DECODING WITHOUT A DECODER: DIRECT PIXEL PREDICTION FROM DIFFUSION TRANSFORMERS

Hansheng Chen[1,2] **Shengqu Cai**[1] **Jan Ackermann**[1] **Rayhan Zirvi**[1,3]
Gordon Wetzstein[1] **Xinchen Yan**[*] **Leonidas Guibas**[1]

[1]Stanford University [2]Luma AI [3]California Institute of Technology

https://hanshengchen.com/dwd



Figure 1: **Given the same seed, DWD preserves similar content to the original model while improving fine detail. (a)** On latent-space video diffusion, DWD removes the decoder and keeps text rendering legible across frames, whereas the baseline distorts it. **(b)** On pixel-space image diffusion, DWD recovers photorealistic skin and hair texture that the baseline oversmooths.

ABSTRACT

Latent diffusion models rely on a decoder to map denoised latents to pixels, adding complexity and inference cost. We introduce *Decoding Without a Decoder* (DWD), which lets a finetuned diffusion transformer produce final pixels directly through a linear head. Unlike previous methods, where the denoising output leads to final pixels, DWD separates the two roles: the original denoising head continues to drive sampling, then the new DWD head predicts the pixels at the final step. We extend pretrained latent models with DWD efficiently by finetuning only the last few transformer blocks, training the DWD head with pixel quality losses while anchoring the denoising head to the pretrained model to preserve the sample trajectory. Across latent image and video generators, DWD removes the decoder while maintaining or even improving pixel quality. On Wan video generation, it visibly reduces detail distortion and flickering, lowering warping error by 23%. Remarkably, the same separation also improves pixel diffusion models that do not even use a decoder: a final-step DWD head produces substantially better texture detail in text-to-image generation, and improves ImageNet FID from 1.55 to 1.43, a new state of the art among pixel flow models.

[*]Work done while at Luma AI.

1 INTRODUCTION

Latent diffusion (Rombach et al., 2022; Ho et al., 2020; Song et al., 2021; Lipman et al., 2023) is the dominant recipe for image and video generation (Esser et al., 2024; Black Forest Labs, 2024; Wang et al., 2025; Kong et al., 2025; HaCohen et al., 2024; Z-Image Team et al., 2025; Black Forest Labs, 2025). Typically, a diffusion transformer (DiT) (Peebles & Xie, 2023) generates in a compressed latent space, and a convolutional decoder (Kingma & Welling, 2014; Esser et al., 2021) maps the latents to pixels. The decoder complicates deployment and can add substantial inference cost. For example, on H100, generating a 5-second video with a 4-step Wan 1.3B model (Zheng et al., 2026b) takes 2.8 seconds for sampling and another 1.9 seconds for decoding, adding about 70% overhead. Recent work obtains much faster decoders with comparable quality (Zhu et al., 2026; Kang & Kwak, 2026; Li et al., 2026; Gu et al., 2026), suggesting that the decoder is largely redundant. This raises the question: what is the minimal network architecture for producing final pixels from a latent diffusion model?

We show that a simple linear head can replace the decoder while improving pixel quality. Our method, *Decoding Without a Decoder* (DWD), finetunes a pretrained latent DiT by adding a linear output head (the *DWD head*) in parallel with the DiT's original output layer (the *denoising head*). The denoising head continues to drive denoising steps, then the DWD head predicts the pixels at the final step only.

The dual-head separation distinguishes DWD from previous approaches, whose denoising output leads to final pixels: latent models decode it, pixel models output it directly. This couples pixel fidelity to denoising, causing a tradeoff between detail rendering and denoiser learnability (Yao et al., 2025; Skorokhodov et al., 2025). DWD decouples the two roles: the DWD head is trained with pixel quality losses (Esser et al., 2021; Zhang et al., 2018), while the denoising head is anchored to the frozen pretrained model to preserve sample content.

The separation is most evident on pixel diffusion models, where the denoising head already predicts the final output. These predictions smooth fine detail because they estimate conditional means under ℓ_2 loss. Applying perceptual losses directly to them (Lin & Yang, 2023; Berrada Ifriqi et al., 2025; Ma et al., 2026b) risks collapsing sample diversity, as our shared-head ablation fails to improve FID (Table 1). In contrast, a separate head improves ImageNet FID from 1.55 to 1.43.

We evaluate DWD across latent and pixel generators on ImageNet, text-to-image, and video generation (Fig. 1). Removing the decoder improves nearly all FID metrics across tested latent image models. On Wan (Wang et al., 2025) video, warping error drops by 23%, with visibly less detail distortion and flickering. On text-to-image pixel diffusion, DWD recovers photorealistic textures.

To summarize, we show that:

- A latent DiT needs no decoder: finetuning its last few blocks with an additional linear head suffices, reducing inference cost, improving ImageNet FIDs, and lessening detail distortion and flickering in Wan video.
- The dual-head separation also improves pixel diffusion models, producing better texture detail in text-to-image generation and a state-of-the-art ImageNet FID of 1.43 among pixel flow models.

2 RELATED WORK

Latent-space diffusion. VQGAN (Esser et al., 2021) trains autoencoders with reconstruction, perceptual (Zhang et al., 2018), and adversarial (Goodfellow et al., 2014) losses, and LDM (Rombach et al., 2022) uses this perceptual compression to reduce diffusion cost. VA-VAE (Yao et al., 2025), REPA-E (Leng et al., 2025), and RAE (Zheng et al., 2026a) address the gap between reconstruction quality and diffusability. DC-AE (Chen et al., 2025) and Qwen-Image-VAE-2.0 (Zhang et al., 2026) increase spatial compression while preserving reconstruction fidelity. SSDD (Vallaeys et al., 2025) and PiD (Lu et al., 2026) replace conventional decoders with distilled pixel diffusion models. PiD and LTX-Video (HaCohen et al., 2024) also stop latent sampling early, letting the decoder complete the remaining denoising. Zhang et al. (2024) improve latent generators by applying losses to images produced through the VAE decoder. These approaches retain a standalone decoding network. DWD instead produces the final pixels through a linear head on the existing DiT, reusing its last-layer tokens and computation without an extra decoder.

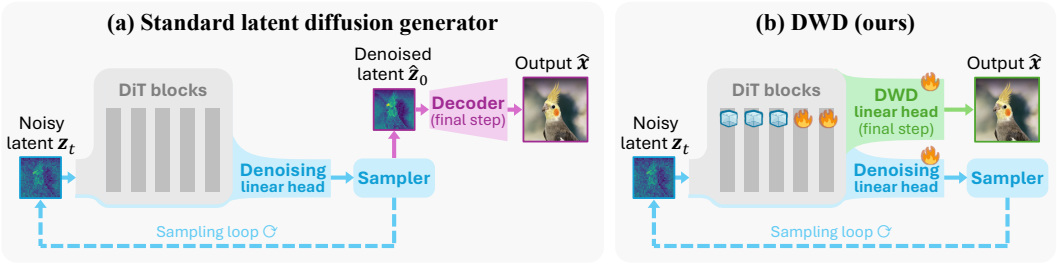


Figure 2: **Standard latent generation vs. DWD. (a)** A standard generator samples a denoised latent and maps it to pixels with a decoder. **(b)** DWD retains the denoising head for latent denoising and adds a parallel DWD head that produces the final pixels only at the final step, bypassing this decoder. DWD finetunes the pretrained generator's final blocks and heads (fire icon), while freezing all previous blocks (ice icon). Time t and conditioning $\boldsymbol{c}$ are omitted for clarity.

Pixel-space diffusion. Pixel-space diffusion models denoise directly in pixel space without an autoencoder. They overcome difficulties in high-resolution generation through multiscale architectures and specialized output pathways (Hoogeboom et al., 2023; Crowson et al., 2024; Yu et al., 2026; Wang et al., 2026b; Ma et al., 2026a), or improved prediction targets (Li & He, 2026; Chen et al., 2026a; Zirvi et al., 2026). AsymFlow (Chen et al., 2026a) and L2P (Chen et al., 2026d) finetune pretrained latent models to perform denoising in pixel space, altering the flow trajectory and sample content (see Appendix B.4 for a comparison). In contrast, DWD preserves the original latent trajectory rather than converting the model to pixel diffusion. DWD finetuning is also more efficient, since only the last few transformer blocks and output heads are updated with shallow backpropagation, whereas latent-to-pixel conversion must backpropagate through the entire network to adapt to pixel-space inputs.

3 Decoding Without a Decoder

A pretrained latent DiT's last-layer tokens already encode rich pixel information (Gan et al., 2025; Akarken et al., 2026; Wang et al., 2026c): a least-squares linear projection recovers most of the image structure before any finetuning (Fig. 3). DWD exploits this by appending a linear head that produces final pixels directly from these tokens, bypassing the decoder at inference (Fig. 2(b)). Separating the pixel readout from the denoising output lets each be trained with its own objective: pixel quality losses for the DWD head, and trajectory anchoring to preserve denoising behavior.

3.1 Preliminaries

We describe latent diffusion (Rombach et al., 2022) using the flow matching convention (Lipman et al., 2023). A pretrained autoencoder encodes an image or video $\boldsymbol{x}$ into a latent representation $\boldsymbol{z}_0$. The forward process mixes the latent $\boldsymbol{z}_0$ with Gaussian noise to form $\boldsymbol{z}_t = (1-t)\boldsymbol{z}_0 + t\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $t \in [0, 1]$ denotes diffusion time. A flow model $(\boldsymbol{z}_t, t, \boldsymbol{c}) \mapsto \hat{\boldsymbol{u}}$ trains a network with extra conditioning $\boldsymbol{c}$ to estimate the conditional mean of velocity $\boldsymbol{u} := \boldsymbol{\epsilon} - \boldsymbol{z}_0$, using an ℓ_2 flow matching loss. At inference, the model uses this velocity to progressively transform Gaussian noise into a denoised latent sample $\hat{\boldsymbol{z}}_0$, running from $t = 1$ toward $t = 0$ (Song et al., 2021; Liu, 2022; Albergo & Vanden-Eijnden, 2023). Finally, the decoder maps $\hat{\boldsymbol{z}}_0$ to the final image or video $\boldsymbol{x}$.

For the models studied here, the denoising network is a diffusion transformer (DiT) (Peebles & Xie, 2023; Esser et al., 2024), which processes tokenized noisy latent $\boldsymbol{z}_t$ with multiple transformer blocks (Vaswani et al., 2017), and feeds the last-layer tokens $\boldsymbol{h}$ (after normalization and modulation) to a linear output head that produces a patch-wise denoising prediction. This is the *denoising head*, shown in Fig. 2(a).

3.2 DWD head

Our goal is to produce final pixels from a pretrained latent DiT without a decoder, subject to two constraints: minimal additional computation, and preservation of the pretrained denoising trajectory.

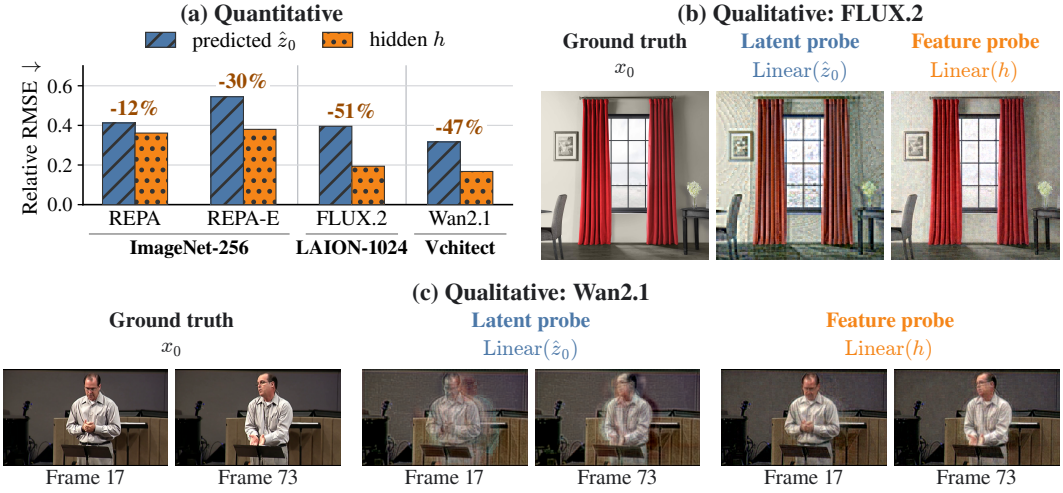


Figure 3: **Last-layer tokens support more accurate linear pixel reconstruction.** We fit patch-wise linear probes from denoised latents $\hat{\boldsymbol{z}}_0$ and last-layer tokens $\boldsymbol{h}$ to ground-truth pixels, keeping the pretrained models frozen. **(a)** Held-out relative RMSE, with percentages showing error reduction over the latent probe. **(b, c)** Probes of last-layer tokens preserve finer image detail and reduce video ghosting before finetuning.

A linear probe shows that the last-layer tokens $\boldsymbol{h}$ already support pixel reconstruction without finetuning, more accurately than probing the denoised latents $\hat{\boldsymbol{z}}_0$, as shown in Fig. 3. The remaining reconstruction mismatch can be addressed by finetuning the DiT. This inspires us to adopt a simple linear *DWD head* that maps each token to a pixel patch:

$$\hat{\boldsymbol{x}}_i = \mathrm{Linear}(\boldsymbol{h}_i) = \boldsymbol{W}\boldsymbol{h}_i + \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are learnable weights and bias, and $\hat{\boldsymbol{x}}_i$ is the pixel patch corresponding to token $\boldsymbol{h}_i$. This design places the DWD head in parallel with the denoising head, both reading the same last-layer tokens. The linear head is the minimal form of this separate pixel output, which can optionally be extended with a lightweight convolutional refiner (Appendix A.1). The predicted patches are then assembled into the final image or video $\hat{\boldsymbol{x}}$, with color space conversion where needed (Appendix A).

Efficiency. By reusing the DiT's nonlinear computation, the linear head is a lightweight readout with negligible cost for common pixel patch sizes. Pixel patch size follows the combined downsampling factor of the autoencoder and tokenizer. The most common spatial factor of 16 gives a 16×16×3 patch per token. For video models, an additional temporal factor of 4 expands each patch to four frames. For most tested models, the total number of output dimensions per token stays below the DiT's hidden dimension.

Separating denoising and pixel outputs. DWD achieves trajectory preservation and pixel rendering through a clear separation of outputs: the DWD head is trained with pixel quality losses, while the denoising head drives sampling and is anchored to the pretrained model (Sec. 3.3). This dual-head separation distinguishes our work from prior diffusion-based image and video generators, where the denoising output serves as a bottleneck pathway to final pixels.

Final-step pixel rendering. The DWD head is only used at the final denoiser evaluation step, where the DiT input t is small but nonzero. The nonzero timestep is important: treating DiT–DWD as a latent decoder at $t{=}0$ substantially worsens quality (Sec. 4.1), while the $t{>}0$ noise gives the pixel head freedom to create detail that the denoised latent would have smoothed away. Classifier-free guidance (Ho & Salimans, 2021) is not used in the final step since its role is to steer the trajectory, whereas our final step does not advance the trajectory.

3.3 FINETUNING

We finetune the last K transformer blocks (typically $K{=}4$–8 for large text-conditioned models, see Appendix A for details), both output heads, and the last modulation, keeping the prefix frozen. The

timestep embedding MLP is shared across all blocks, so we duplicate its weights for the adapted blocks, allowing their modulations to be fully tuned without disturbing the frozen prefix. This restricts backpropagation to the end of the network and retains most pretrained parameters, making training up to 3× faster than full backpropagation.

The training objective combines pixel quality with trajectory anchoring:

$$\mathcal{L} = \mathcal{L}_{\text{pix}} + \lambda \mathcal{L}_{\text{traj}}, \tag{2}$$

where λ is the trajectory anchoring weight (Appendix A). Both terms use noisy inputs $\boldsymbol{z}_t$ encoded from real image data $\boldsymbol{x}$, but their timesteps t are sampled separately to match the roles of the two outputs. The two losses are backpropagated before one optimizer step on accumulated gradients, so the last-layer tokens adapt for pixel quality while the denoising head's prediction remains anchored to the pretrained model. The full training loop is shown in Algorithm 1.

Pixel quality. The DWD head predicts the clean image under the L1 reconstruction, LPIPS perceptual, and adversarial objectives commonly used to train autoencoders (Esser et al., 2021; Rombach et al., 2022; Zhang et al., 2018; Goodfellow et al., 2014):

$$\mathcal{L}_{\text{pix}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{c},t\sim p_{\text{pix}}}\Big[\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 + \text{LPIPS}(\hat{\boldsymbol{x}}, \boldsymbol{x}) + \ell_{\text{adv}}(\hat{\boldsymbol{x}})\Big], \tag{3}$$

where p_{pix} is typically uniform over $[0.0, 0.1]$ before timestep shifting (Esser et al., 2024), matching the small-t regime where the DWD head operates at inference. Following Zheng et al. (2026a) and Singh et al. (2026), the adversarial term uses the StyleGAN-T discriminator (Sauer et al., 2023) with adaptive gradient balancing, and is switched on only in the last stage of training (Appendix A).

Trajectory anchoring. To preserve the pretrained denoising behavior as the final blocks adapt, we match the denoising head's prediction to that of a frozen copy of the original model. Let $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{u}}^{\text{ref}}$ denote the adapted and frozen reference velocity predictions for the same input. We minimize

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{c},t\sim p_{\text{traj}}}\Big[\big\|\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}^{\text{ref}}\big\|_2^2\Big], \tag{4}$$

where p_{traj} is typically uniform over $[0.04, 1.0]$ before timestep shifting, covering the intermediate denoising steps. Both the original and finetuned models share a single forward pass through the frozen prefix: only the final K blocks and output components run separately, avoiding two full-model passes.

Algorithm 1: DWD finetuning with typical t choice.

```
for x_gt, z, cond in data:
    model.zero_grad(); discr.zero_grad()
    # == Trajectory loss with shared prefix ==
    t = shift(uniform(0.04, 1.0))
    with no_grad():
        cache = frozen_prefix(
            add_noise(z, t), t, cond)
        u_ref = original_tail(cache)
    u, _ = tuned_tail(cache)
    loss_traj = weight * mse(u, u_ref)
    loss_traj.backward()
    # ==== Pixel loss at small timesteps =====
    t = shift(uniform(0.0, 0.1))
    with no_grad():
        cache = frozen_prefix(
            add_noise(z, t), t, cond)
    _, x = tuned_tail(cache)
    # loss_pix ← L1 + LPIPS + l_adv
    loss_pix = pixel_loss(x, x_gt)
    loss_pix.backward()
    # ========== Discriminator loss ==========
    loss_D = discr_loss(x.detach(), x_gt)
    loss_D.backward()  # Only for discr update
    # ============ Optimizer step ============
    model.step(); discr.step()
```

Table 1: **Results on ImageNet 256×256.** Shaded rows are our DWD extensions, finetuned from the baseline above them. Italic rows are ablations: *DWD-conv* adds convolutional refinement, and *w/o separate head* fuses the DWD and denoising heads so that a single head receives both pixel losses and trajectory anchoring. DWD has slightly more parameters than the baseline denoiser mainly due to the duplicated timestep embedding MLP.

	Method	Decoder	Params (M) (+ decoder)	GFLOPs (final step)	FID↓	IS↑
Latent models	SiT-XL/2-REPA	SD-VAE	674.8 + 49.5	2 × 237.2 + 622.2	1.38	306.2
	DWD	–	678.5	237.7	1.36 (–0.02)	312.5 (+6.3)
	DWD-conv	–	678.9	239.2	1.35 (–0.03)	313.6 (+7.4)
	SiT-XL/2-REPA-E	E2E-VA-VAE	674.9 + 41.4	2 × 237.3 + 252.2	1.12	302.9
	DWD	–	678.5	237.7	1.06 (–0.06)	303.1 (+0.2)
	DWD-conv	–	679.0	239.2	1.10 (–0.02)	305.7 (+2.8)
Pixel models	JiT-H/16	–	952.5	2 × 363.9	1.90	300.9
	DWD	–	956.7	363.9	1.85 (–0.05)	305.7 (+4.8)
	DWD-conv	–	957.2	365.4	1.87 (–0.03)	305.3 (+4.4)
	w/o separate head	–	955.8	363.9	1.91 (+0.01)	302.4 (+1.5)
	AnisoFlow-H/16	–	953.1	2 × 364.8	1.55	300.1
	DWD	–	957.3	363.9	1.43 (–0.12)	303.5 (+3.4)
	DWD-conv	–	957.8	365.4	1.44 (–0.11)	303.6 (+3.5)
	w/o separate head	–	956.4	364.8	1.71 (+0.16)	302.9 (+2.8)

3.4 Extension to pixel diffusion and few-step models

Pixel diffusion models. The quality benefit of the dual-head separation is most pronounced on pixel diffusion models, which have no decoder to confound the comparison. We still add a DWD head in parallel with the denoising head, treating pixel denoising the same as the latent case. The training objectives are unchanged except for a stronger anchoring weight. Fewer tunable blocks and a shorter training schedule suffice, since the DiT already has pixel knowledge. Details are in Appendix A.

Few-step latent models. Few-step distilled models (Yin et al., 2024b;a; Sauer et al., 2024; Chen et al., 2026c) are the practical frontier of fast generation, where the fixed decoding cost takes a proportionally larger share of inference time. Validating DWD here therefore has direct practical significance. A simple adaptation is to adjust p_{pix} and p_{traj} to cover new ranges around the timesteps used by the few-step sampler. More variants are discussed in Appendix B.2 and B.5.

4 Experiments

4.1 ImageNet generation

We show that DWD consistently improves FID across both latent and pixel baselines. We then ablate the separate head design and show that fusing the two heads degrades quality despite the same training losses, and also study the effect of head capacity and final timestep.

Setup. We test two latent models, SiT-XL/2-REPA (Ma et al., 2024; Yu et al., 2025) with the SD-VAE decoder (Rombach et al., 2022) and SiT-XL/2-REPA-E (Leng et al., 2025) with its end-to-end tuned E2E-VA-VAE decoder (Yao et al., 2025), and two pixel models, JiT-H/16 (Li & He, 2026) and AnisoFlow-H/16 (Zirvi et al., 2026). DWD extensions are finetuned on ImageNet (Deng et al., 2009), with training schedules detailed in Appendix A.2. We report FID (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016) under the ADM evaluation protocol (Dhariwal & Nichol, 2021).

Improvement over latent diffusion baselines. Removing the decoder improves both latent baselines (Table 1): FID improves from 1.38 to 1.36 for REPA and from 1.12 to 1.06 for REPA-E, with IS improving as well. The two baselines use different latent spaces and decoders, yet a linear head improves on both. Removing the decoder also saves 622 decoder GFLOPs for REPA, 2.6× the cost of a single denoiser evaluation, and 252 decoder GFLOPs for REPA-E.

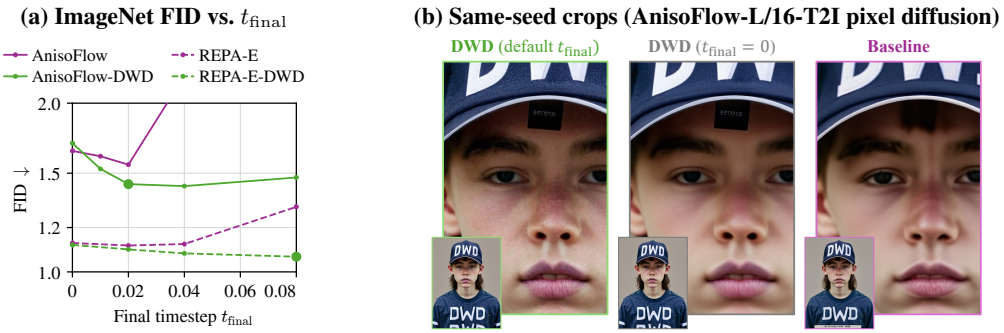


Figure 4: **A nonzero final timestep is necessary for the DWD quality gain. (a)** At t_{final}=0, where the DWD head receives a fully denoised input, FID matches or trails the baseline's, so the gain vanishes; a small positive timestep recovers it. Larger circles mark the defaults used in Table 1. **(b)** Qualitative results also confirm the trend: DWD at its default timestep recovers skin and fabric texture, while DWD at t_{final}=0 is as smooth as the baseline. The large panels are zoomed crops, with the full image inset.

Improvement over pixel diffusion baselines. The pixel models have no decoder to remove, so DWD only adds a second output trained with pixel quality losses. As shown in Table 1, DWD improves JiT from 1.90 to 1.85 FID and AnisoFlow from 1.55 to 1.43, a new state of the art among pixel flow models, while raising IS by 3–5 points. This also narrows the FID gap between pixel and latent models on ImageNet.

Separate head vs. shared head. The pixel-model gains have two sources: the pixel quality losses, which latent decoders are trained with but pixel models normally lack, and the separate head that receives them. The *w/o separate head* rows in Table 1 isolate the latter by fusing the two heads in both training and inference, so the denoising head receives both pixel losses and trajectory anchoring at their respective timesteps, with everything else unchanged. This degrades baseline FID (1.55 to 1.71 for AnisoFlow), so the losses alone do not account for the gain. The degradation reflects a conflict: the ℓ_2 denoising loss favors mode coverage, whereas LPIPS and adversarial losses favor perceptual quality but collapse diversity (better IS but worse FID than the baseline). Trajectory anchoring cannot fully counteract this interference, so the pixel losses need a separate head.

Linear vs. convolutional head. We ablate the capacity of the linear DWD head by adding a small convolutional residual refiner conditioned on the linear prediction and the last-layer tokens (*DWD-conv*; Appendix A). This extension brings no consistent improvement across the ImageNet models in Table 1. A linear head therefore suffices to obtain the quality gains under our tested settings.

Nonzero final timestep. Fig. 4 sweeps the final timestep at which the DWD head is evaluated. Quality degrades toward t=0, where the input is the fully denoised state a decoder would receive, and recovers at a small positive timestep. This shows that the DWD head needs a partially denoised input to produce high-quality pixels, rather than mimicking a decoder applied to a clean state.

4.2 Text-to-image generation

We show that DWD replaces the decoder with matched quality on latent baselines, including a few-step distilled model, and recovers photorealistic texture detail on the pixel baseline. We also ablate the adversarial loss to show that it is not critical on text-to-image generation.

Setup. We test two latent models built on FLUX.2 klein 9B (Black Forest Labs, 2025), the 32-step Base model and its distilled 4-step variant, and one pixel model, AnisoFlow-L/16-T2I (Zirvi et al., 2026) (0.9B). All DWD extensions are finetuned on public real-image datasets (Wang et al., 2026a; Desai et al., 2021; Schuhmann et al., 2022) at mixed aspect ratios. We evaluate at 1024×1024 on COCO-10k (Lin et al., 2014; 2024) (FID, patch FID (pFID) (Lin et al., 2024), HPSv3 (Ma et al., 2025), HPSv2.1 (Wu et al., 2023)), OneIG-Bench (Chang et al., 2025), and LongText-Bench (Wu et al., 2025). Training and sampling details are in Appendix A.

Table 2: **Results of text-to-image generation (1024×1024).** Shaded rows are our DWD extensions, finetuned from the baseline above them.

	Method	Params (B) (+ decoder)	TFLOPs (final step)	COCO-10k FID↓	COCO-10k pFID↓	COCO-10k HPSv3↑	COCO-10k HPS v2.1↑	OneIG↑	LongText↑
Latent	FLUX.2 klein Base 9B	9.08 + 0.05	2×75.51 + 10.47	15.13	18.69	10.54	**0.284**	**0.556**	**0.909**
	DWD	9.15	75.54	**15.05**	**18.38**	10.54	**0.284**	**0.556**	0.908
	w/o adversarial loss	9.15	75.54	15.10	18.97	**10.57**	**0.284**	0.555	0.907
	FLUX.2 klein 9B (4-step)	9.08 + 0.05	75.51 + 10.47	**21.50**	25.52	**12.23**	**0.307**	0.535	0.867
	DWD	9.15	75.54	21.92	**25.33**	12.15	**0.307**	**0.537**	**0.871**
Pixel	AnisoFlow-L/16-T2I	0.92	2×5.99	28.86	36.59	**10.72**	0.297	0.528	0.850
	DWD	0.94	5.98	**27.54**	**32.26**	10.63	**0.301**	**0.533**	**0.852**
	w/o adversarial loss	0.94	5.98	28.93	34.52	10.67	0.299	0.532	0.843

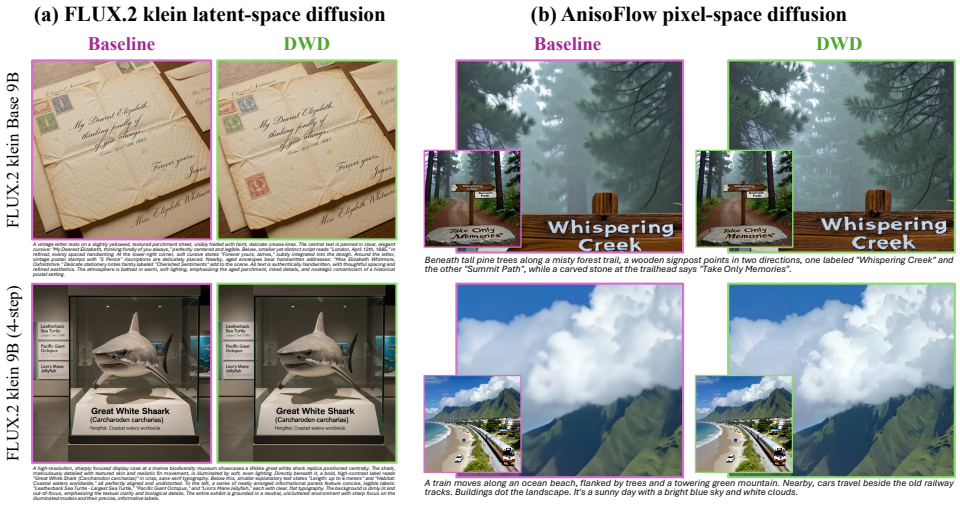


Figure 5: **Text-to-image comparisons.** Each pair shows the baseline (left) and its DWD extension (right) for the same prompt and seed. **(a)** On the latent FLUX.2 klein models, DWD closely matches the baseline's content and texture without the decoder (zoom in to view detail). **(b)** On the pixel AnisoFlow model, DWD preserves the composition while recovering photorealistic texture in bark, foliage, and clouds. The large panels are zoomed crops, with the full image inset.

Decoder removal without quality loss. On FLUX.2 klein Base, DWD and the baseline are effectively tied on every metric in Table 2, and same-seed samples are nearly indistinguishable (Fig. 5(a)). This is expected: the FLUX.2 latent is much wider than those of other VAEs, with 128 latent dimensions per token for a 768-dimensional pixel patch, compared with 64 for a 3072-dimensional patch in Wan, so decoding from it is easy, leaving little room for improvement. The result nevertheless shows that a linear head can replace it, removing 10.47 TFLOPs of decoder computation per image.

Compatibility with few-step model. The distilled 4-step FLUX.2 klein tests the few-step regime, where the decoder takes a larger share of the total inference cost. With only the timestep ranges adjusted (Appendix A.2), DWD is again effectively tied with the baseline in Table 2 and visually indistinguishable in Fig. 5(a), even though its final step operates at a much larger timestep (t=0.7 after shifting) than in the 32-step model.

Improved detail over pixel diffusion baseline. Pixel diffusion models tend to fall short of latent models in very fine texture, even with very large denoisers (Cai et al., 2026) or AnisoFlow's improved parameterization (Zirvi et al., 2026). With a linear head and 3k finetuning iterations, DWD recovers photorealistic skin, hair, foliage, and cloud detail while preserving composition (Fig. 1(b), Fig. 4(b), Fig. 5(b)). The gain also shows up in the low-level metrics in Table 2: COCO FID improves from 28.86 to 27.54 and pFID from 36.59 to 32.26, the latter showing a more significant improvement due to its sensitivity to fine detail. The remaining higher-level metrics are effectively tied, as the trajectory is preserved.

Table 3: **Results of text-to-video generation (832×480, 81 frames) on VBench.** Shaded rows are our DWD extensions, finetuned from the baseline above them. *Finetuned decoder* keeps the baseline DiT and finetunes its decoder on the same data with the same pixel losses as DWD (Appendix B.1). The 8-bit baseline and its DWD extension both use FP8 W8A8 quantization with 8-bit SageAttention2 (Zhang et al., 2025).

Method	Params (B) (+decoder)	PFLOPs (final step)	Warping error ($\times 10^{-3}$)↓	Image metrics		VBench dimensions						
				HPSv3↑	HPS v2.1↑	Subject consist.↑	Backgr. consist.↑	Temporal flicker.↑	Motion smooth.↑	Dynamic degree↑	Aesthetic quality↑	Imaging quality↑
Wan2.1 14B	14.29 + 0.07	2 × 1.68 + 0.28	1.52	**9.06**	**0.266**	0.951	0.981	0.990	0.983	**0.736**	0.682	**0.677**
DWD	14.49	1.68	**1.18**	9.04	**0.266**	**0.952**	**0.982**	**0.993**	**0.984**	0.711	**0.685**	0.671
finetuned decoder	14.29 + 0.07	2 × 1.68 + 0.28	1.39	9.03	**0.266**	0.951	**0.982**	0.992	0.983	0.731	0.681	0.676
Wan2.1 14B (8-bit)	14.29 + 0.07	2 × 1.68 + 0.28	1.69	8.99	0.265	0.950	0.979	0.990	0.982	**0.728**	0.675	**0.676**
DWD	14.49	1.68	**1.20**	**9.01**	**0.267**	**0.951**	**0.982**	**0.993**	**0.984**	0.703	**0.683**	0.668

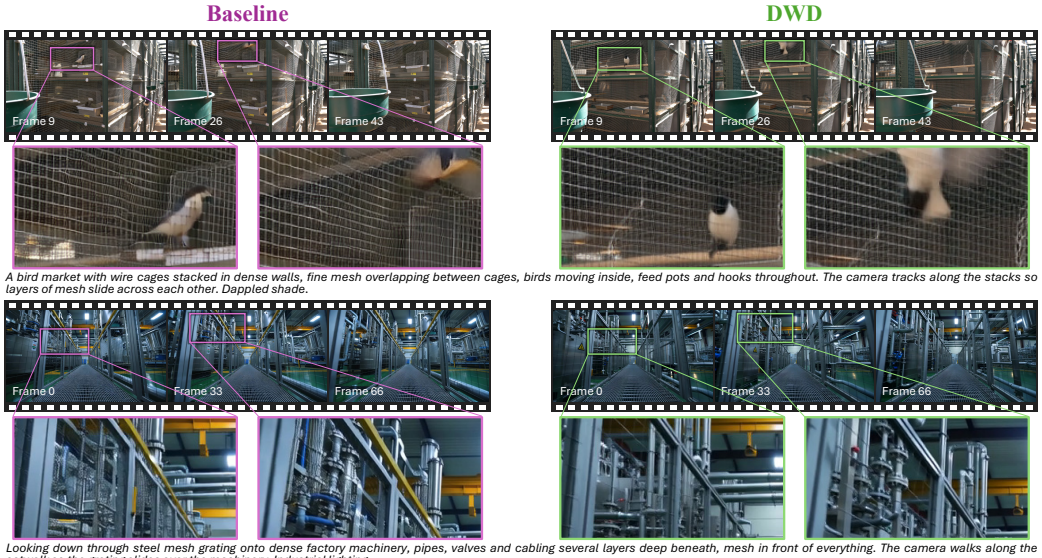


Figure 6: **Text-to-video comparisons (BF16).** Three frames from the baseline Wan2.1 14B (left) and its DWD extension (right) for the same prompt and seed, with two zoomed crops per strip. The baseline decoder warps the wire mesh into wavy, uneven lines (top) and breaks the dense pipes into speckle (bottom); neither defect is static, both shifting from frame to frame. DWD keeps the mesh regular and the pipe structure intact, and holds them steady as the camera tracks across the scene.

Adversarial loss. Without the adversarial loss (Table 2), FLUX.2 klein and AnisoFlow's DWD extensions stay close to full DWD on most metrics, except that pFID worsens more noticeably, though still close to or outperforming the baseline. The adversarial loss is thus helpful but not critical, in line with recent decoder research that drops it (Wu et al., 2025; Zhang et al., 2026).

4.3 Text-to-video generation

We apply DWD to the Wan2.1 14B video model, where each token must predict 4 frames rather than a single image patch. Despite this harder setting, DWD not only removes the decoder but visibly reduces distortion and flickering, lowering warping error by at least 23%.

Setup. We finetune Wan2.1 T2V 14B (Wang et al., 2025) on Vchitect T2V DataVerse (Fan et al., 2025) clips at mixed 360p, 480p, and 720p resolutions, all downsampled from sources of at least 720p to obtain sharper real pixels (Appendix A.2). We evaluate 81-frame 832×480 generations on extended VBench prompts (Huang et al., 2024) from Huang et al. (2025) and report warping error (Lai et al., 2018), frame-level HPSv3 and HPSv2.1, and seven VBench quality dimensions. For both the baseline and DWD, we additionally evaluate a quantized model with FP8 weights and activations (W8A8) and 8-bit SageAttention2 kernels (Zhang et al., 2025), which better reflects efficient deployment settings.

Decoder removal with reduced distortion and flickering. DWD removes the 0.28-PFLOP Wan decoder and replaces it with a linear head at negligible cost (Table 3), while also improving visual quality. The baseline decoder distorts fine detail such as small text and dense structures, and the distortion changes from frame to frame, which appears as flicker (Fig. 1(a) and Fig. 6). With the decoder removed, the linear DWD head keeps these details stable and legible, noticeably reducing flickering (see the project page for video comparisons). Warping error captures this artifact directly: it aligns consecutive frames by optical flow, masks occlusions, and measures the remaining photometric error, which DWD reduces by 23% (1.52 to 1.18) in Table 3. VBench's temporal flickering score is less suitable here, since it is the raw pixel difference between consecutive frames without accounting for residual motion, and it saturates near its maximum for both models. The remaining VBench dimensions and HPS metrics are effectively tied, except for a slightly lower dynamic degree, though qualitative results show no visible motion reduction. A decoder finetuned on the same data with the same pixel losses reduces warping error only to 1.39 (Table 3 and Appendix B.1), so most of the gain comes from DWD rather than from the training data.

8-bit inference. Quantization degrades the baseline noticeably (warping error 1.52 to 1.69) but barely affects DWD (1.18 to 1.20), widening the warping error gap to 29% in Table 3. The other metrics follow the same pattern as under BF16. The DWD extension is thus robust to reduced-precision inference, and the quality gain over the decoder is larger in the deployment setting.

5 Conclusion

DWD replaces the decoder with a linear pixel head on the diffusion transformer, finetuning only the last few blocks and output heads. Across latent image and video generators, DWD preserves or improves pixel quality while removing the decoder entirely. On video, this visibly reduces detail distortion and flickering. On pixel diffusion models, a separate DWD head recovers photorealistic texture that the denoising head oversmooths. The broader lesson is that final pixel prediction should be separated from intermediate denoising, so that pixel-space objectives can improve quality without interfering with the sampling trajectory.

Limitations. The DWD quality gain shrinks when the decoder is already strong, as on FLUX.2. We have not proved that the linear head is sufficient for spatial patch sizes beyond 16×16 pixels, where a stronger head like DWD-conv may be needed. Also, we have not tuned the pixel losses per model, leaving potential room for improvement.

AI Use Statement

In this work, we used generative AI tools (LLM-based assistants) to edit the paper for grammar and readability and to accelerate the implementation of our experiments. All AI-assisted work was reviewed and verified by the authors, and we take responsibility for the final content of this paper.

Reproducibility Statement

Appendix A gives the implementation details: the head and its initialization (Appendix A.1), the per-model finetuning settings and training data (Appendix A.2 and Table 4), the inference procedure (Appendix A.3), the video evaluation protocol (Appendix A.4), and the parameter and FLOP accounting (Appendix A.5).

References

Yagmur Akarken, Orest Kupyn, and Christian Rupprecht. MMDiff: Extending diffusion transformers for multi-modal generation. In *ECCV*, 2026.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023.

Mustafa Ayazoglu. Extremely lightweight quantization robust real-time single-image super resolution for mobile devices. In *CVPR Workshops*, 2021.

Tariq Berrada Ifriqi, Pietro Astolfi, Melissa Hall, Marton Havasi, Yohann Benchetrit, Adriana Romero-Soriano, Karteek Alahari, Michal Drozdzal, and Jakob Verbeek. Boosting latent diffusion with perceptual objectives. In *ICLR*, 2025. URL https://openreview.net/forum?id=y4DtzADzd1.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Black Forest Labs. Flux.2: Frontier visual intelligence. https://bfl.ai/blog/flux-2, 2025.

Qi Cai, Jingwen Chen, Chengmin Gao, Zijian Gong, Yehao Li, Yingwei Pan, Yi Peng, Zhaofan Qiu, Kai Yu, Yiheng Zhang, Hao Ai, Siying Bai, Yang Chen, Zhihui Chen, Fengbin Gao, Ying Guo, Dong Li, Zhen Shen, Leilei Shi, Jing Wang, Siyu Wang, Yimeng Wang, Rui Zheng, Ting Yao, and Tao Mei. HiDream-O1-Image: A natively unified image generative foundation model with pixel-level unified transformer. *arXiv preprint arXiv:2605.11061*, 2026. URL https://arxiv.org/abs/2605.11061.

Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. In *NeurIPS Datasets and Benchmarks*, 2025.

Hansheng Chen, Jan Ackermann, Minseo Kim, Gordon Wetzstein, and Leonidas Guibas. Asymmetric flow models. In *NeurIPS*, 2026a.

Hansheng Chen, Shengqu Cai, Jiaming Song, Gordon Wetzstein, Xinchen Yan, and Leonidas Guibas. pi-H3: Video Generation in 2–4 Steps Without a VAE Decoder. https://hanshengchen.com/blog/pi-h3/, November 2026b.

Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. π-flow: Policy-based few-step generation via imitation distillation. In *ICLR*, 2026c.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f062da1973ac9ac61fc6d44dd7fa309f-Abstract-Conference.html.

Zhennan Chen, Junwei Zhu, Xu Chen, Jiangning Zhang, Jiawei Chen, Zhuoqi Zeng, Wei Zhang, Chengjie Wang, Jian Yang, and Ying Tai. L2P: Unlocking latent potential for pixel generation. *arXiv preprint arXiv:2605.12013*, 2026d. URL https://arxiv.org/abs/2605.12013.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *ICML*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*, 2021.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *NeurIPS*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Weichen Fan, Chenyang Si, Junhao Song, Zhenyu Yang, Yinan He, Long Zhuo, Ziqi Huang, Ziyue Dong, Jingwen He, Dongwei Pan, Yi Wang, Yuming Jiang, Yaohui Wang, Peng Gao, Xinyuan Chen, Hengjie Li, Dahua Lin, Yu Qiao, and Ziwei Liu. Vchitect-2.0: Parallel transformer for scaling up video diffusion models. *arXiv preprint arXiv:2501.08453*, 2025.

Chaofan Gan, Yuanpeng Tu, Xi Chen, Tieyuan Chen, Yuxi Li, Mehrtash Harandi, and Weiyao Lin. Unleashing diffusion transformers for visual correspondence by modulating massive activations. In *NeurIPS*, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/a654298de44ec95dec28aa37e8d0af51-Abstract-Conference.html.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html.

Shiqiao Gu, Yang Yong, Fengjun Zhong, Ruihao Gong, Chengtao Lv, Xiangyu Fan, Lei Yang, and Xianglong Liu. LightVAE: Towards compact and efficient video autoencoders. *arXiv preprint arXiv:2610.14726*, 2026.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024. URL https://arxiv.org/abs/2501.00103.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, pp. 13213–13232, 2023.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *NeurIPS*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

Minguk Kang and Suha Kwak. FlashDecoder: Real-time latent-to-pixel streaming decoder with transformers. In *CVPR*, 2026.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Daquan Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2025. URL https://arxiv.org/abs/2412.03603.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. In *ICCV*, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *CVPR*, 2026.

Yitong Li, Jincheng Yu, Junsong Chen, Haopeng Li, Shuchen Xue, Haozhe Liu, Ping Luo, Song Han, and Enze Xie. Sol-H3: Recursive self-improvement for MiniMax-H3 inference acceleration on Sol-Engine across cloud and edge. *arXiv preprint arXiv:2609.35110*, 2026.

Shanchuan Lin and Xiao Yang. Diffusion model with perceptual loss. *arXiv preprint arXiv:2401.00110*, 2023. URL https://arxiv.org/abs/2401.00110.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. URL https://arxiv.org/abs/2402.13929.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755. Springer International Publishing, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Yifan Lu, Qi Wu, Jay Zhangjie Wu, Zian Wang, Huan Ling, Sanja Fidler, and Xuanchi Ren. PiD: Fast and high-resolution latent decoding with pixel diffusion. *arXiv preprint arXiv:2605.23902*, 2026. URL https://arxiv.org/abs/2605.23902.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *ICCV*, 2025.

Zehong Ma, Longhui Wei, Shuai Wang, Shiliang Zhang, and Qi Tian. DeCo: Frequency-decoupled pixel diffusion for end-to-end image generation. In *CVPR*, 2026a.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. *arXiv preprint arXiv:2602.02493*, 2026b.

Björn Ottosson. A perceptual color space for image processing, 2020. URL https://bottosson.github.io/posts/oklab/.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. StyleGAN-T: Unlocking the power of GANs for fast large-scale text-to-image synthesis. In *ICML*, 2023.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia*, 2024.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks*, 2022. URL https://openreview.net/forum?id=M3Y74vmsMcY.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026. URL https://arxiv.org/abs/2605.18324.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. In *ICML*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

Théophane Vallaeys, Jakob Verbeek, and Matthieu Cord. SSDD: Single-step diffusion decoder for efficient image tokenization. *arXiv preprint arXiv:2510.04961*, 2025. URL https://arxiv.org/abs/2510.04961.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

Alex Jinpeng Wang, Dongxing Mao, Jiawei Zhang, Weiming Han, Zhuobai Dong, Linjie Li, Yiqi Lin, Zhengyuan Yang, Libo Qin, Fuwei Zhang, Lijuan Wang, and Min Li. Textatlas5m: A large-scale dataset for dense text image generation. In *ICML*, 2026a.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. In *ICLR*, 2026b. URL https://openreview.net/forum?id=BDnOrExHmt.

Zanyi Wang, Xin Lin, Haodong Li, Dengyang Jiang, and Yijiang Li. From RGB generation to dense field readout: Pixel-space dense prediction with text-to-image models. *arXiv preprint arXiv:2607.06553*, 2026c. URL https://arxiv.org/abs/2607.06553.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. URL https://arxiv.org/abs/2508.02324.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. URL https://arxiv.org/abs/2306.09341.

Yiran Xu, Taesung Park, Richard Zhang, Yang Zhou, Eli Shechtman, Feng Liu, Jia-Bin Huang, and Difan Liu. VideoGigaGAN: Towards detail-rich video super-resolution. In *CVPR*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024a. URL https://openreview.net/forum?id=tQukGCDaNT.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024b.

Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *CVPR*, 2026.

Z-Image Team, Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, Zhen Li, Zhong-Yu Li, David Liu, Dongyang Liu, Junhan Shi, Qilong Wu, Feng Yu, Chi Zhang, Shifeng Zhang, and Shilin Zhou. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025. URL https://arxiv.org/abs/2511.22699.

Boya Zeng, Tianze Luo, Shu Pu, Jucheng Shen, Taiming Lu, Gabriel Sarch, and Zhuang Liu. i1: A simple and fully open recipe for strong text-to-image models. *arXiv preprint arXiv:2606.11289*, 2026.

Christina Zhang, Simran Motwani, Matthew Yu, Ji Hou, Felix Juefei-Xu, Sam S. Tsai, Peter Vajda, Zijian He, and Jialiang Wang. Pixel-space post-training of latent diffusion models. *arXiv preprint arXiv:2409.17565*, 2024. URL https://arxiv.org/abs/2409.17565.

Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. SageAttention2: Efficient attention with thorough outlier smoothing and per-thread INT4 quantization. In *ICML*, 2025. URL https://proceedings.mlr.press/v267/zhang25ae.html.

Richard Zhang, Phillip Isola, Alexei Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zekai Zhang, Deqing Li, Kuan Cao, Yujia Wu, Chenfei Wu, Yu Wu, Liang Peng, Hao Meng, Jiahao Li, Jie Zhang, Kaiyuan Gao, Kun Yan, Lihan Jiang, Ningyuan Tang, Shengming Yin, Tianhe Wu, Xiao Xu, Xiaoyue Chen, Yan Shu, Yanran Zhang, Yilei Chen, Yixian Xu, Yuxiang Chen, Zhendong Wang, Zihao Liu, Zikai Zhou, Yiliang Gu, Yi Wang, Xiaoxiao Xu, and Lin Qu. Qwen-image-vae-2.0 technical report. *arXiv preprint arXiv:2605.13565*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026a. URL https://openreview.net/forum?id=0u1LigJaab.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. In *ICLR*, 2026b. URL https://openreview.net/forum?id=2uNlM353RI.

Lunjie Zhu, Yushi Huang, Xingtong Ge, Yufei Xue, Zhening Liu, Yumeng Zhang, Zehong Lin, and Jun Zhang. Flash-VAED: Plug-and-play VAE decoders for efficient video generation. In *ICML*, 2026.

Rayhan Zirvi, Hansheng Chen, Gordon Wetzstein, and Leonidas Guibas. Learning anisotropic prediction targets for flow models. *arXiv preprint arXiv:2611.17767*, 2026.

A Implementation Details

This section complements Sec. 3 with the specifics of the DWD head, finetuning, inference, and evaluation, with per-model settings collected in Table 4.

A.1 DWD Head

Output format. The linear head maps each last-layer token to one pixel patch, and the patches are assembled into the image. Image models use 16×16 patches. Wan uses 4×16×16 spatiotemporal patches and a separate 1×16×16 head for the first latent frame, which covers a single pixel frame in the Wan VAE. Following AsymFlow (Chen et al., 2026a), latent models and AnisoFlow-L/16-T2I predict pixels in normalized Oklab color space (Ottosson, 2020), while the pixel quality losses are applied to the converted RGB values, matching the common autoencoder training recipe. The ImageNet pixel models predict RGB directly, matching their native output.

Least-squares initialization. The head is initialized by linear regression from last-layer tokens to ground-truth pixel patches. We run the frozen pretrained model on 10k training images with timesteps drawn from p_{pix}, the same distribution used for the pixel losses, and solve the normal equations for the weight and bias jointly. The same procedure produces the probes in Fig. 3, where the latent probe regresses from the denoised latent instead of the tokens.

Convolutional head (DWD-conv). The DWD-conv ablation keeps the linear head and adds a residual correction, following lightweight super-resolution networks (Shi et al., 2016; Yu et al., 2018; Ayazoglu, 2021): the nonlinear convolutions run at half resolution, a final sub-pixel convolution (pixel shuffle) restores the full resolution, and the branch predicts a residual over a linear base prediction. It packs each 2×2 neighborhood of the linear pixel prediction into channels and refines this packed grid with a convolutional branch. The last-layer tokens are projected through a 128-dimensional bottleneck and expanded into a 32-channel spatial feature map, which is added to a 1×1 projection of the packed pixels. Two residual blocks with 3×3 convolutions and SiLU activations predict the correction through a zero-initialized output convolution, so the head starts as the linear head. For video, the convolutions operate on each frame independently.

A.2 Finetuning

Table 4: **DWD finetuning settings per model.** K is the number of finetuned blocks out of the total. Adv. start is the iteration at which the generator adversarial loss is switched on, after a discriminator warmup. λ is the trajectory anchoring weight in Eq. (2), and p_{traj} and p_{pix} are the timestep ranges of the two losses before the timestep shift. Mixed-resolution runs use the batch sizes given in Appendix A.2, and FLUX.2 klein Base uses the model's resolution-dependent shift (dyn.). GPU hours are estimated for H100 GPUs.

Model	K	Iters	Batch	Adv. start	λ	p_{traj}	p_{pix}	Shift	Final t	Pixel space	GPU hours
SiT-XL/2-REPA	16 / 28	100k	1024	52.5k	5	$[0.04, 1]$	$[0, 0.1]$	1	0.08	Oklab	170
SiT-XL/2-REPA-E	8 / 28	200k	1024	152.5k	10	$[0.04, 1]$	$[0, 0.1]$	1	0.08	Oklab	200
JiT-H/16	2 / 32	4k	1024	2.5k	50	$[0.04, 1]$	$[0, 0.1]$	1	sampler	RGB	8
AnisoFlow-H/16	2 / 32	4k	1024	2.5k	50	$[0.04, 1]$	$[0, 0.1]$	1	sampler	RGB	8
FLUX.2 klein Base 9B	4 / 32	26k	mixed	21k	10	$[0.04, 1]$	$[0, 0.1]$	dyn.	0.3	Oklab	350
FLUX.2 klein 9B (4-step)	4 / 32	26k	128	21k	10	$[0.3, 1]$	$[0, 0.5]$	7	sampler	Oklab	340
AnisoFlow-L/16-T2I	4 / 23	3k	128	2k	500	$[0.04, 1]$	$[0, 0.1]$	5	sampler	Oklab	14
Wan2.1 T2V 14B	8 / 40	40k	mixed	21k	5	$[0.04, 1]$	$[0.02, 0.1]$	7	0.3	Oklab	4300
Wan2.1 T2V 1.3B (4-step)	8 / 30	40k	mixed	35k	5	$[0.3, 0.94]$	$[0.02, 0.3]$	5.5	sampler	Oklab	980

Trainable components. Table 4 lists the number of finetuned blocks. The selected blocks are trained in full, except for the SiT models, where we freeze the query and key projections because SiT has no QK normalization and finetuning them can be unstable. All models also train both output heads, the final modulation, and the duplicated timestep embedding MLP (together with the class, guidance, or text modulation MLPs where present), all initialized from the pretrained weights.

Pixel losses. The ℓ_1 and LPIPS terms are weighted 1:1 in all runs. The discriminator follows the StyleGAN-T design (Sauer et al., 2023) as adopted by RAE (Zheng et al., 2026a). For the text-to-image and video models, it scores multiple random 256×256 crops of each image or frame rather than the full resolution. For video, the discriminator features are extracted per frame, and we insert a

residual block with a 1D convolution over the frame axis (kernel size 3) before the per-token scoring. The adversarial weight is set adaptively to match the gradient norm of the ℓ_1 and LPIPS loss at the predicted pixels, following the gradient balancing of Esser et al. (2021) but measured at the pixels rather than at the last layer's weights. The discriminator is trained for a short warmup before the generator adversarial loss is switched on at the iteration listed in Table 4 (2.5k iterations of warmup for the ImageNet models, 1k for the others).

Optimization. All runs use Adam (Kingma & Ba, 2015) with learning rate 10^{-4}, betas $(0.9, 0.95)$, no weight decay, a short linear warmup, and a 5× learning rate for the DWD head. Weights are evaluated with the exponential moving average of Karras et al. (2024) ($\gamma = 7$), and Wan additionally caps the EMA momentum at 0.998. For FLUX.2 klein and Wan, the frozen blocks are quantized to FP8 during finetuning to reduce memory and speed up training. The frozen reference model is also quantized to FP8, while the trainable layers are trained in FP32 with BF16 mixed precision. The trainable layers therefore learn from a slightly degraded FP8 reference, which works against DWD rather than favoring its 8-bit results. All main and qualitative results use BF16 inference.

Training data. The ImageNet models are finetuned on the ImageNet training set at 256×256. The text-to-image models are finetuned on the one-megapixel training data mix of AnisoFlow (Zirvi et al., 2026), consisting of TextAtlas (Wang et al., 2026a), RedCaps (Desai et al., 2021), and LAION-Aesthetics (Schuhmann et al., 2022) images, with captions from i1 (Zeng et al., 2026) and Asym-Flow (Chen et al., 2026a). FLUX.2 klein Base additionally trains on the same images downsampled to roughly 0.5 and 0.25 megapixels. The three tiers use per-iteration batch sizes of 128, 256, and 384 images and are sampled in the ratio 1:2:3. The video model is finetuned on clips from Vchitect T2V DataVerse (Fan et al., 2025) with the dataset's captions. Clips are bucketed by resolution and length: 352×640 with 81 or 33 frames (per-iteration batch sizes of 64 and 192 clips), 480×832 with 81, 49, or 25 frames (32, 64, and 128 clips), and 720×1280 with 33 or 17 frames (32 and 64 clips). Only source videos of at least 720p are used, so every training clip is downsampled from a higher or equal source resolution rather than upscaled from low-bitrate video, which we found essential for a sharp ground truth.

A.3 Inference

Guidance and final step. Sampling follows each baseline's default sampler and guidance, with one exception: the SiT-XL/2-REPA DWD extension uses guidance scale 2.3 on the interval $[0, 0.56]$. The final DWD step is evaluated once with the conditional input only, without a classifier-free guidance pair. Since latent models favor a larger final timestep (Fig. 4(a)), their final timestep is set manually as in Table 4, whereas the pixel models and the 4-step models use the last timestep of the baseline sampler. For Wan2.1 14B, the final timestep $t = 0.3$ after a shift of 7 corresponds to $t \approx 0.058$ before shifting. Its pixel loss skips timesteps below 0.02 before shifting, where Wan's own predictions show strong color drift and detail degradation (common sampling grids do not use these timesteps either).

8-bit inference. The 8-bit setting quantizes all linear layers to FP8 weights and activations (W8A8) and replaces attention with the 8-bit SageAttention2 kernels (Zhang et al., 2025), for both the baseline and DWD.

A.4 Video Evaluation

Warping error (Lai et al., 2018) is computed for every pair of adjacent frames at native resolution: we estimate forward and backward optical flow with RAFT (Teed & Deng, 2020), warp the second frame onto the first, mask occlusions and motion boundaries by forward-backward consistency as in the released scorer of Lai et al. (2018), and average the squared RGB error over valid pixels following Xu et al. (2025). Errors are averaged over pairs within each video, then over videos. HPSv3 and HPSv2.1 are computed on five frames sampled uniformly from each video (one frame per second at 16 fps) and averaged.

A.5 Cost Accounting

The parameter counts and final-step FLOPs in Tables 1–3 are computed as follows. Parameter counts include the denoiser and, for baselines, the decoder, excluding text encoders. For DWD extensions, the count includes the duplicated timestep embedding MLP and the linear head, so some DWD

models have slightly more parameters than their baselines, though the MLP contributes negligible compute. Final-step FLOPs count one denoiser evaluation, plus the decoder for the baselines, at each model's evaluated resolution. The greyed factor of two marks the baselines' classifier-free guidance pair, which DWD skips in the final step.

B Additional Experiments

B.1 Decoder-Retaining Baseline

To separate decoder removal from the effect of the training data and pixel losses, we finetune the native Wan2.1 decoder on the same video data with the same ℓ_1, LPIPS, and adversarial losses as DWD (Appendix A.2), using the same optimizer and EMA settings. The pretrained decoder is already well trained and its training loss stays flat, so we evaluate it after 2k iterations. In Table 3, the finetuned decoder reduces warping error only from 1.52 to 1.39, while DWD reduces it to 1.18, and the other metrics are effectively tied. Most of the DWD gain therefore comes from the method rather than from the training data. Finetuning the decoder is also not cheap: one decoder training iteration takes 24 s, about 2× slower than a DWD iteration on Wan2.1 14B (12.5 s).

B.2 Few-Step Video Generation

We apply DWD to the 4-step Wan2.1 T2V 1.3B model distilled with rCM (Zheng et al., 2026b), the example from Sec. 1. We finetune its last 8 of 30 blocks on the 352×640 and 480×832 tiers of the video data, and keep the sampler's exact timesteps (Table 4). The first 32.5k iterations use $p_{\mathrm{traj}} = [0.1, 0.94]$ and $p_{\mathrm{pix}} = [0.02, 0.1]$, after which we switch to the ranges in Table 4 to match the final timestep of the 4-step sampler. In Table 5, DWD reduces warping error by 27% and improves temporal flickering. Note that the 1.3B model is an unfavorable setting for DWD. It is much smaller than Wan2.1 14B, and tuning only 8 blocks leaves limited learnable capacity.

Self-distillation. With very few steps, the final step has to generate more content, so its pairing with real pixels becomes ambiguous, potentially making ℓ_1 and perceptual losses less effective. We therefore also test a self-distillation variant, which replaces the real pixels with a target produced by a chain: the reference model's native final step, renoising to a small timestep, and the pixel output of a previously trained small-time DWD model. The results are shown in Table 5. In this experiment, the small-time DWD model is from the standard run after its first 32.5k iterations. The self-distillation DWD resumes from this checkpoint and continues to 40k iterations. This variant has nearly the same dynamic degree as the baseline (0.822 vs. 0.825) and still reduces warping error by 14%.

Table 5: **Few-step video generation on 4-step Wan2.1 T2V 1.3B (BF16).** All models use the same sampler timesteps.

Method	Warping error ($\times 10^{-3}$)↓	Image metrics		VBench dimensions						
		HPSv3↑	HPS v2.1↑	Subject consist.↑	Backgr. consist.↑	Temporal flicker.↑	Motion smooth.↑	Dynamic degree↑	Aesthetic quality↑	Imaging quality↑
Wan2.1 1.3B (4-step)	2.84	8.32	**0.265**	0.938	**0.962**	0.976	0.979	**0.825**	**0.660**	**0.669**
DWD	**2.07**	**8.46**	0.258	**0.946**	0.959	**0.984**	**0.983**	0.811	0.657	0.648
DWD (self-distillation)	2.45	8.45	0.261	0.942	0.956	0.982	0.981	0.822	0.656	0.631

B.3 End-to-End Latency and Memory

Table 6 reports end-to-end latency and peak VRAM for the 4-step models, with FLUX.2 klein 9B at 1024×1024 and Wan2.1 1.3B at 81 frames of 480×832, both in eager mode (without `torch.compile`). DWD makes end-to-end generation 1.12× faster on FLUX.2 and 1.69× faster on Wan, with 5% and 16% lower peak VRAM, respectively.

Table 6: **End-to-end latency and peak VRAM of 4-step models.**

Model	Latency (s)↓	Peak VRAM (GB)↓
FLUX.2 klein 9B (4-step)	0.776	37.31
DWD	**0.695**	**35.54**
Wan2.1 1.3B (4-step)	4.661	19.22
DWD	**2.757**	**16.15**

B.4 Comparison with Latent-to-Pixel Adaptation

AsymFlow (Chen et al., 2026a) and L2P (Chen et al., 2026d) also remove the decoder, by adapting a pretrained latent model into a pixel model, which changes its sampling trajectory. We compare the official AsymFlow and L2P models against their respective base models, FLUX.2 klein Base 9B (Black Forest Labs, 2025) and Z-Image Turbo (Z-Image Team et al., 2025). Both methods are substantially worse than their base models on the reported metrics (Table 7), especially in text rendering (Fig. 7). For AsymFlow, this is partly due to its sensitivity to the training data, which contains little text rendering, and partly due to the difficulty of adapting a latent trajectory to a pixel trajectory. In contrast, DWD matches the base model while preserving seed-level content.

DWD is also much cheaper to train, requiring only about a third of AsymFlow's training cost (350 vs. 1100 H100 GPU hours) even with a longer, saturated schedule (26k vs. 15k iterations). This is because DWD finetunes only the last few blocks with shallow backpropagation.

Table 7: **Comparison with latent-to-pixel adaptation (1024×1024).** Z-Image Turbo results are from its official report (Z-Image Team et al., 2025).

Method	FID↓	pFID↓	OneIG↑	LongText↑	Training cost (H100 GPU hours)
FLUX.2 klein Base 9B	15.13	18.69	**0.556**	**0.909**	–
AsymFlow	18.10	22.62	0.506	0.777	1100 (15k iters)
DWD	**15.05**	**18.38**	**0.556**	0.908	350 (26k iters)
Z-Image Turbo	–	–	**0.528**	**0.917**	–
L2P	–	–	0.499	0.787	–

B.5 Preliminary Results on MiniMax-H3

We also apply DWD to 2-step and 4-step distilled MiniMax-H3 video models (Chen et al., 2026b). These models use a decoder distillation variant, where the final step's ℓ_1 and perceptual target becomes the output of the reference model's native final step followed by the native decoder.

Due to the extreme compression of the MiniMax-H3 autoencoder, each token covers a 32×32 spatial patch over a maximum of 4 frames, so a linear head would have up to 4×32×32×3 = 12288 output dimensions, more than twice the DiT's hidden dimension of 5376. In addition, MiniMax-H3 is a single-stream model, whose transformer weights are shared across text, visual, and audio tokens. Therefore, a linear head has very limited capacity on these models, which benefit from a lightweight refiner (e.g., the DWD-conv head in Appendix A.1).

Please refer to our blog post (Chen et al., 2026b) for video results. These results are purely preliminary, since we have not yet determined the optimal refiner architecture.

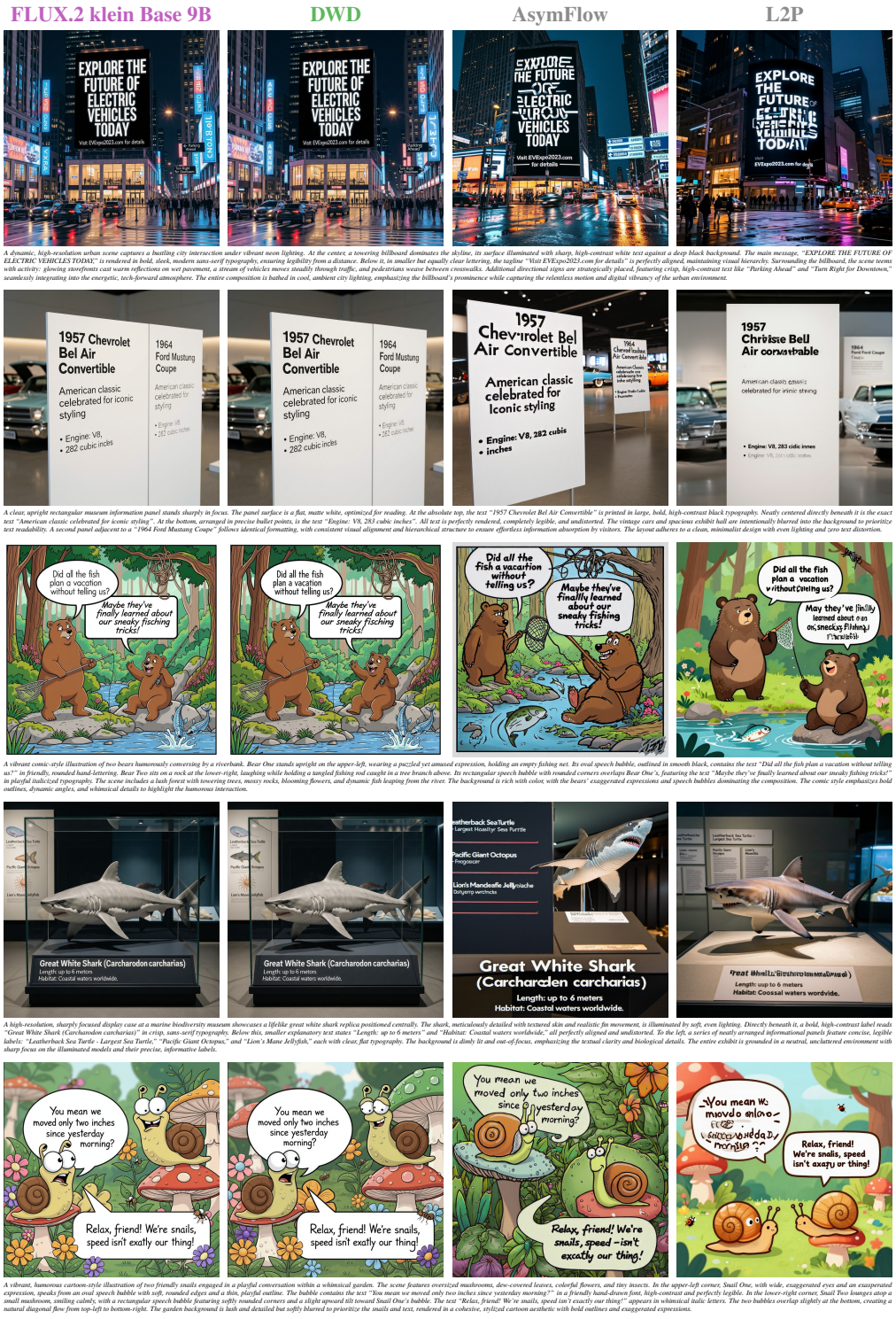


Figure 7: **Text rendering on LongText-Bench prompts (1024×1024).** Both DWD and AsymFlow are finetuned from FLUX.2 klein Base 9B, while L2P is finetuned from Z-Image Turbo. All images use the same seed. DWD preserves its base model's text rendering capability, whereas AsymFlow and L2P distort letters and words.

C Additional Qualitative Results

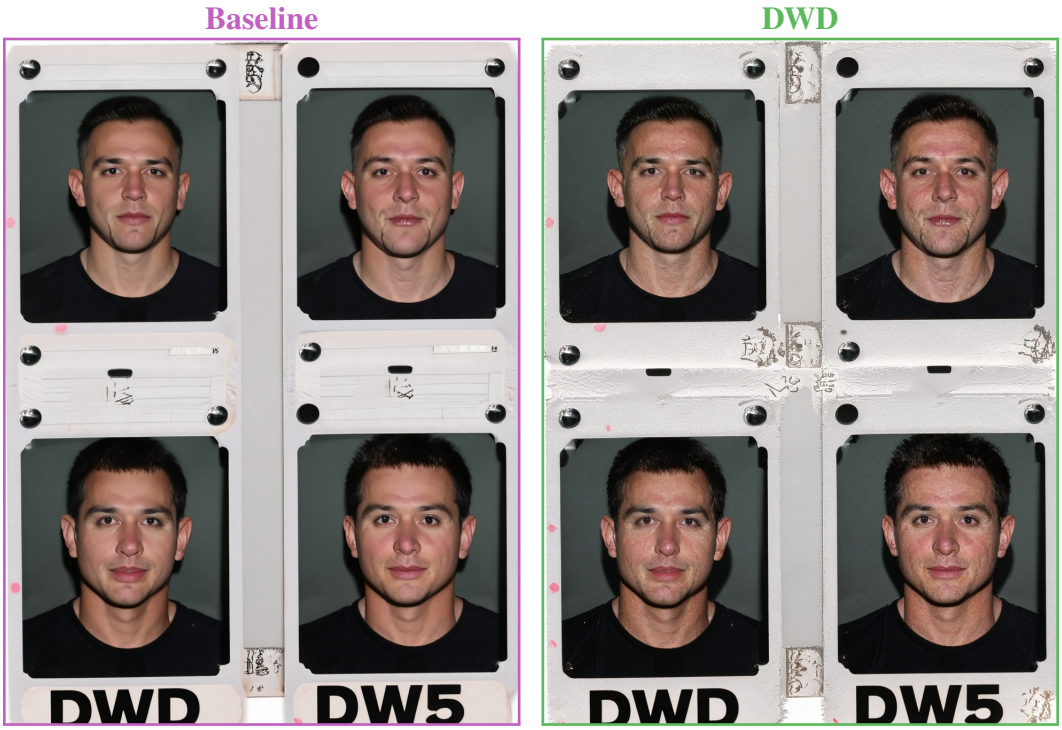


Avant-garde fashion portrait, a feathered headdress with a band reading "DWD", stark studio lighting, black background.

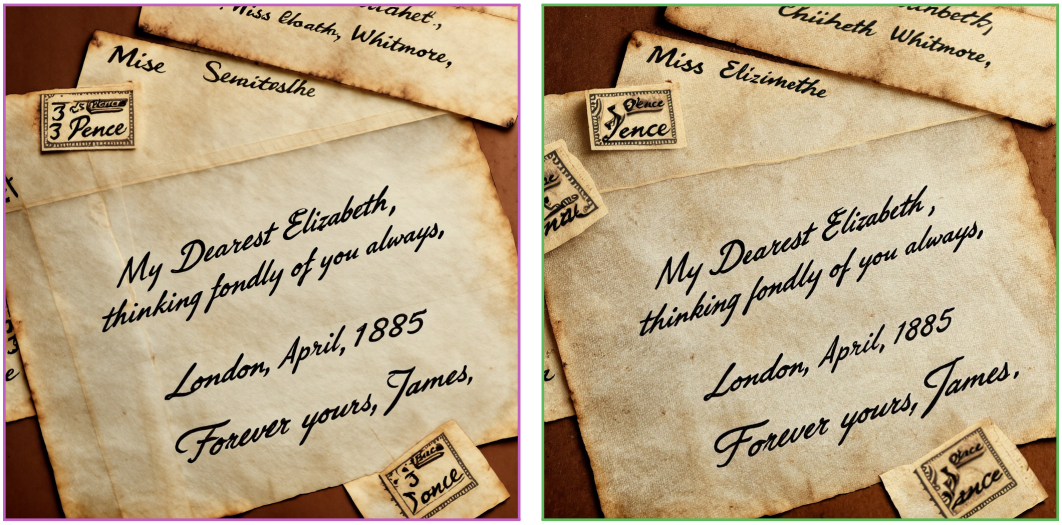


A fashion photo of a model in a coat with "DWD" on the collar.

Figure 8: **Additional same-seed comparisons on AnisoFlow-L/16-T2I (1/3).** Feather barbs and wool weave are recovered by DWD; the baseline blurs both into flat regions.



A 1995 photo booth strip of four frames, "DWD" stamped across the bottom edge.



A vintage letter rests on a slightly yellowed, textured parchment sheet, visibly folded with faint, delicate crease lines. The central text is penned in clear, elegant cursive: "My Dearest Elizabeth, thinking fondly of you always," perfectly centered and legible. Below, smaller yet distinct script reads "London, April 12th, 1885," in refined, evenly spaced handwriting. At the lower-right corner, soft cursive states "Forever yours, James," subtly integrated into the design. Around the letter, vintage postal stamps with "3 Pence" inscriptions are delicately placed. Nearby, aged envelopes bear handwritten addresses: "Miss Elizabeth Whitmore, Oxfordshire." Delicate stationery notes faintly labeled "Cherished Sentiments" add to the scene. All text is authentically handwritten, with thoughtful spacing and refined aesthetics. The atmosphere is bathed in warm, soft lighting, emphasizing the aged parchment, inked details, and nostalgic romanticism of a historical postal setting.

Figure 9: **Additional same-seed comparisons on AnisoFlow-L/16-T2I (2/3).** Skin texture in the photo booth frames and the small stamp and envelope lettering on the letter are sharper under DWD. The letter prompt is from LongText-Bench.



A man in his skiing gear is posing for the camera.



An older woman riding a train while sitting under it's window.



A cobblestone trail with rainbow chalk stripes leads to a pale blue gothic church featuring white-framed arc windows and a single tower, surrounded by dark green trees and a sunlit Nordic mountain.

Figure 10: **Additional same-seed comparisons on AnisoFlow-L/16-T2I (3/3).** COCO and OneIG prompts. DWD recovers facial detail, fabric texture, and the individual cobblestones that the baseline smooths.