zheng, C.; Xie, W.; and Zisserman, A. 2023. A General Protocol to Probe Large Vision Models for 3D Physical Understanding. In *NeurIPS*.

Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.-L.; and Grundmann, M. 2020. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *CVPR*, 3836–3847.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6): 1452–1464.

Zhou, L.; Du, Y.; and Wu, J. 2021. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 5826–5835.

Towards Spatially Consistent Image Generation: On Incorporating Intrinsic Scene Properties into Diffusion Models

Supplementary Material

A Theoretical Motivation of Utilizing Intrinsics to Reflect the Underlying Scene

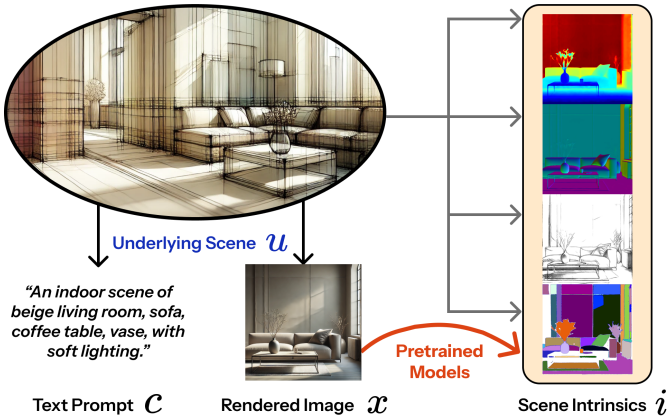


Figure 7: We assume that an image x is constructed from a latent underlying scene u, while text c and intrinsic scene properties i are different instantiations entailed by u, i.e., $u \rightarrow (x, i, c)$. Instead of learning u directly, which is difficult to obtain, we leverage pre-trained models to estimate various intrinsics that reflect u.

A.1 Log-likelihood with Underlying Scene

Given a dataset $(x, c) \sim D$ of image x and text c pairs, a typical generative model, such as DDPMs (Ho, Jain, and Abbeel 2020), aims to maximize the empirical log-likelihood:

$$\arg\max_{\theta} \mathbb{E}_{(x,c)\sim D}[\log p_\theta(x|c)]. \tag{6}$$

However, considering the latent underlying scene u from which the data is built, we can consider a relationship as in Fig. 7, where x and c are generated from the underlying scene u. For low $p_\theta(x|u)$, the generated x will not accurately reflect u, resulting in a spatially inconsistent image. Thus, the objective of generating x from c with reflecting u can be rewritten as:

$$\arg\max_{\theta} \mathbb{E}_{(x,c)\sim D}\mathbb{E}_{p_\theta(u|c)}\left[\log p_\theta(x|u)\right], \tag{7}$$

which is equivalent to the following (see Sec. B.2):

$$\arg\max_{\theta} \mathbb{E}_{(x,c)\sim D}[\log p_\theta(x|c) - KL(p_\theta(u|c)||p_\theta(u|x,c))]. \tag{8}$$

In other words, considering the underlying scene u, we aim to maximize the conventional log-likelihood while minimizing the Kullback-Leibler (KL) divergence between $p_\theta(u|c)$ and $p_\theta(u|x,c)$. However, information on u in c is typically very limited and the KL term is intractable since u is not directly observable. For this, we consider intrinsic scene properties that provide rich information about u.

A.2 Introducing Intrinsic Scene Properties

In this work, we consider four intrinsic scene properties; depth map, surface normal, segmentation map, and line drawing, to reflect u from various perspectives. Notably, such intrinsics i could be easily obtained using existing pre-trained estimators (Hu et al. 2024; Kirillov et al. 2023), without requiring an additional layer of training or data collection. With an augmented data set $(x, i, c) \sim D^+$, the objective can be written as follows:

$$\arg\max_{\theta} \mathbb{E}_{(x,i,c)\sim D^+}[\log p_\theta(x|i,c) - KL(p_\theta(u|i,c)||p_\theta(u|x,i,c))]. \tag{9}$$

Our key insight is that the more information i contains about u, the better the trained model can minimize KL divergence (see Sec. B.3 for formal derivation). Maximization of $p_\theta(x|i,c)$ implies optimizing the process of providing intrinsics as control elements to the image, such as ControlNet (Zhang, Rao, and Agrawala 2023). In practice, since our goal is not to provide i to the text-to-image generation process, we instead maximize $\log p_\theta(x,i|c) = \log p_\theta(x|i,c) + \log p_\theta(i|c)$.

B Formal Derivation for Eqs. (8) and (9)

We first start by constructing a graphical model of u, x, i, and c. Next, we show that Eq. (7) is equivalent to Eq. (8). Finally, we show that the more information i contains about u, the lower the objective Eq. (9).

B.1 Graphical model with the underlying scene

As illustrated in Fig. 7, we consider an image x, text c, and intrinsic scene properties i as different instantiations entailed by a latent underlying scene u. In other words, we can construct a probabilistic graphical model (PGM) relation $u \to x, i, c$. Using Bayes' Rule, the following equation holds:

$$p_\theta(u|x,c) = \frac{p_\theta(u,x,c)}{p_\theta(x,c)} = \frac{p_\theta(x|u)p_\theta(c|u)p_\theta(u)}{p_\theta(x,c)} = \frac{p_\theta(x|u)p_\theta(u|c)p_\theta(c)}{p_\theta(x,c)} = \frac{p_\theta(x|u)p_\theta(u|c)}{p_\theta(x|c)}. \tag{10}$$

B.2 Equivalence between Eq. (7) and Eq. (8)

Starting from the KL divergence between $p_\theta(u|c)$ and $p_\theta(u|x,c)$ in Eq. (8),

$$KL(p_\theta(u|c)||p_\theta(u|x,c)) = \int p_\theta(u|c) \log \frac{p_\theta(u|c)}{p_\theta(u|x,c)} du = \int p_\theta(u|c) \left[\log p_\theta(u|c) - \log p_\theta(u|x,c)\right] du. \tag{11}$$

Substituting Eq. (10) into Eq. (11),

$$\begin{aligned} KL(p_\theta(u|c)||p_\theta(u|x,c)) &= \int p_\theta(u|c) \left[\log p_\theta(u|c) - (\log p_\theta(x|u) + \log p_\theta(u|c) - \log p_\theta(x|c))\right] du \\ &= \int p_\theta(u|c) \left[-\log p_\theta(x|u) + \log p_\theta(x|c)\right] du \\ &= -\mathbb{E}_{p_\theta(u|c)} \left[\log p_\theta(x|u)\right] + \log p_\theta(x|c). \end{aligned} \tag{12}$$

Thus,

$$\mathbb{E}_{p_\theta(u|c)} \left[\log p_\theta(x|u)\right] = \log p_\theta(x|c) - KL(p_\theta(u|c)||p_\theta(u|x,c)). \tag{13}$$

which indicates the equivalence between Eq. (7) and Eq. (8). By replacing c with (i,c), we can get $\mathbb{E}_{p_\theta(u|i,c)} \left[\log p_\theta(x|u)\right] = \log p_\theta(x|i,c) - KL(p_\theta(u|i,c)||p_\theta(u|x,i,c))$ in the same way.

B.3 Inequality between Eq. (8) and Eq. (9)

To support our claim in Line 210, we first show that the inequality

$$KL(p_\theta(u|c)||p_\theta(u|x,c)) \geq \mathbb{E}_{p_\theta(i|c)} KL(p_\theta(u|i,c)||p_\theta(u|x,i,c) \tag{14}$$

holds. This can be proved by the chain rule for KL divergence and the conditional independence $i \perp (x,c)|u$.

Applying the KL chain rule

$$KL(p(x,y)||q(x,y)) = KL(p(y)||q(y)) + \mathbb{E}_{p(y)}[KL(p(x|y)||q(x|y))] \tag{15}$$

in $KL(p_\theta(u,i|c)||p_\theta(u,i|x,c)$ with decomposing by conditioning on i and on u respectively,

$$KL(p_\theta(u,i|c)||p_\theta(u,i|x,c)) = \mathbb{E}_{p_\theta(i|c)}[KL(p_\theta(u|i,c)||p_\theta(u|x,i,c))] + KL(p_\theta(i|c)||p_\theta(i|x,c)) \tag{16}$$

$$KL(p_\theta(u,i|c)||p_\theta(u,i|x,c)) = \mathbb{E}_{p_\theta(u|c)}[KL(p_\theta(i|u,c)||p_\theta(i|x,u,c))] + KL(p_\theta(u|c)||p_\theta(u|x,c)). \tag{17}$$

Then, applying the conditional independence $i \perp (x,c)|u$ implied by the relation $u \to x, i, c$,

$$p_\theta(i|u,x,c) = p_\theta(i|u) \tag{18}$$

$$p_\theta(i|u,c) = p_\theta(i|u), \tag{19}$$

$KL(p_\theta(i|u,c)||p_\theta(i|x,u,c)) = KL(p_\theta(i|u)||p_\theta(i|u)) = 0$ in Eq. (17). Since $KL(p_\theta(i|c)||p_\theta(i|x,c)) \geq 0$ in Eq. (16),

$$\begin{aligned} KL(p_\theta(u,i|c)||p_\theta(u,i|x,c)) &= \mathbb{E}_{p_\theta(u|c)}[\cancel{KL(p_\theta(i|u,c)||p_\theta(i|x,u,c))}] + KL(p_\theta(u|c)||p_\theta(u|x,c)) \\ &= 0 + \cancel{KL(p_\theta(u|c)||p_\theta(u|x,c))} \\ &= \mathbb{E}_{p_\theta(i|c)}[KL(p_\theta(u|i,c)||p_\theta(u|x,i,c))] + KL(p_\theta(i|c)||p_\theta(i|x,c)) \\ &\geq \mathbb{E}_{p_\theta(i|c)}[KL(p_\theta(u|i,c)||p_\theta(u|x,i,c))]. \end{aligned} \tag{20}$$

Hence,

$$KL(p_\theta(u|c)||p_\theta(u|x,c)) \geq \mathbb{E}_{p_\theta(i|c)}[KL(p_\theta(u|i,c)||p_\theta(u|x,i,c))]. \tag{21}$$

Applying for $u \to i_1, i_2, ...i_k$ in the same way, we can get:

$$\begin{aligned} KL(p_\theta(u|c)||p_\theta(u|x,c)) &\geq \mathbb{E}_{p_\theta(i_1|c)}[KL(p_\theta(u|i_1,c)||p_\theta(u|x,i_1,c))] \\ &\geq \mathbb{E}_{p_\theta(i_1,...,i_k|c)}[KL(p_\theta(u|i_1,...,i_k,c)||p_\theta(u|x,i_1,...,i_k,c))], \end{aligned} \tag{22}$$

which supports our claim that with more information $i = i_1, .., i_k$, the lower the objective Eq. (9).

C Experimental Details

C.1 Dataset

For the image dataset, we use a subset of 542k images with aesthetic scores of 6.5 or higher from LAION-5B (Schuhmann et al. 2022). We use Metric3D (Hu et al. 2024) with a ViT-L backbone to estimate the depth map and surface normal. Following the method of Marigold (Ke et al. 2024), we normalized each depth map by mapping its 2% and 98% percentiles to -1 and 1, respectively. We use an automatic mask generator with *sam2.1_hiera_base_plus* model (Kirillov et al. 2023) to estimate segmentation maps, with *pred_iou_thresh* 0.7, *stability_score_thresh* 0.92, *stability_score_offset* 0.7, *crop_n_layers* 1, and *crop_n_points_downscale_factor* 2. To estimate line drawings, we use a publicly available image-to-line drawing model[1]. All intrinsics are normalized to a range of $[-1, 1]$.

C.2 Training Intrinsic VAE

For the input and output of the Intrinsic VAE, all intrinsics are encoded in RGB format (total 4×3 channels). We train intrinsic VAE with a batch size of 256 for 2000 steps with a learning rate of $1e-4$ using the AdamW optimizer for each base model, with resolution 768×768 for SD2.1 and 1024×1024 for SDXL. After 200 training steps, we apply LPIPS loss on a scale of 1.0 and discriminator loss on a scale of 0.5. The discriminator uses a 5-layer CNN architecture. To improve training stability, random noise is added to the fake images during the initial training stages, and the noise proportion is gradually reduced to zero over the first 160 steps. A colormap is applied to the grayscale depth maps to ensure that different depth levels are perceptually distinguishable.

C.3 Training I-LDM

To train the denoising process in the intrinsic domain, we adapt LoRA for the self-attention layers with rank 32, training with a batch size of 128 for 5000 steps with a learning rate of $2e$-4 using the AdamW optimizer. We set $\lambda = 4$ in Eq. (5). Since the surface normal estimator occasionally provides a flat surface for images such as paintings, we discarded the data with a surface normal standard deviation of 0.03 or less when training the UNet. The resolution of the images for training is fixed at 512 using the same random cropping method suggested in Podell et al. (2023), while we sample the image with a resolution of 768×768. For the loss function Eq. (5) with SD2.1, we use the corresponding loss function with v-prediction (Salimans and Ho 2022) instead of ϵ-prediction. We selected hyperparameters based on prior works and rule of thumb, rather than extensive tuning, due to the high cost of each experimental run.

For weight scheduling, we assign layer indices l as follows: SD2.1 consists of four down-blocks, four corresponding up-blocks, and one mid-block. We assign indices 1 through 4 from the early block to the four down-blocks, 5 to the mid-block, and 6 through 9 to the up-blocks. We denote timestep t as the range of DDPM [0, 1000] (Ho, Jain, and Abbeel 2020). For the hyperparameters in weight scheduling mechanism (Eq. (2)), we tested τ and σ in increments of 100, and tested α with values of 0.5 and 1.0.

C.4 Metrics

The CMMD values (Jayasumana et al. 2024) reported in Tabs. 1, 3 and 5 are computed by comparing generated images with images from the base model. The CMMD values in Tab. 4 are computed by comparing the generated images with the 3,493 hand images from the InterHand dataset used during training. For HPSv2 (Wu et al. 2023), we use the HPS v2.1 model for evaluation. To evaluate the AI preference for spatial inconsistency using GPT-4o, we use the following procedure. First, we provide the model with two images generated by the base model and I-LDM with the same prompt and seed, horizontally aligned with a separator line and vertically resized to 512 resolution. We then provide the instruction in the following format, similar to the prompts proposed by PixArt-Σ (Chen et al. 2024a): *"As an AI visual assistant, you are analyzing two images arranged to the left and right of a given image. When presented with a specific caption, it is required to evaluate and determine whether the two images are equally good or which image better reflects the actual physical scene. Please pay attention to the key information, probable architectures, consistency for occlusion, avoidance of paradoxical visual effects, correct object shapes and structures, and anatomical accuracies. The caption for the two images is: {prompt}". "Please respond me strictly in the following format: <The left image is better> or <The right image is better> or <The two images are tied>. The reason is (give your reason here)."*

C.5 Network Configuration.

Tab. 6 summarizes the parameter sizes, time, and memory requirements for generating single images. Note that for Gaussian weight scheduling, we disable the denoising process for steps in $t \leq 500$, since it only uses information from the intrinsic domain near $\tau = 800$. Since I-LDM generates images and intrinsics simultaneously, the time consumption is higher than for the base model. However, the comparison between Base $\times 2$ and I-LDM (Gauss) in Tabs. 1 and 4 shows that our model produces more consistent images with similar time consumption. It also reveals that the number of additional parameters and memory size is about 10% of the base model.

[1] https://huggingface.co/spaces/awacke1/Image-to-Line-Drawings/blob/main/model.pth

Table 6: Model configuration (A6000 GPU, fp32, single image).

	# params	inference time	memory
Base	865M (UNet) 83.6M (VAE)	8.76s 17.2s (×2)	6.65 GiB
Ours	872M (UNet) 167M (VAE)	28.9s (Drop) 19.0s (Gauss)	7.37 GiB

Algorithm 1: Inference Pipeline of I-LDM

Require: scheduler timesteps $\mathcal{T}$, CFG scale s, prompt and negative prompt embeddings p, n.

Sample noise $\epsilon \sim \mathcal{N}(0, I)$.
Initialize latent batch $z = [\epsilon, \epsilon, \epsilon, \epsilon]$ and condition batch $c = [n, p, n, p]$.
for each $t \in \mathcal{T}$ **do**
  Set input to model: (z, t, c).
  for each transformer layer $l = 1, \ldots, L$ **do**
    Attention processor:
      Receives $z^{(l)} = \left[z_x^{(l,n)},\ z_x^{(l,p)},\ z_i^{(l,n)},\ z_i^{(l,p)}\right]$, weights $Q_x^{(l)}, K_x^{(l)}, V_x^{(l)}$, LoRA weights $Q'^{(l)}, K'^{(l)}, V'^{(l)}$.
      Compute $Q_x^{(l)} \cdot z_x^{(l,\cdot)}, K_x^{(l)} \cdot z_x^{(l,\cdot)}, V_x^{(l)} \cdot z_x^{(l,\cdot)}$ with *disabling* LoRA.
      Compute $\left(Q_x^{(l)} + Q'^{(l)}\right) \cdot z_i^{(l,\cdot)}, \left(K_x^{(l)} + K'^{(l)}\right) \cdot z_i^{(l,\cdot)}, \left(V_x^{(l)} + V'^{(l)}\right) \cdot z_i^{(l,\cdot)}$ with *enabling* LoRA.
      Compute $q., k., v.$ (Eq. (1)) and $W^{(l,t)}$ (Sec. 3.1)
    Compute attention: $Attn_x^{(l,t)}, Attn_i$ (Eq. (4))
  end for
  Obtain model outputs $\left[\epsilon_x^{(n)},\ \epsilon_x^{(p)},\ \epsilon_i^{(n)},\ \epsilon_i^{(p)}\right]$.
  Classifier-free guidance (per-domain):
    $\hat{\epsilon}_x \leftarrow \epsilon_x^{(n)} + s\left(\epsilon_x^{(p)} - \epsilon_x^{(n)}\right)$
    $\hat{\epsilon}_i \leftarrow \epsilon_i^{(n)} + s\left(\epsilon_i^{(p)} - \epsilon_i^{(n)}\right)$
  $z \leftarrow \text{SchedulerUpdate}\left(z, [\hat{\epsilon}_x,\ \hat{\epsilon}_x,\ \hat{\epsilon}_i,\ \hat{\epsilon}_i], t\right)$
end for
return z_x.

C.6 Implementation details

To efficiently co-generate images and intrinsics, we leverage our architecture design in which the intrinsic LDM corresponds to the image LDM augmented only with LoRA parameters. By selectively enabling or disabling these LoRA components, we switch between the image and intrinsic domains within a single shared model. For implementation, we first concatenate the image and intrinsic latents batch-wise. Inside each attention processor of the LDM, we split the batch into two halves to compute queries, keys, and values of each domain. For the first half of the batch, LoRA weights are not applied ($Q_x \cdot z_x, K_x \cdot z_x, V_x \cdot z_x$), while for the second half, LoRA weights are applied ($Q_i \cdot z_i, K_i \cdot z_i, V_i \cdot z_i$). Keys and values of both domains are then concatenated feature-wise to form cross-domain co-generation as in Eq. (1). Finally, the queries, keys, and values from both domains are concatenated batch-wise and passed to the attention operation.

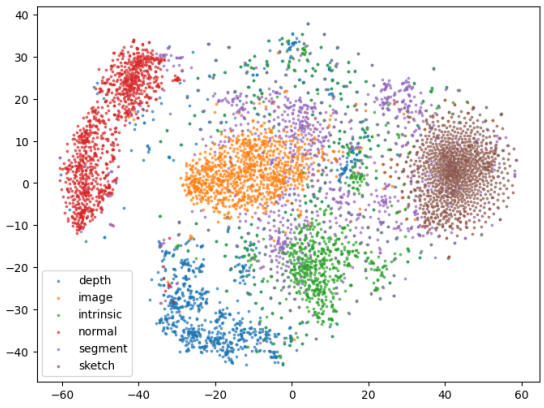
During inference, classifier-free guidance (CFG) is applied to both the image and intrinsic domains, requiring four batch elements per sample. The weight scheduling mechanism introduced in Sec. 3.1 is used, passing the current layer index l and denoising timestep t to each attention processor. The processor computes the weight matrix $W^{(l,t)}$ from 4, which is added as a weight bias in the attention computation. The inference pipeline is summarized in Sec. C.6.

C.7 Computing Infrastructure

Experiments are conducted on a workstation equipped with 2× NVIDIA RTX A6000 GPUs, AMD Ryzen Threadripper 3960X 24-Core CPU, and 256GB of RAM, running Ubuntu 22.04.4 LTS. We use PyTorch 2.4.1 with CUDA 12.3 as the primary deep learning framework, and employ diffusers v0.28.0 and transformers v4.44.2 for implementation.

D Latent Distribution for Images and Intrinsics

As described in Sec. 3.2, if the latent distribution of the intrinsic features encoded by the VAE differs significantly from that of the images, challenges may arise during the training and sampling processes. Fig. 8a visualizes these distributions



(a) Visualizations of latent distributions with t-SNE.

Intrinsic	MMD
Intrinsic VAE	**7.921**
Depth map	8.225
Surface normal	24.349
Line drawings	44.401
Segmentation map	8.893

(b) MMD between image and intrinsic.

Figure 8: Quantitative and qualitative analysis of the latent distribution of each intrinsic and intrinsic VAE.

using t-SNE: it shows the distribution of 1,000 images from the LAION dataset encoded by the pre-trained VAE (orange); the distributions of each intrinsic encoded by the pre-trained VAE, depth maps (blue), surface normals (red), line drawings (purple), and segmentation maps (brown); and the distribution of all intrinsics encoded by our intrinsic VAE (green). Fig. 8b presents a table listing the Maximum Mean Discrepancy (MMD) values, calculated using a Gaussian kernel, between the latent distributions of the images and each intrinsic. The results show that certain intrinsics, such as surface normals and line drawings, are far from the image distribution. In contrast, the distribution of the intrinsics encoded by the intrinsic VAE closely resembles that of the images.

E Adapting I-LDM into SDXL and PixArt-α

E.1 Training details

For SDXL, we adapt LoRA weights to self and cross-attention layers with rank 8, with batch size 64 for 10000 steps with a learning rate of $2e-5$, while other settings are equal to SD2.1. The resolution of the images for training is fixed at 512, while we sample the image with a resolution of 1024×1024. For sampling, the cfg scale is set to 7.5. For weight scheduling, we assign layer indices l as follows: SDXL consists of three down-blocks, three corresponding up-blocks, and one mid-block; we assign indices 1 through 3 from the early block to the three down-blocks, 4 to the mid-block, and 5 through 7 to the up-blocks. For generating samples from Multi prompts, we set $L = \{4\}, \tau = 900$ for w_d, $\alpha = 0.5, \tau = 800, \sigma = 50$ for w_g. For complex hand structure generation task, we set $L = \{3, 4, 5\}, \tau = 900$ for w_d, $\alpha = 1, \tau = 800, \sigma = 100$ for w_g.

For PixArt-α, we use PixArt-α-512 model as the base model. We adapt LoRA weights to self-attention layers with rank 32, with batch size 128 for 2000 steps with a learning rate of $2e-6$, while other settings are equal to SD2.1. The resolution of the images for training is fixed at 384, while we sample the image with a resolution of 512×512. For sampling, the cfg scale is set to 4.5. PixArt-α consists of 28 Transformer blocks, indicated by 1 through 28 from the first block. For generating samples from Multi prompts, we set $L = [1, 28], \tau = 900$ for w_d, $\alpha = 1, \tau = 800, \sigma = 0.3$ for w_g.

E.2 Quantitative Results

Table 7: Quantitative results on generated images in Multi prompts based on SDXL and PixArt-α. The evaluation is done with images sampled from 3 seeds for each prompt. The best result is bolded.

	SDXL (Podell et al. 2023)						PixArt-α (Chen et al. 2024b)					
			human preference estimator		LLM (GPT-4o)				human preference estimator		LLM (GPT-4o)	
Method	CMMD(↓)	CLIP(↑)	ImageReward(↑)	HPSv2(↑)	win(↑)	lose(↓)	CMMD(↓)	CLIP(↑)	ImageReward(↑)	HPSv2(↑)	win(↑)	lose(↓)
Base	-	0.2876	-0.1136	0.2624	-	-	-	0.2707	-0.0665	0.2891	-	-
I-LDM (Drop)	0.01729	0.2892	-0.0484	0.2624	31.8%	15.7%	0.0044	**0.2718**	**-0.0428**	**0.2899**	**21.3%**	**12.2%**
I-LDM (Gauss)	0.02694	**0.2909**	**-0.0469**	**0.2634**	**40.6%**	**14.1%**	0.0032	0.2713	-0.0522	0.2897	19.0%	13.0%

Tabs. 7 and 8 summarizes the quantitative results for Multi prompts and InterHand experiment based on SDXL and PixArt-α. The qualitative results, including the generated intrinsic, are visualized in Sec. G.3. This shows that our method is also applicable to more scalable base models and other architectures. However, the decrease in improvement over SD2.1 for the human preference estimator may be due to the fact that as the scalability of the base model increases, the score becomes

Table 8: Quantitative results on 1000 generated images based on SDXL and PixArt-α trained with InterHand samples. The best result is bolded.

	SDXL (Podell et al. 2023)				PixArt-α (Chen et al. 2024b)			
Method	CMMD(↓)	HPSv2(↑)	Detection(↑)	Confidence(↑)	CMMD(↓)	HPSv2(↑)	Detection(↑)	Confidence(↑)
Base	0.9140	**0.1640**	57.03%	0.9181	**1.2529**	0.1734	77.00%	0.8704
I-LDM (Drop)	**0.7690**	0.1636	72.40%	0.9301	1.2774	**0.1785**	89.00%	**0.8721**
I-LDM (Gauss)	0.8554	0.1621	**74.10%**	**0.9372**	1.2707	0.1732	**92.20%**	0.8688

more dominated by other evaluation factors (e.g., human aesthetics, text alignment) as the spatial inconsistency becomes less pronounced. On the other hand, the win-loss ratio of GPT-4o and the high detection rate and confidence of the InterHand experiment, which excludes other evaluation factors besides spatial inconsistency, show that our model produces more spatially consistent images.

F Additional Analysis for the Robustness to Intrinsic Errors

In this section, we perform further quantitative and qualitative analysis of the relation between intrinsic and image quality.

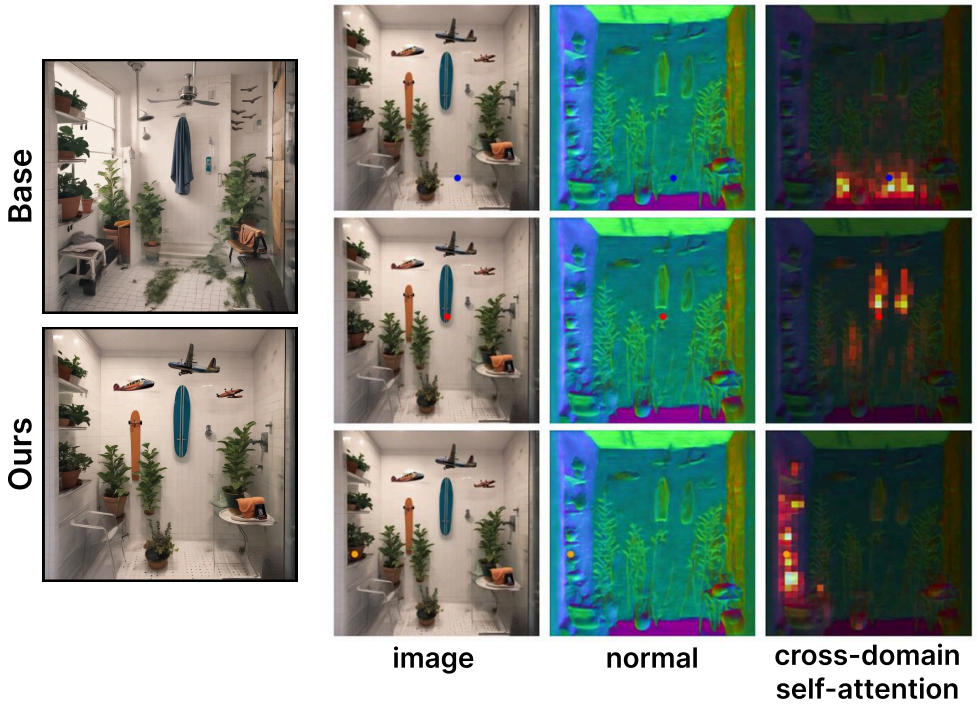


Figure 9: Visualizations of the generated image (Left) and surface normal (Middle) from I-LDM (Drop) based on SDXL with prompt ”potted plants and chairs and airplanes and snowboards in a shower.”. (Right) The heatmap of cross-domain self-attention in colored dot areas where the image domain references from the intrinsic domain. Cross-domain self-attention to corners of walls and floors (Top), snowboards (Middle), and pots (Bottom).

To evaluate the accuracy of the generated intrinsics, we estimate depth maps and surface normals from the generated images using Metric3D and compare the errors between the generated intrinsics. Tab. 9 shows the intrinsic accuracy and human preference estimatior results of I-LDM with Drop, Gaussian, and no weight scheduling. Here, depth RMSE denotes the root mean square error of the depth maps, and Mean $^\circ$ denotes the average angular error in degrees between the surface normals. This is comparable in accuracy to the existing work (Du et al. 2023), which trained LoRA to generate intrinsic from Stable Diffusion (while the difference in scale in depth RMSE is due to differences in depth encoding). Removing weight scheduling can slightly reduce the misalignment between image and intrinsic, but this results in a degradation of image quality and the generation of unnatural object layouts, as visualized in Fig. 5.

We further verify that the cross-domain self-attention mechanism can convey meaningful information even in the presence of cross-domain misalignment. Fig. 9 illustrates the regions in the intrinsic domain that the image domain references through

Table 9: RMS error for generated depth map and mean ° error for surface normal for Multi prompts.

Base	Schedule	Depth RMSE (↓)	Mean ° (↓)	ImageReward(↑)	HPSv2(↑)
SD2.1	Drop	0.2974	14.66	-0.6148	0.2461
	Gaussian	0.3059	15.48	-0.5739	0.2450
	X	0.2773	13.78	-0.5191	0.2427
SDXL	Drop	0.3664	18.52	-0.0484	0.2624
	Gaussian	0.3486	18.40	-0.0469	0.2634
	X	0.2455	15.34	-0.4483	0.2432

	I-LDM (Gaussian)			(I-LDM (Drop)		
Challenge	ΔCLIP	ΔImageReward	ΔHPSv2	ΔCLIP	ΔImageReward	ΔHPSv2
Basic	-0.0011	0.0657	0.0071	-0.0018	-0.0005	0.0050
Complex	0.0031	0.1566	0.0104	-0.0001	0.1112	0.0093
Fine-grained Detail	0.0023	0.1418	0.0089	0.0009	0.1061	0.0069
Imagination	0.0007	0.2368	0.0096	-0.0052	0.1140	0.0059
Linguistic Structures	-0.0009	0.1290	0.0061	-0.0039	0.0502	0.0037
Perspective	0.0039	0.2436	0.0114	0.0033	0.1588	0.0072
Properties & Positioning	0.0068	0.2810	0.0073	0.0048	0.2685	0.0097
Quantity	-0.0004	0.0092	0.0072	-0.0012	-0.0053	0.0052
Simple Detail	0.0010	0.0659	0.0068	-0.0005	0.0212	0.0043
Style & Format	0.0012	0.2600	0.0108	-0.0023	0.1811	0.0098
Writing & Symbols	0.0033	0.1137	0.0084	0.0008	0.1037	0.0045

(a) Differences by challenge.

	I-LDM (Gaussian)			(I-LDM (Drop)		
Category	ΔCLIP	ΔImageReward	ΔHPSv2	ΔCLIP	ΔImageReward	ΔHPSv2
Abstract	-0.0051	0.0220	0.0087	-0.0032	-0.0410	0.0097
Vehicles	0.0018	0.1519	0.0073	0.0000	0.1017	0.0076
Illustrations	0.0034	0.1499	0.0102	-0.0002	0.1037	0.0090
Arts	-0.0008	0.1321	0.0029	-0.0080	0.0152	0.0064
World Knowledge	0.0018	0.2430	0.0108	-0.0015	0.1706	0.0087
People	0.0008	0.2226	0.0117	-0.0016	0.1230	0.0083
Animals	0.0015	0.1724	0.0085	-0.0005	0.1164	0.0036
Artifacts	-0.0006	0.0423	0.0059	-0.0009	0.0151	0.0037
Food & Beverage	0.0060	0.0612	0.0084	0.0011	0.0091	0.0051
Produce & Plants	0.0004	0.0065	0.0059	-0.0018	-0.0454	0.0029
Outdoor Scenes	0.0034	0.1730	0.0105	0.0017	0.1054	0.0098
Indoor Scenes	0.0003	0.0160	0.0066	0.0003	0.0789	0.0065

(b) Differences by category.

Table 10: Difference in metrics between SD2.1 and I-LDM for each (a) challenge and (b) category in Parti prompt. The red numbers indicate cases where the metric of I-LDM is low, while others represent equal or high metric.

cross-domain self-attention in when the alignment between domains is not perfectly matched. In the example shown in the first row, the blue dot is located at the corner between the wall and the floor in the image. In the generated surface normal map, however, the same position points to the wall instead of the corner. Despite this misalignment, the cross-domain self-attention heatmap is concentrated around the corner region in the surface normal map. Similarly, in the examples shown in the second and third rows, the colored dots are placed on the snowboard and the pot, respectively, in the image, but do not correspond to the same objects in the intrinsic domain. Nevertheless, the heatmaps show that attention is focused on semantically similar regions. These observations support our claim that meaningful information can be exchanged by focusing on semantically similar regions, even when spatial misalignments exist.

G Qualitative Results and Further Analysis in T2I Generation

G.1 Analysis of metrics by category and challenge for Parti prompts

Tab. 10a and Tab. 10b summarize the performance difference between the base model and ours for each challenge and category of Parti prompts. Overall, we observe that the scores of the human preference estimators are substantially higher in all categories and challenges. In particular, scores are maintained or even improved in the Abstract and Art categories, which involve the representation of abstract scenes, and in the Style & Format category, which includes various rendering styles. This suggests that our method preserves the base model's capability to manage general textual conditions.

G.2 Comparisons between Baseline Methods

Fig. 10 illustrates a qualitative comparison between our method and the baseline approaches introduced in the baseline section of Sec. 4. These methods have difficulty resolving spatial inconsistencies and significantly degrade the quality of the base model. Our analysis of each method is as follows:

1. Adding additional input, output channels to the UNet (+CNN, +SpatialAdd): The main reason for the failure of simply adding CNN channels is that, unlike the base model, our work also requires the generation of intrinsics. Since changes to the up-block representations are necessary, the quality of the image domain degrades by simply fine-tuning the T2I model. The same problem also occurs with +SpatialAdd. Unlike ControlNet, which only uses intrinsics as inputs, generating intrinsics requires changes in Up-block representations. Thus, fine-tuning the final convolution layer or the whole Up-block network is required, which inevitably degrades image quality. We report the results of fine-tuning only the final convolution layer in Tab. 1, as this approach yields relatively better performance.

2. Fine-Tuning UNet with a joint image-depth VAE (LDM3D, LDM3D+): LDM3D trains a VAE to jointly encode and decode images and depth maps, followed by fine-tuning a UNet model in this latent space. We hypothesize two possible scenarios depending on the VAE training: First, if the depth input minimally influences the latent representation—meaning the latent space does not effectively encode depth information—and the decoder is trained to predict depth aligned with the image,

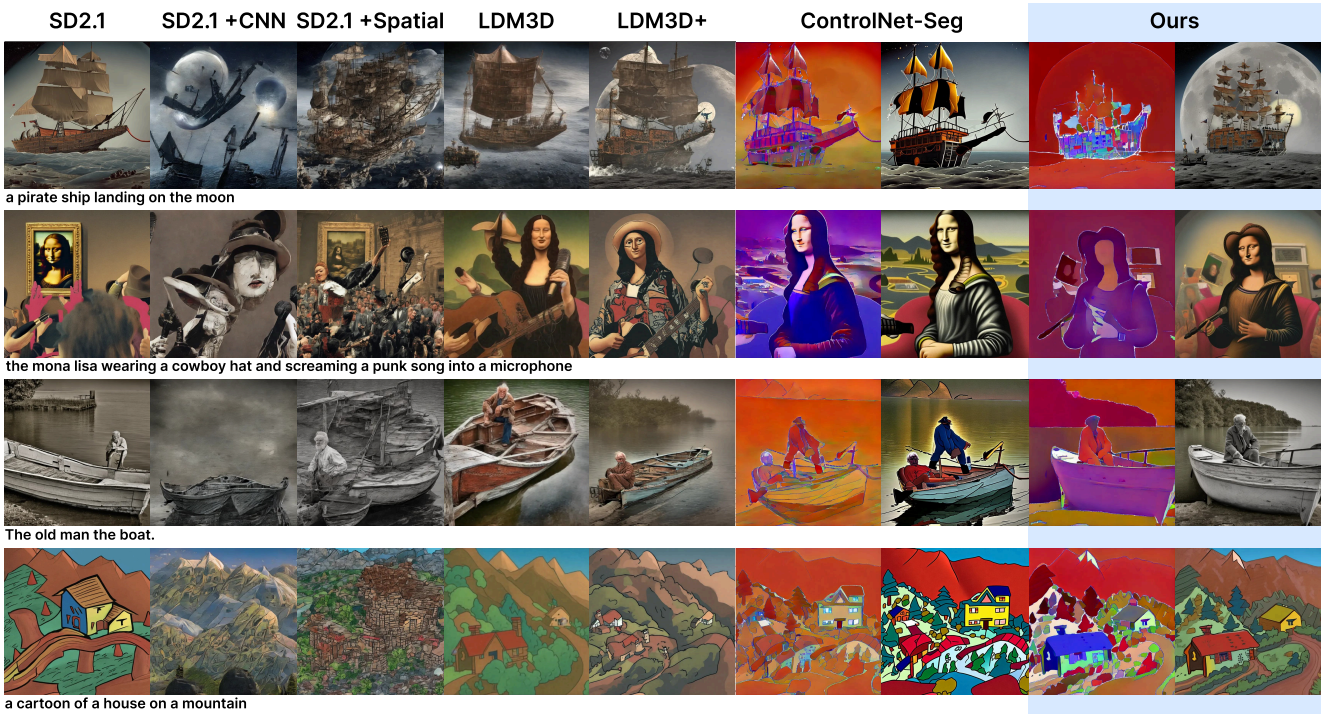


Figure 10: Visualizations of samples from baseline approaches, SD2.1, SD2.1 +CNN, SD2.1 +SpatialAdd, LDM3D, LDM3D+, ControlNet-Seg with generated segmentation map, and ours.

the latent space remains largely unchanged. While this may help prevent quality degradation, the insufficient encoding of depth in the latent limits the denoising process from learning meaningful depth features. Alternatively, if the encoder integrates depth information and produces a disentangled latent space, this setup may resemble the +CNN experiment, potentially leading to quality degradation, as suggested by the results in Fig. 10.

3. 2-stage generation using ControlNet (ControlNet-Norm, Seg, All): We report the results of using single, and all four intrinsics (depth, normal, segmentation, line) as conditions by feeding each intrinsic into corresponding ControlNet for conditional image generation, as shown in Tab. 1. For single intrinsic conditions, we report surface normal and segmentation map results since it yielded relatively better performance compared to using depth map and line drawings. The main drawback of the 2-stage generation ($c \rightarrow i \rightarrow z$) is that $c \rightarrow i$ cannot leverage image domain information, causing inconsistencies which propagate through x. Fig. 10 visualizes segmentation maps generated without image information, showing that while object boundaries are well distinguished, the lack of image or other intrinsic information leads to unnatural spatial layouts and overall unrealistic scene modeling. When these segmentation maps are used as conditions for ControlNet-based image generation, spatial inconsistencies still occur. The qualitative degradation of intrinsic quality is also evident in the LoRA-only ablation in Tab. 5. By contrast, in our method, since images and intrinsics are modeled complementarily, both intrinsics and images are generated with greater spatial consistency.

In summary, baseline methods either fail to incorporate meaningful information from the intrinsic domain into images, or significantly harms the quality of the base model.

G.3 Qualitative Results

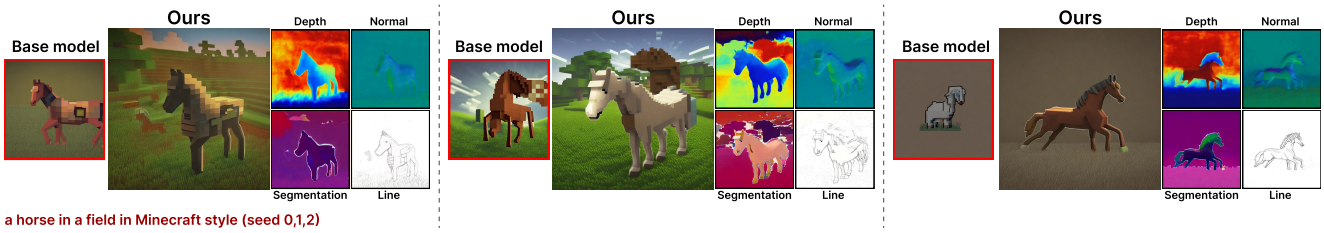


Figure 11: Examples of samples generated from the same prompt (a horse in a field in Minecraft style) with different seeds.

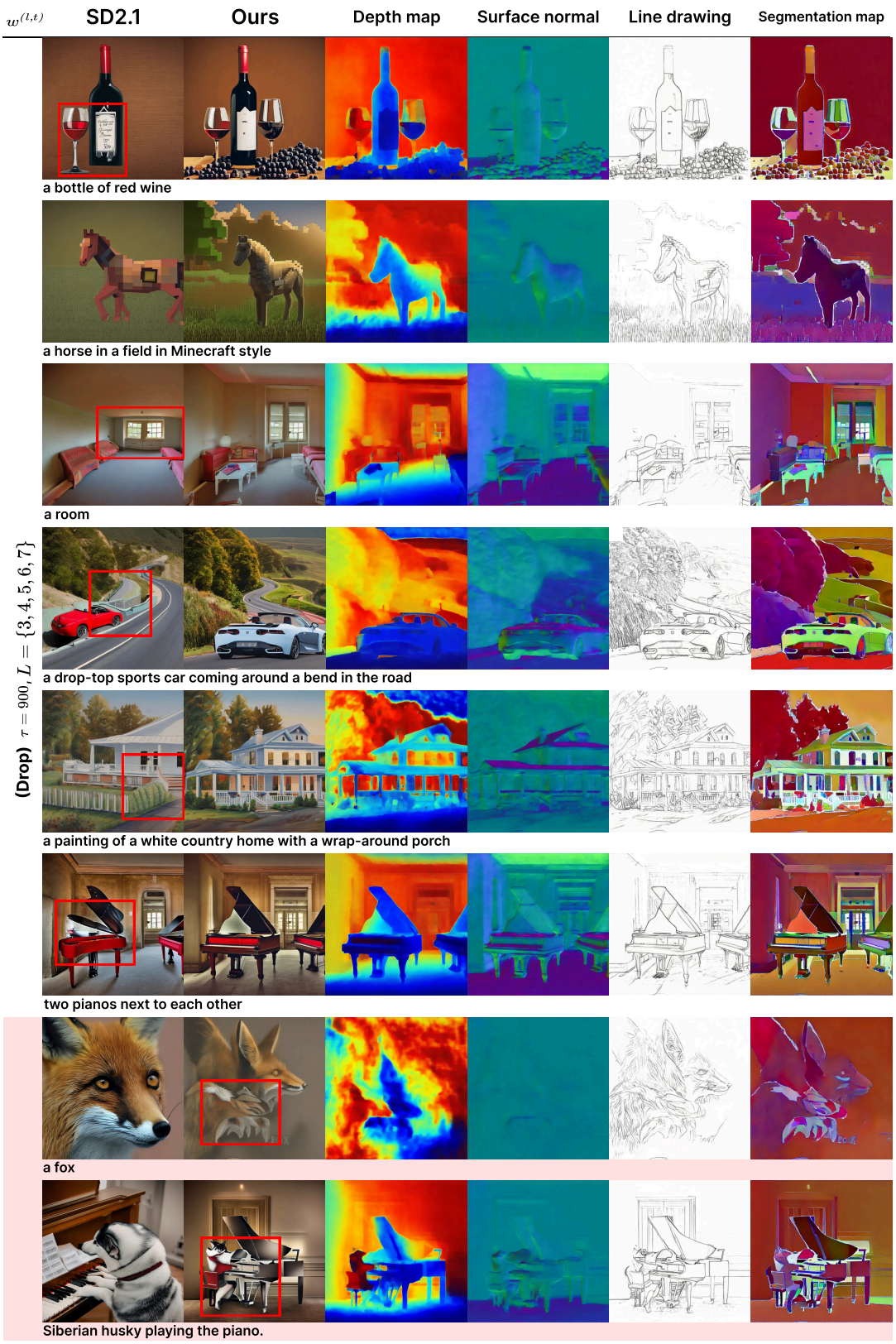


Figure 12: Examples generated from Parti prompts using SD2.1 and ours with drop scheduling. The red box highlights spatial inconsistencies. The top six rows illustrate successful examples of I-LDM, while the bottom two rows illustrate failure cases.

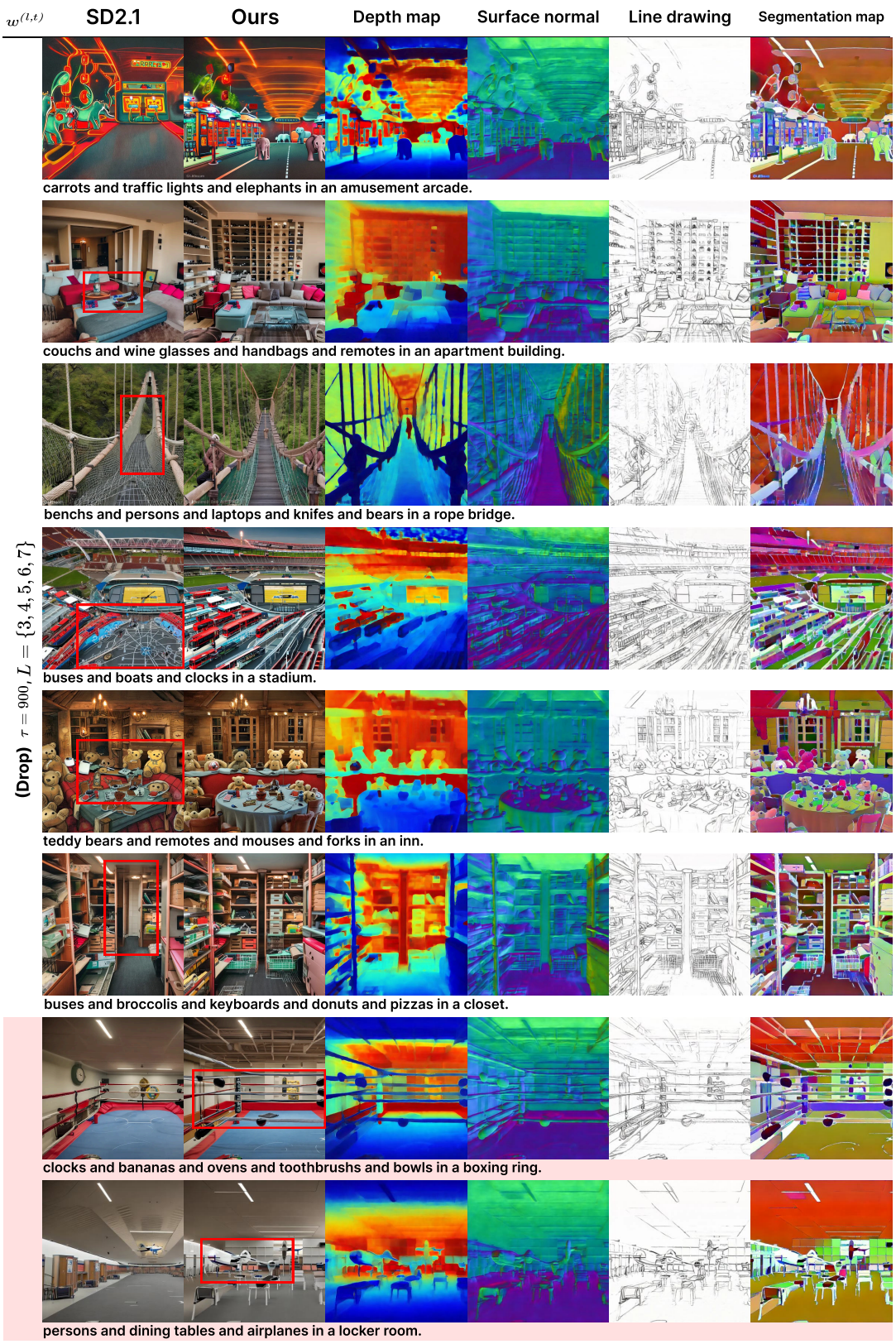


Figure 13: Examples generated from Multi prompts using SD2.1 and ours with drop scheduling. The red box highlights spatial inconsistencies. The top six rows illustrate successful examples of I-LDM, while the bottom two rows illustrate failure cases.

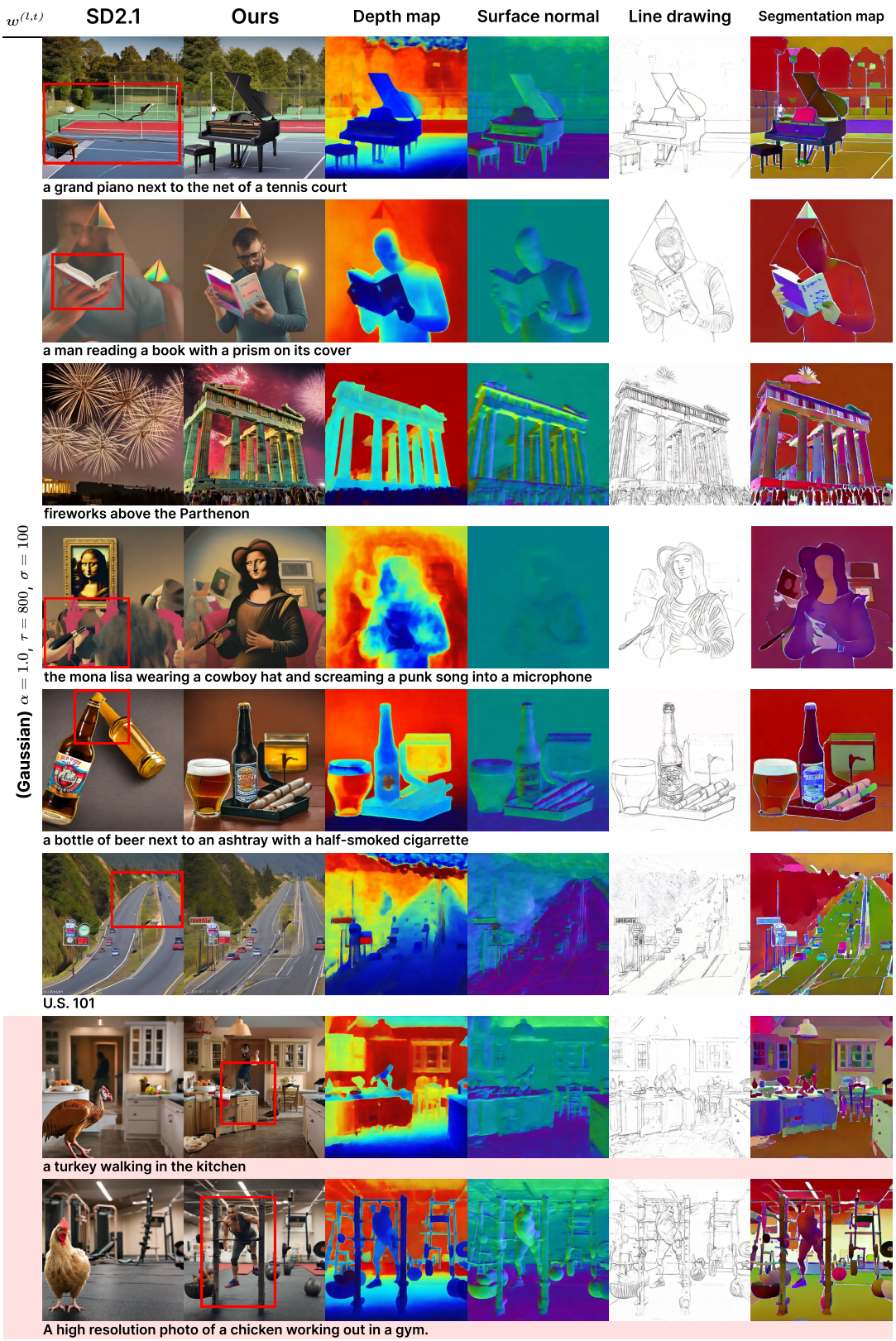


Figure 14: Examples generated from Parti prompts using SD2.1 and ours with Gaussian scheduling. The red box highlights spatial inconsistencies. The top six rows illustrate successful examples of I-LDM, while the bottom two rows illustrate failure cases.

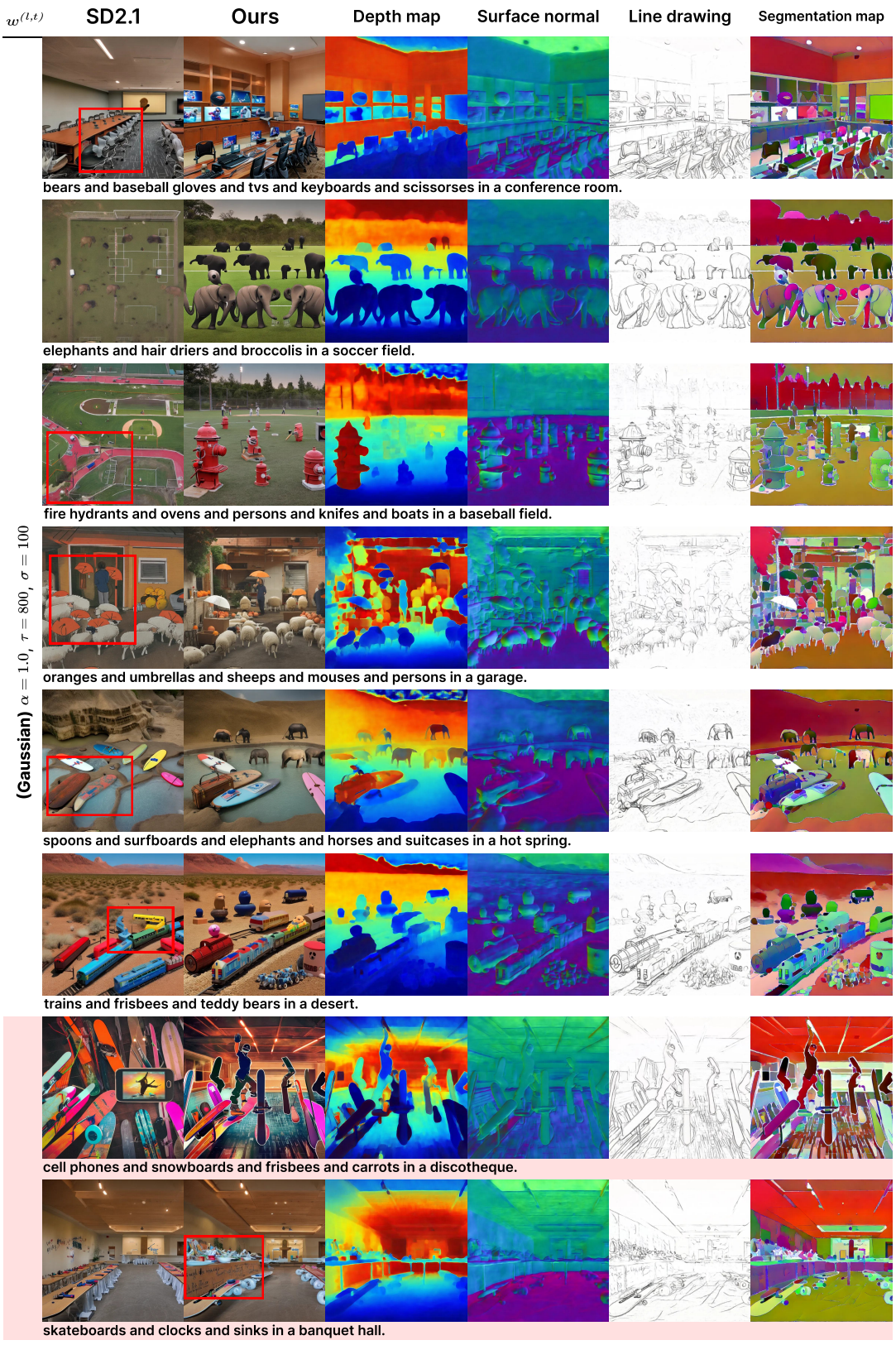


Figure 15: Examples generated from Multi prompts using SD2.1 and ours with Gaussian scheduling. The red box highlights spatial inconsistencies. The top six rows illustrate successful examples of I-LDM, while the bottom two rows illustrate failure cases.

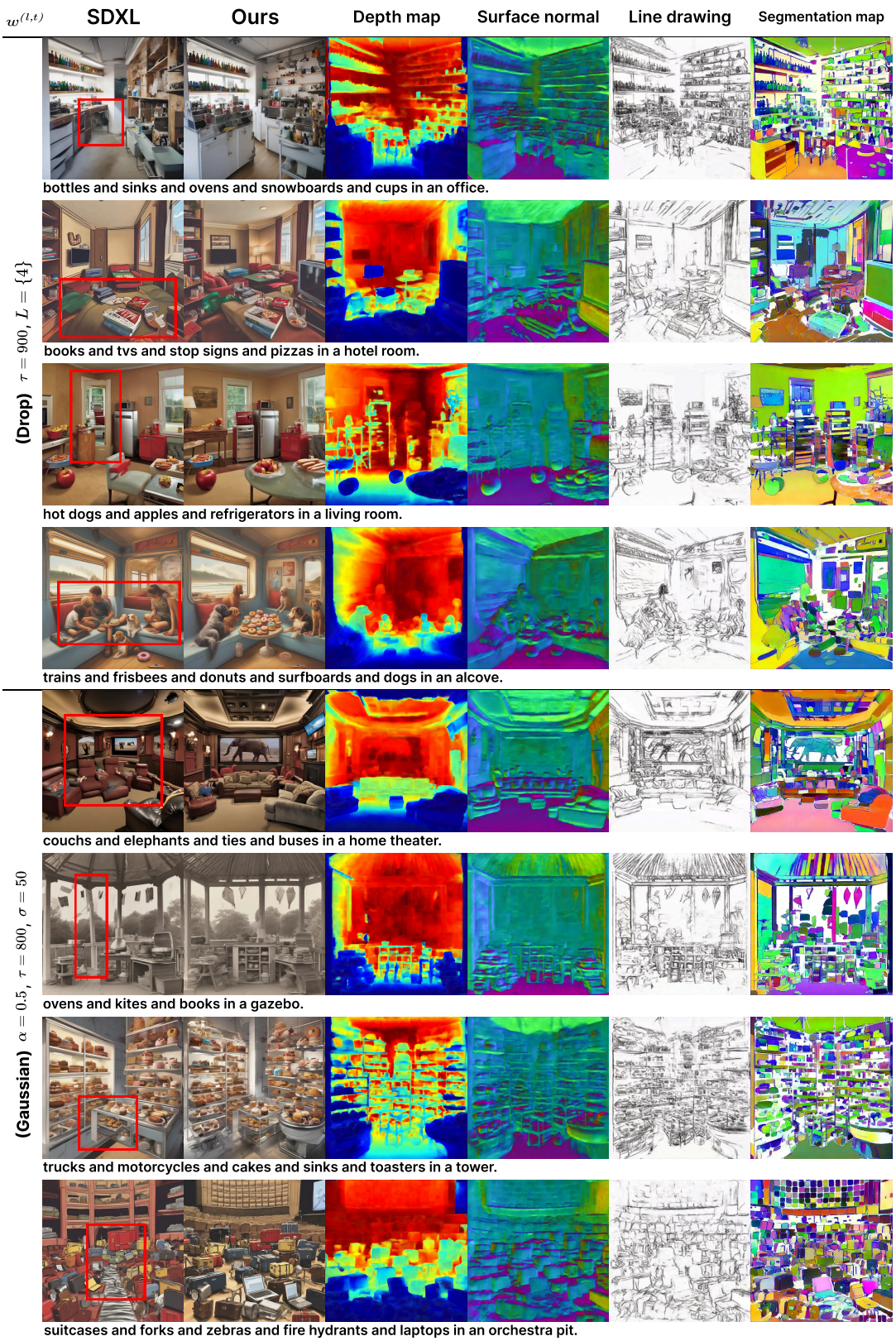


Figure 16: Examples generated from Multi prompts using SDXL and ours with drop scheduling (top 4 rows) and Gaussian scheduling (bottom 4 rows). The red box highlights spatial inconsistencies.

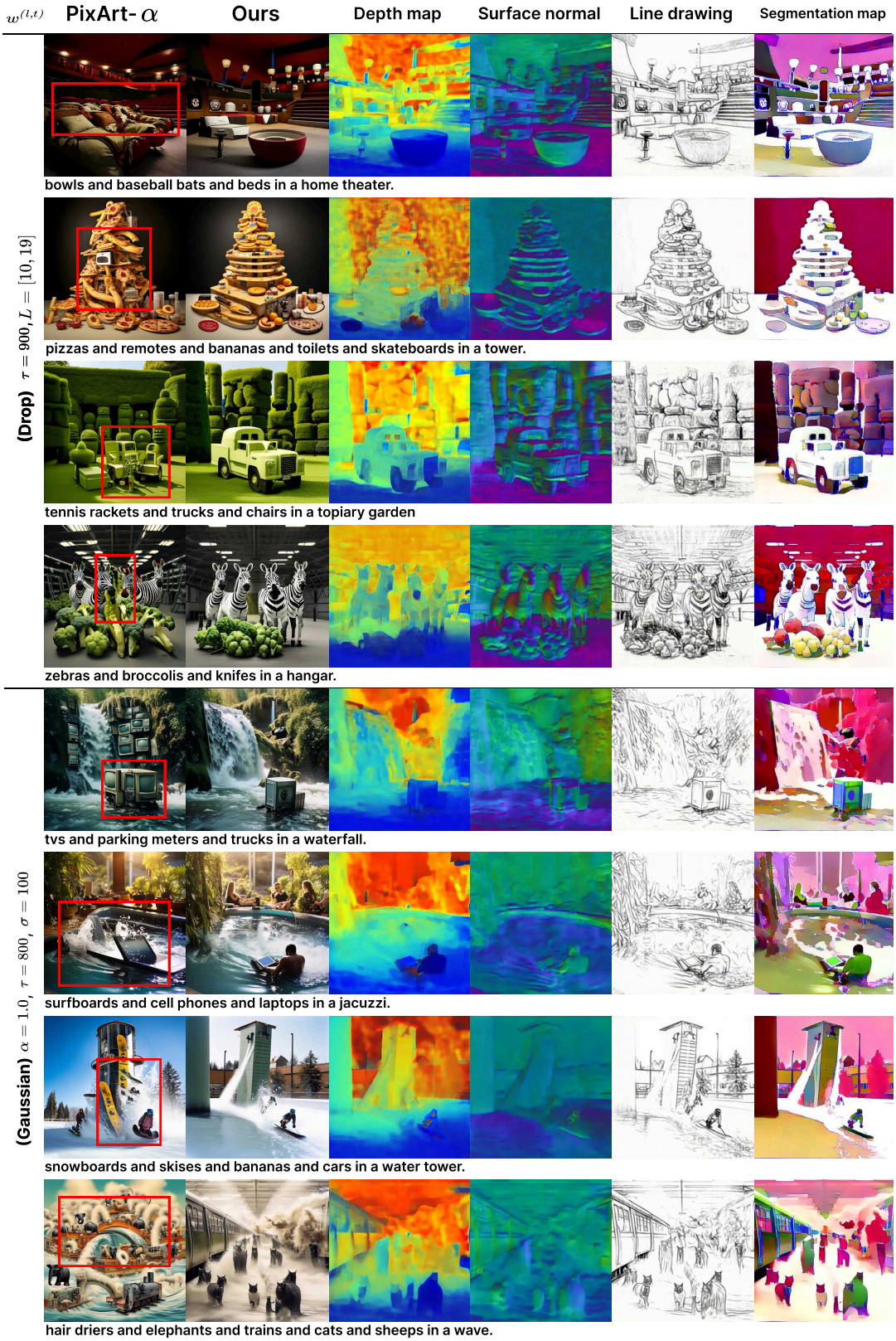


Figure 17: Examples generated from Multi prompts using PixArt-α and ours with drop scheduling (top 4 rows) and Gaussian scheduling (bottom 4 rows). The red box highlights spatial inconsistencies.

H InterHand Experiment Details and Results

H.1 Experimental details

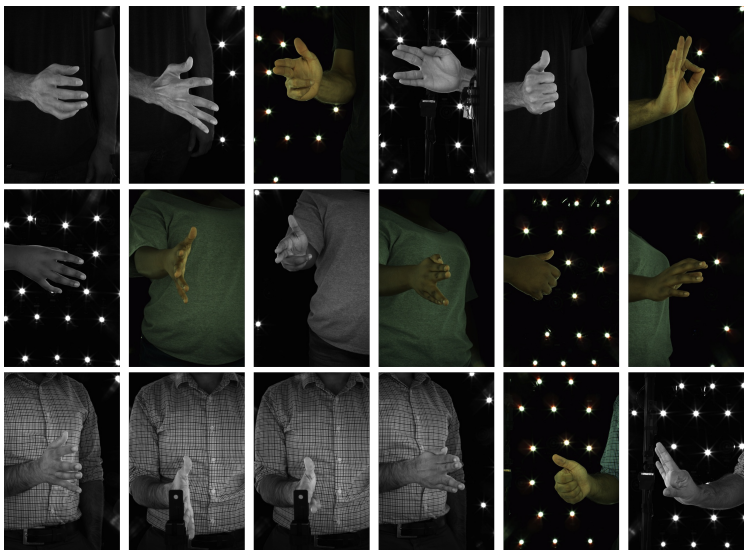


Figure 18: Visualization of samples from the InterHand2.6M dataset, where each row corresponds to the same recording and each column represents the same gesture.

The InterHand2.6M dataset (Moon et al. 2020) contains 2.6 million frames of single and interacting hands along with annotations, while we use only a very small subset of samples from this dataset. Specifically, to preserve the diversity of gestures and subjects, we select the middle frame from a randomly selected camera for each of the total 3,493 gestures contained in 36 recordings. We fine-tune each base model using a rank-8 LoRA applied to the attention layers, training with a batch size of 8 for 3,000 steps. The LoRA trained on the base model is then applied identically to the image domain in I-LDM for evaluation.

H.2 Qualitative results

Fig. 19 presents visualizations of successful examples from I-LDM using Gaussian sampling. Both models represent the geometric structures of each finger in the surface normals, while the line drawing and segmentation map delineate the boundaries of each finger. By incorporating these details into the images, they more clearly represent the complex structure of the hand.

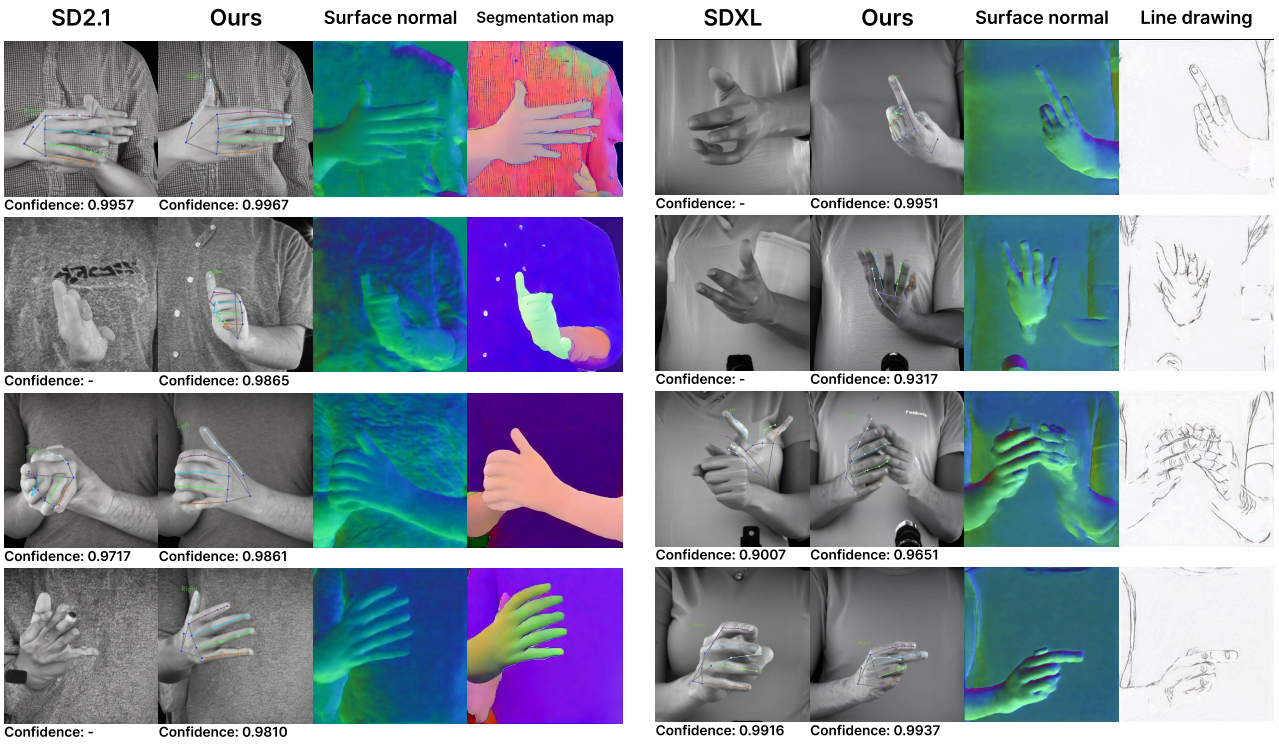
I Qualitative Results for Ablating Intrinsics

Fig. 20 visualizes the effects of each intrinsics constituting I-LDM. When the intrinsics representing the geometric structure of the scene—depth map and surface normal—are removed, the resulting images have a reduced sense of depth and an inaccurate representation of object shapes. Also, when the intrinsics that convey the semantic boundaries between objects in the scene—line drawing and segmentation map—are removed, the representation of object boundaries becomes less precise, and similar textures tend to merge together.

J Qualitative Results and Analysis for Ablating Training Settings

We report the results of intrinsics and images generated under ablated training settings in Tab. 5. "-$w^{(l,t)}$" refers to the experiment where we sample images without scheduling weights by fixing $w^{(l,t)} = 1$. "-VAE" denotes the experiment without intrinsic VAE, where we use a depth map encoded by a pre-trained VAE. "LoRA-only" represents the results of training LoRA weights only from the intrinsic domain without sharing information from the image domain. "+w_{train}" refers to the experiment where we also use $w_g^{(l,t)}$ during training, and "+resolution" denotes training our model at a higher resolution of 768×768. For visualizations of "-$w^{(l,t)}$", please refer to Fig. 5.

First, when we train the model to generate depth encoded by the pre-trained VAE instead of using an intrinsic VAE, we observe that the generation of intrinsics becomes highly unstable and of poor quality. As a result, the images fail to extract useful information from the intrinsics. Second, when we train only the LoRA weights for the intrinsic domain without incorporating cross-domain self-attention from the image domain, the quality of the generated intrinsics degrades, which also affects the images. Finally, applying scheduling to the image domain during training improves the quality of the generated intrinsics. However, this leads to weaker alignment between the images and the intrinsics, resulting in a stronger tendency to generate spatially inconsistent images.



(a) Results based on SD2.1

(b) Results based on SDXL

Figure 19: Visualizations of successful examples of I-LDM in hand generation tasks.

K Limitations and Future Works

While our work reveals the potential of incorporating intrinsics into generating images, there is still room for improvement in performance and computational efficiency. Typically, our LoRA-based learning method, which corresponds to approximately 1% of the denoising network parameter, can effectively preserve the image quality and diversity of the base model, but there remains potential to improve the quality and alignment of the generated intrinsic. To address this issue, co-generating images and intrinsics within a single model trained from scratch would be a promising future direction. Furthermore, we believe that increasing the variety of intrinsics and using training data with more complex structures will improve the effectiveness of our approach. Finally, we expect that enriching quantitative measurements of spatial inconsistency would be a promising future work. Also, investigating the advantages of incorporating intrinsics in various downstream tasks such as image classification (Lee et al. 2023) would be a promising direction, which could be further combined with additional data pre-processing such as data augmentations (Shorten and Khoshgoftaar 2019; Lee et al. 2022; Hwang et al. 2022).

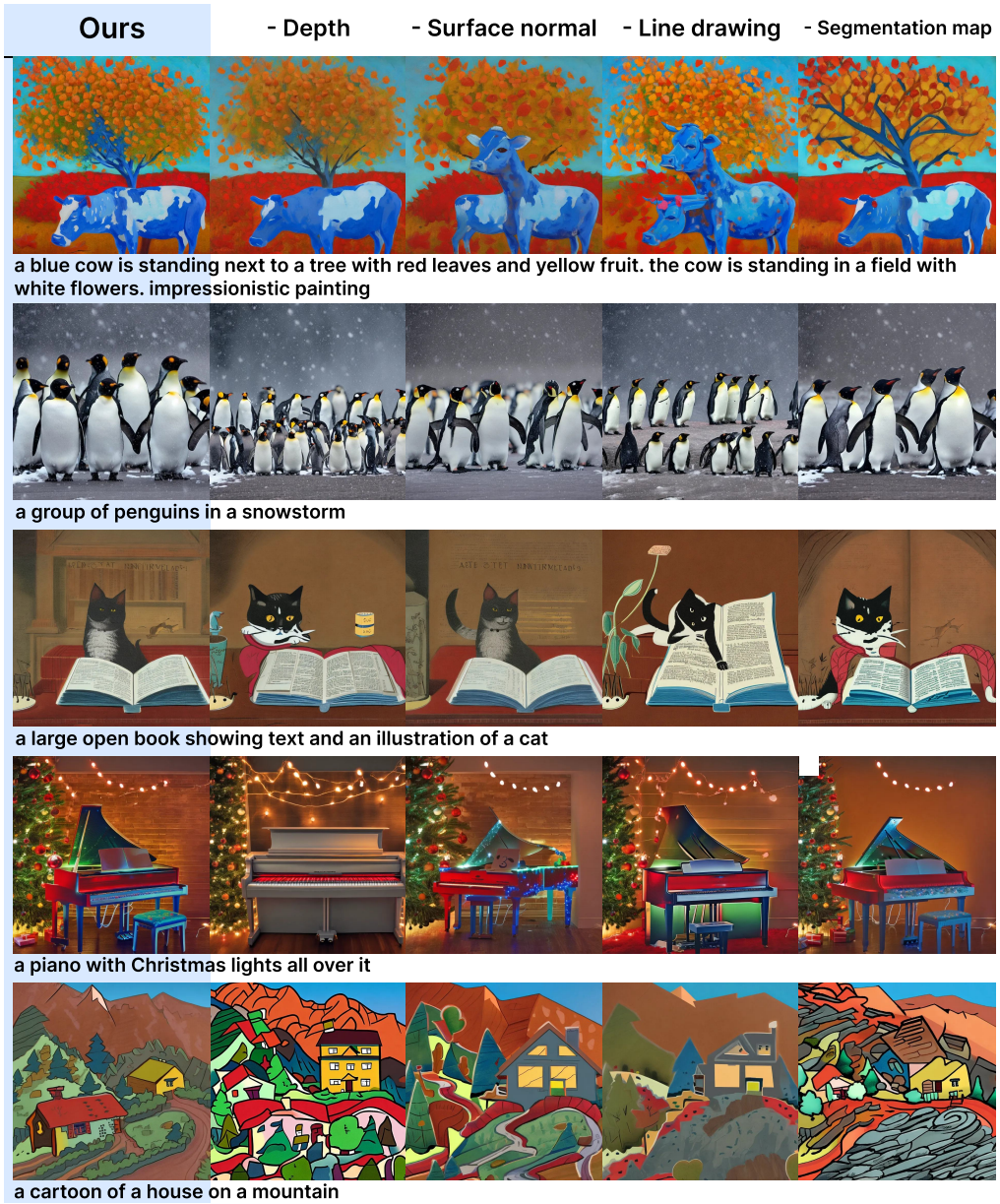


Figure 20: Visualization of the ablation results for each intrinsic component constituting I-LDM.

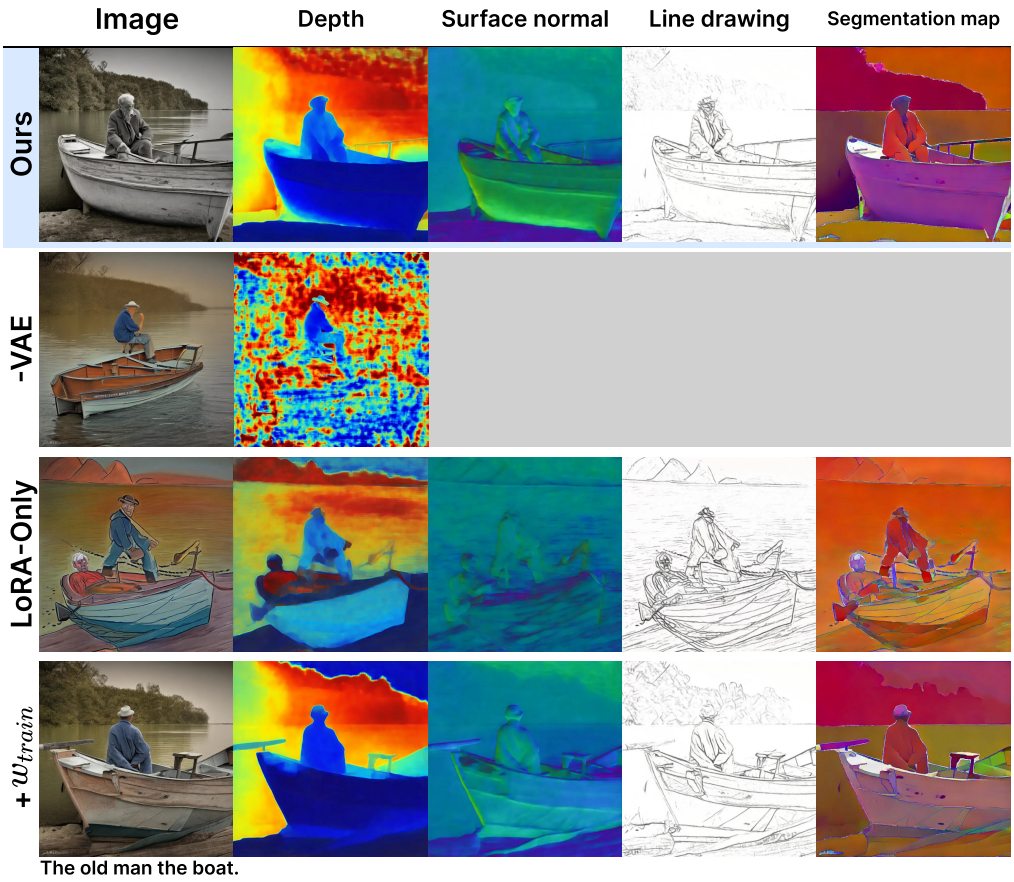


Figure 21: Visualization of the ablation results corresponding to Tab. 5. For the results of "-$w^{(l,t)}$" please refer to Fig. 5.