Representation Alignment for Just Image Transformers is not Easier than You Think

Jaeyo Shin[†], Jiwook Kim[†], and Hyunjung Shim

[†] indicates equal contributions.

KAIST AI,
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
{jaeyo_shin,tom919,kateshim}@kaist.ac.kr

Abstract. Representation Alignment (REPA) has emerged as a simple way to accelerate Diffusion Transformers training in latent space. At the same time, pixel-space diffusion transformers such as Just image Transformers (JiT) have attracted growing attention because they remove a dependency on a pretrained tokenizer, and then avoid the reconstruction bottleneck of latent diffusion. This paper shows that the REPA can fail for JiT. REPA yields worse FID for JiT as training proceeds and collapses diversity on image subsets that are tightly clustered in the representation space of pretrained semantic encoder on ImageNet. We trace the failure to an information asymmetry: denoising occurs in the high dimensional image space, while the semantic target is strongly compressed, making direct regression a shortcut objective. We propose PixelREPA, which transforms the alignment target and constrains alignment with a Masked Transformer Adapter that combines a shallow transformer adapter with partial token masking. PixelREPA improves both training convergence and final quality. PixelREPA reduces FID from 3.66 to 3.17 for JiT-B/16 and improves Inception Score (IS) from 275.1 to 284.6 on ImageNet 256×256, while achieving $> 2\times$ faster convergence. Finally, PixelREPA-H/16 achieves FID= 1.81 and IS= 317.2.

Keywords: Pixel space diffusion model · Representation Alignment

1 Introduction

Diffusion models [14,35,38,39] can be categorized by the choice of data space in which denoising is performed. Latent Diffusion Models (LDMs) [35] reduce computation by mapping pixels into a learned latent space via a pretrained image tokenizer. However, this choice couples the achievable generation quality to the capacity and reconstruction fidelity of the tokenizer: strong compression attenuates fine textures and small structures, imposing an upper bound on what the

Our code is available at https://github.com/kaist-cvml/PixelREPA.

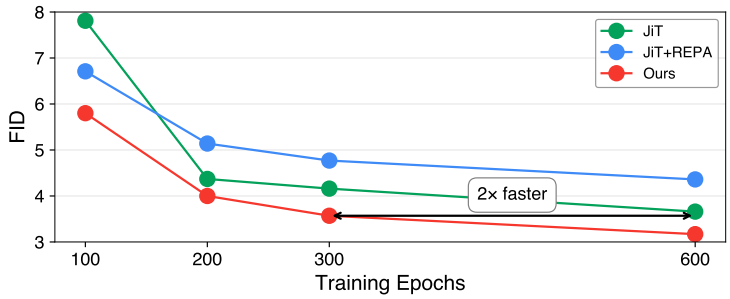


Fig. 1: REPA degrades JiT performance. As training progresses, JiT+REPA yields higher FID [13] (↓) than vanilla JiT, indicating that REPA hinders pixel space diffusion training. PixelREPA prevents overfitting to the external semantic feature target, which accelerates convergence in JiT training. Remarkably, PixelREPA achieves $> 2\times$ faster convergence than the vanilla JiT. All evaluated models utilize JiT-B/16.

generator can express [3, 10]. Just Image Transformers (JiT) [25] revisits pixel-space diffusion [6, 14, 39] and shows that a plain Vision Transformer (ViT) [7] can be trained end-to-end on raw images without any latent tokenizer or auxiliary objectives such as adversarial [9] and perceptual [47] losses, while still achieving strong generation performance. By removing the dependency on the pretrained tokenizer, pixel-space diffusion eliminates the reconstruction bottleneck and opens a path toward fully self-contained diffusion pipelines that can, in principle, represent arbitrary high-frequency detail.

Training such models, however, remains expensive. In parallel with efforts on pixel-space diffusion, a complementary line of work seeks to accelerate latent Diffusion Transformers (DiT) [33] training by injecting semantic structure from large representation encoders. Representation Alignment (REPA) [46] aligns intermediate DiT activations with features from an external semantic encoder such as DINOv2 [31], providing an explicit semantic target and dramatically speeds up convergence. Because pixel-space diffusion faces a similar, often more severe training cost, applying REPA to JiT is a natural next step.

However, we observe the opposite tendency in pixel space, as shown in Fig. 1 (JiT+REPA). REPA unexpectedly degrades performance when pixel-space diffusion training progresses. This observation raises a natural question: *why does REPA accelerate latent-space diffusion yet hinder pixel-space diffusion?*

We trace the root cause to a fundamental *information asymmetry* between the two spaces. In LDMs, the pretrained tokenizer compresses the image and suppresses much of the fine-scale, high-frequency variation [3, 8, 19]. The external semantic encoder is also a compressed representation that is largely insensitive to this fine detail [32]. Because both the denoising space and the alignment target have already passed through *information bottlenecks* [40], their degrees of freedom are roughly matched, and direct feature alignment is effective.

In pixel space, however, denoising operates in the ambient image space with $O(H \times W)$ degrees of freedom, while the semantic encoder still produces a compact, bottleneck representation. Accordingly, many pixel-distinct images therefore map to similar regions in feature space of the semantic encoder, and this

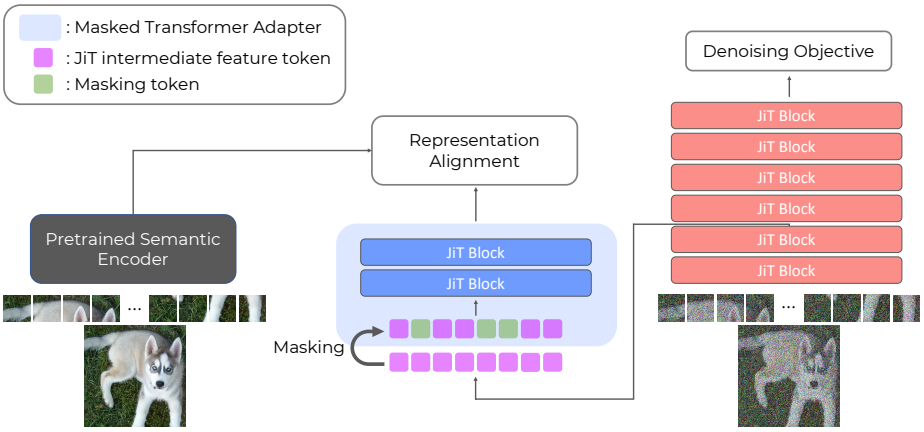


Fig. 2: Overall Framework of PixelREPA. PixelREPA masks a subset of tokens in an intermediate diffusion feature map. The full token sequence, with only a subset masked, are then transformed by a shallow Transformer adapter and aligned to features from a frozen pretrained semantic encoder. This transforms the alignment target and reduces overfitting to the external semantic representation.

ambiguity grows with resolution. Forcing the diffusion model to regress toward such a compressed target leads to *feature hacking*: the model overfits to the narrow external feature space and loses the ability to generate diverse images whose semantic features are highly similar. Our experiments confirm this analysis. REPA improves JiT at 32×32 resolution where the pixel-feature gap is small, but consistently degrades performance at 256×256 where this gap is large. Furthermore, JiT+REPA shows degraded FID compared to vanilla JiT specifically on image subsets that are tightly clustered in feature space of semantic encoder yet visually diverse in pixel space, directly evidencing feature hacking.

These findings reveal that the target of alignment matters. Standard REPA projects diffusion features into the semantic space through a point-wise Multi-Layer Perceptron (MLP) and matches them to feature space of the semantic encoder. This is effectively a feature to pixel alignment: it asks the pixel-space model to conform to a compressed feature target. When the information gap between the two spaces is large, the original REPA formulation trivially minimizes direct regression to feature space of the semantic encoder, collapsing diversity. As a result, REPA encourages intermediate JiT representations to collapse toward semantic feature. Later blocks must then reconstruct pixels from a compressed semantic feature. This semantic to image direction is ill-posed in pixel space, because many distinct images map to similar semantic features [3].

We transform this target. Rather than forcing pixel representations to match a compressed target, we map them into the semantic feature space via a ***shallow Transformer adapter*** and align them to transformed space induced by the adapter. Concretely, we extract an intermediate representation from JiT encoder, pass it through a lightweight two-block Transformer adapter, and align

the adapter output with features of the frozen semantic encoder. This adapter is trained to transform intermediate JiT features toward the semantic target to prevent feature hacking. This preserves the information needed for subsequent JiT blocks to map back to pixels while *selectively* injecting semantic structure into JiT representation. Furthermore, the adapter performs contextual aggregation via self-attention, so each token prediction can leverage information from neighboring tokens before matching $f(\cdot)$, reducing reliance on purely local cues.

A critical design choice accompanies this adapter. Without additional constraints, we find that the adapter can still learn a trivial token-wise mapping that shortcuts directly to the compressed target – empirically, an unmasked adapter improves over REPA but still falls short of vanilla JiT. To prevent this shortcut, we apply random ***partial masking*** to the adapter input. Masking serves two complementary roles. First, by removing a subset of tokens, it forces the adapter to predict the target representation under partial observation, which requires genuine contextual reasoning rather than trivial per-token projection [11]. Second, masking acts as an information bottleneck on the pixel side: it reduces the effective degrees of freedom of the pixel representation before alignment, narrowing the information gap between pixel features and the compressed semantic target. This makes the two spaces more compatible–analogous to the role the tokenizer plays in latent diffusion– without discarding information in the main denoising pathway. Together, the adapter and masking form the ***Masked Transformer Adapter (MTA)***, which turns alignment into a constrained prediction problem well-suited to high-resolution pixel-space diffusion.

This design differs from standard REPA in both the alignment module architecture and the training-time masking mechanism. REPA aligns patch-wise projections of diffusion hidden states to pretrained visual features using a trainable projection head, implemented as a MLP. Our approach replaces this MLP projection with a shallow Transformer adapter and introduces masking on the adapter input, motivated by the pixel space failure mode where a strongly compressed external target can cause direct alignment to overemphasize feature matching. Importantly, MTA is applied only on the alignment branch and does not modify the main denoising pathway; it is used only during training and therefore incurs no additional cost at inference.

In this study, we propose ***PixelREPA***, a REPA-style alignment framework designed for pixel space diffusion by replacing MLP into MTA. On ImageNet 256×256, PixelREPA-B/16 reduces FID [13] from 3.66 to 3.17 against JiT-B/16, and it achieves over $2\times$ faster convergence. PixelREPA-H/16 further reaches FID 1.81, outperforming vanilla JiT-H/16 at 1.86 and even JiT-G/16 at 1.82, which has nearly $2\times$ more parameters. These results show that PixelREPA improves both training efficiency and final generation quality at high resolution.

In summary, the core contributions of this study are as follows:

- We find REPA degrades high resolution pixel-space diffusion training and induces *feature hacking*, where direct regression to a compressed semantic target collapses diversity among samples with similar encoder features.
- We propose PixelREPA, which transforms the alignment target and constrains the alignment branch with a Masked Transformer Adapter that combines a shallow Transformer adapter with partial token masking.
- PixelREPA improves both convergence speed and generation quality on ImageNet 256×256, reducing FID from 3.66 to 3.17 for B/16 and reaching 1.81 for H/16.

2 Preliminaries

2.1 Diffusion and Flow-based Generative Models

DDPM. Diffusion models were popularized through Denoising Diffusion Probabilistic Models (DDPM) [14], which consists of a forward noising process and a learned reverse denoising process. Given a data sample $\boldsymbol{x} \sim p_{\text{data}}$ and Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for timestep $t \in \{0, \dots, T\}$, the diffusion process is defined by two main trajectories, a forward process and a reverse process. The forward process $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$ gradually corrupts the sample by adding noise according to a variance schedule β_t: $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}\big(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}\big)$. The reverse process $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is trained to denoise the corrupted sample and recover the original data: $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\Big(\boldsymbol{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\Big(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\Big), \sigma_t^2 \boldsymbol{I}\Big)$, where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. A neural network model is $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ and σ_t^2 denotes a variance schedule. Finally, the model is trained to predict the added noise by minimizing the following training objective:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{\epsilon}, t}\Big[\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\big\|_2^2\Big]. \tag{1}$$

Flow-based Geneartive Models. From a continuous-time perspective, diffusion models can also be formulated as an ODE-based flow [1, 26, 27]. In this perspective, a noisy sample $\boldsymbol{x}_t = a_t \boldsymbol{x} + b_t \boldsymbol{\epsilon}$ is an interpolation between data $\boldsymbol{x}$ and noise $\boldsymbol{\epsilon}$, with pre-defined noise schedules a_t and b_t and timestep $t \in [0, 1]$. A flow velocity at timestep t is defined as the time-derivative of $\boldsymbol{x}_t$ as $\boldsymbol{v}_t = \boldsymbol{x}_t' = a_t' \boldsymbol{x} + b_t' \boldsymbol{\epsilon}$. Under linear schedules $a_t = t$ and $b_t = 1 - t$, the corresponding velocity can be represented as $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{\epsilon}$. Flow-based models learn a velocity field that deterministically transports samples from noise to clean data, via the following velocity-matching objective:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{\epsilon}, t}\Big[\big\|\boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{v}\big\|_2^2\Big]. \tag{2}$$

2.2 Pixel-space Diffusion

Latent diffusion model (LDM) [35] is the common choice for high-resolution generation, which denoises in a compressed autoencoder latent space. LDM is efficient because operating in the lower-dimensional latent space reduces computation and memory, enabling faster training and sampling at high resolutions. However, there is a reconstruction bottleneck: generated image quality is bounded by the autoencoder [3], and strong compression can remove fine textures and small structures in latent space [19]. Also, since the autoencoder is trained for reconstruction rather than generation, this mismatch can surface as artifacts such as overly smooth textures or slight color shifts. It further adds an extra component to train and maintain, and decoding latents back to pixels adds overhead at sampling time. These limitations make pixel-space diffusion attractive.

Recent works have revisited diffusion directly in pixel space and shown that strong results are possible without an external autoencoder. SiD2 [17] scales pixel-space diffusion model with sigmoid loss weighting and a streamlined U-ViT backbone. More recently, JiT [25] achieves performance comparable to latent-space diffusion by employing a pure Transformer architecture. JiT shows that clean image prediction ($\boldsymbol{x}$-prediction) is necessary, regardless of prediction type. Formally, JiT uses $\boldsymbol{x}$-prediction [37] and velocity-matching objective:

$$\mathcal{L}_{\text{JiT}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},t}\Big[\big\|\tilde{\boldsymbol{v}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{v}\big\|_2^2\Big], \tag{3}$$

where $\tilde{\boldsymbol{v}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = (\mathbf{x}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_t)/(1-t)$ and $\mathbf{x}_{\boldsymbol{\theta}}(\cdot)$ is the $\boldsymbol{x}$-prediction network.

2.3 Representation Alignment for Generation

Recently, REPA [46] has emerged an effective approach for accelerating training and improving sample quality in DiT [33] and SiT [28]. REPA aligns intermediate diffusion features with semantic representations from a frozen pretrained encoder $f(\cdot)$. The alignment objective is simply defined as:

$$\mathcal{L}_{\text{REPA}} = -\mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},t}\Big[\frac{1}{N}\sum_{n=1}^{N}\text{cossim}(f(\boldsymbol{x})^{[n]}, h_\phi(\boldsymbol{h}_t^{[n]}))\Big], \tag{4}$$

where n is a patch index, N is the number of patches, $\boldsymbol{h}_t$ denotes an intermediate feature of diffusion Transformers at timestep t, $h_\phi(\cdot)$ indicates a projection function, and $\text{cossim}(\cdot,\cdot)$ represents a cosine-similarity function.

Given its simplicity and effectiveness, several subsequent studies have been conducted. For instance, REPA-E [24] utilizes this alignment for the end-to-end joint tuning of a VAE and a diffusion model, and Wang *et al.* [43] introduce an early termination strategy, coupled with attention alignment. Furthermore, this approach has been successfully extended to various tasks, including video generation [23,48], 3D-aware generation [20,44], and unified model training [29].

3 Motivation

We begin with our main findings and show experimental analysis to verify these findings. Figure 1 shows naïvely applying REPA [46] to JiT [25], a pixel-space diffusion model, leads to a performance degradation. REPA is a simple regularization strategy that has been shown to accelerate training convergence and improve final performance in latent space diffusion transformers such as DiT [33] and SiT [28]. These advantages provide a clear motivation to apply REPA to JiT. However, JiT+REPA underperforms vanilla JiT on ImageNet [5] 256×256 as training progresses. This gap raises a natural question: *why does REPA facilitate learning in latent space diffusion, yet struggle in pixel-space diffusion?*

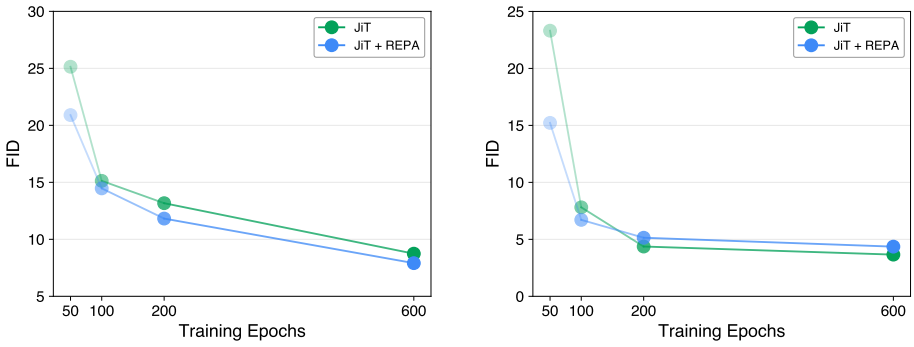
Before diving into this question, we first revisit the key differences between latent space and pixel space. The key differences between latent space and pixel space fall into two aspects [8, 35]: (1) *dimensionality of representation* and (2) *perceptual compression*.

We first focus on *dimensionality*. Latent diffusion [35] performs denoising in a compact token grid whose spatial size and channel capacity are reduced relative to the image, which substantially lowers the degrees of freedom that the denoiser must model. Pixel-space diffusion instead denoises the target in the ambient image space. For an image of resolution $H \times W$, this space contains $O(H \times W)$ degrees of freedom. As H and W increase, the number of local variations grows rapidly, and many of these variations correspond to fine scale intensity changes rather than semantic changes. This high dimensional continuous geometry makes a mapping from semantic features to fine detailed images as highly ill-posed.

Second, latent representations [35] introduce an explicit *perceptual compression*. The pretrained tokenizer maps an image into a compact code representation that prioritizes salient, reconstructable content [35]. As a result, much of the fine grained detail and high frequency variation is attenuated in the latent [19]. Pixel space retains these details in the denoising signal, including textures and micro patterns that are weakly tied to semantics. This discrepancy leads to different learning dynamics between latent and pixel-space diffusion.

3.1 Dimensionality of Representation

We now return to the main question and analyze it through the lens of these two differences. We first investigate whether the performance degradation stems from *dimensionality of representation*. Figure 3 compares JiT and JiT+REPA on ImageNet 32×32 and 256×256. This setup is designed to identify the effect of *dimensionality* on JiT+REPA by varying resolution. Figure 3(a) shows REPA improves over vanilla JiT at low resolution. In contrast, Fig. 3(b) shows REPA degrades performance as training progresses at high resolution. These results suggest that REPA becomes ineffective as the degrees of freedom increase, while remaining beneficial in low dimensional settings. As a result, this experiment presents a following finding:



(a) ImageNet 32×32 **(b) ImageNet** 256×256

Fig. 3: REPA accelerates pixel diffusion training at low resolution, whereas it degrades training at high resolution. This figure illustrates FID scores across different resolutions comparing JiT and JiT+REPA. Results show (a) ImageNet 32×32 and (b) ImageNet 256×256 with varying training epochs.

Finding 1. The failure of REPA in pixel space emerges as dimensionality of representation space increases.

3.2 Perceptual Compression

We next investigate *perceptual compression*. The latent space induced by a pretrained image tokenizer suppresses fine grained detail and high frequency variation compared to pixel space. This *perceptual compression* makes the latent denoising space more compatible with the representation space of a pretrained semantic encoder. As a result, REPA leads to faster convergence and improved performance when aligning the semantic representation to LDMs. In contrast, the alignment degrades performance in high resolution pixel-space diffusion.

We hypothesize this degradation arises because many semantically similar yet visually distinct images map to similar regions in the feature space of the pretrained encoder as high resolution pixel space has substantially more degrees of freedom. To verify this, we compare vanilla JiT and JiT+REPA across samples that are close to, or far from, a mode in the external semantic feature space. For each ImageNet class, we compute a class centroid in the feature space of the external semantic encoder $f(\cdot)$ as Fig. 4. This centroid serves as a proxy for a dense semantic mode, where many semantically similar images concentrate in the encoder representation space. We then extract two subsets: the most similar 100 samples to the centroid and the least similar 100 samples from the centroid. The most similar subset contains images that differ in pixel space, yet remain tightly clustered in $f(\cdot)$ space. Conversely, the least similar subset is widely scattered in $f(\cdot)$ space, indicating low similarity under the external semantic representation.

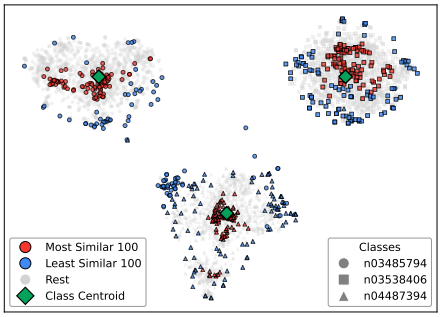
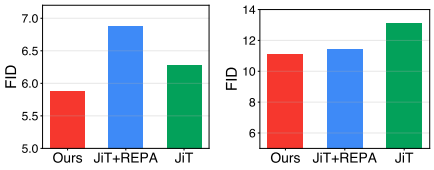


Fig. 4: Visualization of semantic representation distribution by class with t-SNE [30]. For each of classes, we compute a centroid in the semantic feature space based on feature similarity. We mark the 100 samples most similar to the centroid as red dots and the 100 samples least similar to the centroid as blue dots.



(a) Most Similar 100 **(b)** Least Similar 100

Fig. 5: REPA degrades generation diversity compared to the vanilla JiT on the most similar 100 samples for each class. This figure shows FID scores across different training data selection strategies. We compute FID across randomly selected 100 classes using 100 samples per class. Vanilla JiT achieves lower FID on the Most Similar 100 subset, whereas JiT+REPA achieves lower FID on the Least Similar 100 subset. Ours shows the best FID on both settings.

Figure 6 visualizes the two subsets defined in the external semantic feature space. As shown in Fig. 6, the most similar 100 samples share similar global structure and composition, while differing mainly in fine scale details. In contrast, the least similar 100 samples differ substantially from both structure and content. These visualizations suggest that the most similar 100 images cluster tightly and map to highly similar semantic features under the encoder, whereas the least similar 100 images are scattered at semantic space and map to distinct semantic features. These images are perturbed with diffusion noise at $t = 0.2$ to retain part of the original image signal. Each model then denoises these noisy images. (Please see supplementary materials section B.2 for extra analysis.)

As shown in Fig. 5, vanilla JiT achieves lower FID than JiT+REPA on the most similar 100 subset, while opposite holds on the least similar 100 subset. This asymmetry is the signature of feature hacking. The most similar 100 subset is precisely where *feature hacking* manifests: images are pixel-diverse yet semantically clustered, so the alignment loss drives them toward a narrow region of the feature space near the mode. On the least similar subset, where semantic targets are well-separated, alignment is informative and REPA benefits. This confirms that failure of REPA in pixel space is not a uniform degradation but a structured one: it harms generation quality specifically where the feature space is most ambiguous. We refer to this as *feature hacking*. Our second finding is:

Finding 2. In high resolution pixel-space diffusion, REPA induces *feature hacking* and hinders training on sets of images whose semantic features are highly similar under the external semantic encoder.

(a) Most Similar 100

(b) Least Similar 100

Fig. 6: Most Similar 100 and Least Similar 100 images in the external semantic feature space. (a) Most Similar 100 images to the class centroid in feature space of the external semantic encoder. (b) Least Similar 100 images to the centroid.

4 PixelREPA: REPA for Pixel Space Diffusion Models

Our analysis identifies two causes behind REPA [46] failure in pixel-space diffusion: (1) the *dimensionality of representation*, and (2) the *perceptual compression*. Both stem from alignment target of REPA. REPA projects the intermediate features of JiT [25] through a point-wise MLP into the representation space of pretrained semantic encoder and aligns it there. This pulls a compressed semantic representation toward the diffusion space. In latent diffusion, where the diffusion feature is already compact, this gap is manageable. In pixel space, the JiT features carry far richer information than $f(\cdot)$. Then, the MLP enforces the JiT features to conform to the compressed $f(\cdot)$, without learning the fine-grained structure needed for high-quality pixel generation. Later diffusion blocks must then reconstruct diverse pixel outputs from a compressed semantic code—an ill-posed mapping since many distinct images share similar $f(\cdot)$.

We address this by transforming the alignment target and constraining the alignment pathway. Rather than the JiT representations targets to learn $f(\cdot)$, we transform the semantic target through a dedicated module and *align intermediate features into the transformed space induced by the module*. This module, namely the Masked Transformer Adapter (MTA), consists of two components: (1) a *shallow transformer adapter* and (2) a *partial masking strategy*. The shallow transformer adapter performs contextual aggregation over diffusion tokens, so each token prediction can leverage information from other tokens. The partial masking applies random token masking to the adapter input, which regularizes alignment to discouraging overly direct regression to $f(\cdot)$ and acts as an information bottleneck. Figure 2 describes the overall framework of PixelREPA.

4.1 Shallow Transformer Adapter

We introduce a shallow Transformer adapter for transforming alignment target. The adapter transforms an intermediate diffusion feature from the JiT encoder into the external semantic feature space for matching $f(\cdot)$. This adapter consists

of two Transformer blocks with self-attention. The critical difference from the original MLP-based alignment is twofold.

First, the adapter *selectively* learns the transformation from the JiT features into $f(\cdot)$ to transform alignment target. This means the alignment objective no longer pressures the JiT intermediate representation to match a compressed target. Instead, the adapter learns to extract semantic content from the JiT representation and project it into the space of $f(\cdot)$. As a result, the alignment target of the JiT features is *inversely* transformed feature by the adapter from the space of $f(\cdot)$. The JiT features remain free to encode the full range of pixel-level detail needed for high-quality generation, while the adapter selectively distills the semantic signal for alignment.

Second, the adapter performs contextual aggregation via self-attention. Each token prediction incorporates information from neighboring tokens before matching $f(\cdot)$, rather than being mapped in isolation. This forces the adapter to build semantic predictions from a broader spatial context, producing a more structured transformation than a point-wise mapping. Contextual aggregation also reduces reliance on per-token correspondence, weakening the trivial regression pathway that underlies feature hacking.

The adapter remains lightweight by using only two transformer blocks, while providing a more structured and stable alignment pathway than an MLP head in high resolution pixel space diffusion.

In the perspective of self-supervised learning [2], JiT can be composited of two functions as $\mathbf{x}_\theta = g_\theta \circ f_\theta$, where the encoder $f_\theta : \mathcal{X} \to \mathcal{H}$ and the decoder $g_\theta : \mathcal{H} \to \mathcal{X}$. The encoder $f_\theta(\boldsymbol{x}_t)$ maps a noisy image $\boldsymbol{x}_t$ to the intermediate representation $\boldsymbol{h}_t$, as $f_\theta(\boldsymbol{x}_t) = \boldsymbol{h}_t \in \mathcal{H}$. Image space, hidden space, and semantic feature space are denoted as $\mathcal{X}, \mathcal{H}$, and $\mathcal{R}$, respectively. Finally, the transformer adapter $d_\phi : \mathcal{H} \to \mathcal{R}$ transforms $\boldsymbol{h}_t$ to predict the external semantic feature.

4.2 Partial Masking Strategy

The shallow Transformer adapter transforms the alignment target and introduces contextual aggregation, but is not sufficient on its own. Without additional constraints, the adapter can still learn a near-trivial mapping from the JiT representation to $f(\cdot)$ and our experiments confirm this: an unmasked adapter (mask ratio 0.0) achieves FID 4.68 at 200 epochs, which improves over JiT+REPA (5.14) but falls behind vanilla JiT (4.37), as demonstrated in Tab. 3. This intermediate result reveals that transforming the alignment target reduces but does not eliminate the shortcut, because the adapter can still exploit per-token correspondence between $\boldsymbol{h}_t$ and $f(\boldsymbol{x})$ when all tokens are visible.

We propose a partial masking strategy on the adapter input to address this residual shortcut. During training, we randomly mask a fraction r of tokens in the intermediate JiT feature map before passing the shallow transformer adapter. The adapter must then predict the full semantic target from partial observations, using the neighboring tokens as context.

Masking serves two complementary roles. (1) Shortcut prevention: By removing a subset of input tokens, masking breaks the per-token correspondence

between the JiT representations and semantic features. This requires genuine contextual reasoning and prevents the trivial regression pathway. (2) Information bottleneck [40] on the pixel side: Masking reduces the effective degrees of freedom of the adapter input from $O(N \cdot d)$ to $O((1-r) \cdot N \cdot d)$, where N is the number of tokens, d is the hidden dimension, and r is the mask ratio. This narrows the information gap between the pixel representation and the compressed target, analogous to the dimensionality reduction that tokenizer performs in latent diffusion, but applied selectively to the alignment pathway rather than to the denoising process itself. The main denoising pathway retains the full, unmasked token sequence.

Finally, PixelREPA is defined as follows:

$$\mathcal{L}_{\text{PixelREPA}} := -\mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},t}\left[\frac{1}{N}\sum_{n=1}^{N}\text{cossim}(f(\boldsymbol{x})^{[n]}, d_\phi(m \odot \boldsymbol{h}_t^{[n]})\right], \tag{5}$$

where m denotes a patch-wise mask. The final objective function becomes:

$$\mathcal{L} := \mathcal{L}_{\text{JiT}} + \lambda\mathcal{L}_{\text{PixelREPA}}, \tag{6}$$

where $\lambda > 0$ is a regularization hyperparameter.

5 Experiments

5.1 Setup

Implementation details. Our implementation and configuration strictly follow the implementation of JiT [25]. Specifically, each model configuration follows JiT paper across all sizes, except for MTA. We use a 2-layer transformer adapter, with masking ratio $r = 0.2$, regardless of the model size. For external semantic encoder, we employ DINOv2 [31] as REPA [46]. The intermediate features input to the MTA were obtained from the layer directly prior to the in-context start block, specific to each model size. We fix a regularization hyperparameter $\lambda = 0.1$ for every model size. Our experiments are conducted on 8 NVIDIA H200 GPUs. The detailed configurations are described in the supplementary materials.

Evaluation. We evaluate FID [13] and Inception Score (IS) [36] with 50K samples, which is a common setting. Following JiT, we use a 50-step Heun [12] ODE solver with CFG [15] interval $[0.1, 1]$ [22].

5.2 Analysis

Comparisons on ImageNet 256×256. As shown in Tab. 1, PixelREPA consistently outperforms JiT across all model scales. For the B/16 architecture, PixelREPA reduces the FID from 3.66 to 3.17 (13.4%) improvement. This trend holds for larger models: the L/16 variant yields 10.6% relative gain, and the H/16

Table 1: Quantitative comparison of diffusion models on ImageNet 256×256. FID and IS are evaluated with 50K samples.

Model	params	FID↓	IS↑
Latent-space Diffusion			
DiT-XL/2 [34]	675+49M	2.27	278.2
SiT-XL/2 [28]	675+49M	2.06	277.5
REPA [46], SiT-XL/2	675+49M	1.42	305.7
LightningDiT-XL/2 [45]	675+49M	1.35	295.3
DDT-XL/2 [42]	675+49M	1.26	310.6
RAE [49], DiT^{DH}-XL/2	839+415M	1.13	262.6
Pixel-space Diffusion			
ADM-G [6]	559M	7.72	172.7
RIN [18]	320M	3.95	216.0
SiD [16]	2B	2.44	256.3
VDM++ [21], UViT/2	2B	2.12	267.7
SiD2 [17], UViT/2	-	1.73	-
SiD2 [17], UViT/1	-	1.38	-
PixelFlow [4], XL/4	677M	1.98	282.1
PixNerd [41], XL/16	700M	2.15	297.0
JiT-B/16 [25]	131M	3.66	275.1
JiT-L/16 [25]	459M	2.36	298.5
JiT-H/16 [25]	953M	1.86	303.4
JiT-G/16 [25]	2B	1.82	292.6
PixelREPA-B/16	131M	3.17	284.6
PixelREPA-L/16	459M	2.11	309.5
PixelREPA-H/16	953M	1.81	317.2

variant shows further 2.7% improvement. These consistent improvements confirm that PixelREPA is robust to the scalability. Notably, PixelREPA-H/16 surpasses JiT-G/16, nearly 2× larger model, demonstrating more effective parameter utilization. Furthermore, PixelREPA achieves competitive results against recent pixel-space diffusion models without any modifications of transformer architecture, showcasing its robustness in pixel-level generation. (Please see supplementary materials section B for results on T2I and 512×512 resolution.)

Effectiveness of partial masking. First, we verify the effectiveness of partial marking by comparing whether mask is used. MTA without masking surpasses JiT+REPA but falls behind the JiT baseline as Tab. 3. This shows our transformer adapter improves the generation performance than the standard REPA but suffers from accelerating JiT training. Therefore, partial masking is essential

Table 2: Ablation study for masking ratio. This comparison is evaluated on ImageNet 256×256 by FID with same model size B/16. Red colored box is the best result.

Mask ratio	0.1	0.2	0.3	0.4	0.5
200-ep	4.26	4.00	4.38	4.32	4.58

Table 3: Ablation study for masking. All models are trained on ImageNet 256 × 256, evaluated by FID. The size of all models are fixed on B/16. PixelREPA† is PixelREPA without masking. Red colored box indicates the best result.

Model	JiT	JiT+REPA	PixelREPA†	PixelREPA
200-ep	4.37	5.14	4.68	4.00

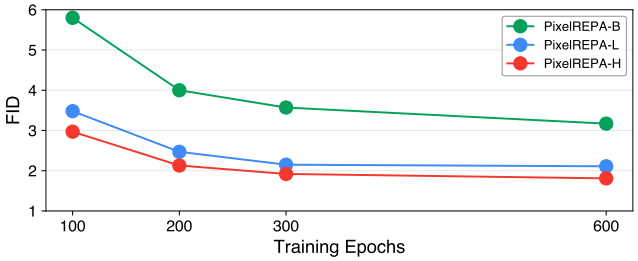


Fig. 7: Scalability. As model size grows, PixelREPA shows better performance.

to mitigate feature hacking. This constraint discourages shortcut learning and reduces overfitting to the external semantic feature.

To investigate the effect of partial masking ratio, we compare PixelREPA by varying mask ratios. As shown in Tab. 2, the best performance is achieved at mask ratio $r = 0.2$. However, further increasing mask ratio r to 0.5 leads to performance degradation. Masking removes supervision on the masked subset and then blocks the gradient signal to the JiT. As a result, large mask ratio hinders JiT blocks to learn semantic features due to excessive information bottleneck, leading to the degraded performance. Based on this analysis, we use a mask ratio of 0.2 for all PixelREPA models. (Please refer to supplementary materials section B for extra .)

REPA vs. PixelREPA on JiT. PixelREPA effectively overcomes the inferior performance of REPA on JiT. As illustrated in Fig. 3(b) and Tab. 3, JiT+REPA improves upon JiT early in training. However, this trend reverses with prolonged training, ultimately resulting in a 17.6% degradation in FID (5.14 vs. 4.37) compared to the baseline at 200 epochs. In contrast, PixelREPA consistently outperforms JiT, achieving an 8.5% improvement over vanilla JiT at 200 epoch. This suggests that PixelREPA stabilizes the alignment for pixel-space diffusion by reducing overfitting in the limited target feature space, making it a suitable alternative to the standard REPA in the high resolution image setting.

Figure 5 shows PixelREPA achieves the best FID scores at both the most and the least similar 100 samples. However, JiT+REPA struggles on the most

similar 100 setting as discussed at Sec. 3. This verifies PixelREPA is robust to synthesize images at the near centroids and thus mitigates feature hacking.

Scalability. We investigate the scalability of PixelREPA by varying model size. PixelREPA achieves lower FID as the model scales up, as Tab. 1 and Fig. 7. The improvement is consistent across training epochs at each model size, indicating better sample quality and diversity. PixelREPA also consistently outperforms vanilla JiT at matched model sizes.

6 Conclusion

We revisit representation alignment for JiT and identify a failure mode of standard REPA at high resolution, where alignment to a compressed semantic target leads to feature hacking and degraded training. PixelREPA addresses this issue by transforming the alignment target and constraining the alignment pathway with a shallow Transformer adapter and partial token masking. The resulting Masked Transformer Adapter stabilizes optimization, scales with model size, and improves ImageNet 256×256 results across JiT backbones. PixelREPA reduces FID from 3.66 to 3.17 for B/16 and achieves 1.81 for H/16.

Acknowledgements

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (No. RS-2025-00520207); Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (Nos. RS-2024-00457882, 2022-0-01045, 2022-0-00680, RS-2019-II190075); the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea (Ministry of Science and ICT); the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT), Republic of Korea (No. RQT-25-120217); a grant partly supported by both IITP (MSIT) and Korea Evaluation Institute of Industrial Technology (KEIT) (MOTIE) (No. RS-2025-02217259); and the next-generation CultureTech technology development (R&D) project for cultural space AX transformation through the Korea Creative Content Agency (KOCCA) grant funded by the Ministry of Culture, Sports and Tourism (MCST) in 2026 (No. RS-2026-25508598).

References

1. Albergo, M.S., Vanden-Eijnden, E.: Building normalizing flows with stochastic interpolants. arXiv preprint arXiv:2209.15571 (2022)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
3. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6228–6237 (2018)
4. Chen, S., Ge, C., Zhang, S., Sun, P., Luo, P.: Pixelflow: Pixel-space generative models with flow. arXiv preprint arXiv:2504.07963 (2025)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
9. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
10. Gu, Y., Wang, X., Ge, Y., Shan, Y., Shou, M.Z.: Rethinking the objectives of vector-quantized tokenizers for image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7631–7640 (2024)
11. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
12. Heun, K., et al.: Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. Z. Math. Phys **45**(23-38), 7 (1900)
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
15. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
16. Hoogeboom, E., Heek, J., Salimans, T.: simple diffusion: End-to-end diffusion for high resolution images. In: International Conference on Machine Learning. pp. 13213–13232. PMLR (2023)
17. Hoogeboom, E., Mensink, T., Heek, J., Lamerigts, K., Gao, R., Salimans, T.: Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18062–18071 (2025)
18. Jabri, A., Fleet, D., Chen, T.: Scalable adaptive computation for iterative generation. arXiv preprint arXiv:2212.11972 (2022)

19. Jiang, L., Dai, B., Wu, W., Loy, C.C.: Focal frequency loss for image reconstruction and synthesis. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13919–13929 (2021)
20. Kim, J., Lee, S., Shin, J., Choi, J., Shim, H.: Dreamcatalyst: Fast and high-quality 3d editing via controlling editability and identity preservation. arXiv preprint arXiv:2407.11394 (2024)
21. Kingma, D., Gao, R.: Understanding diffusion objectives as the elbo with simple data augmentation. Advances in Neural Information Processing Systems **36**, 65484–65516 (2023)
22. Kynkäänniemi, T., Aittala, M., Karras, T., Laine, S., Aila, T., Lehtinen, J.: Applying guidance in a limited interval improves sample and distribution quality in diffusion models. Advances in Neural Information Processing Systems **37**, 122458–122483 (2024)
23. Lee, D., Jeong, H., Kim, J., Ceylan, D., Ye, J.C.: Improving video diffusion transformer training by multi-feature fusion and alignment from self-supervised vision encoders. arXiv preprint arXiv:2509.09547 (2025)
24. Leng, X., Singh, J., Hou, Y., Xing, Z., Xie, S., Zheng, L.: Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18262–18272 (2025)
25. Li, T., He, K.: Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720 (2025)
26. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
27. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
28. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: European Conference on Computer Vision. pp. 23–40. Springer (2024)
29. Ma, Y., Liu, X., Chen, X., Liu, W., Wu, C., Wu, Z., Pan, Z., Xie, Z., Zhang, H., Yu, X., et al.: Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7739–7751 (2025)
30. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
31. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
32. Park, N., Kim, W., Heo, B., Kim, T., Yun, S.: What do self-supervised vision transformers learn? arXiv preprint arXiv:2305.00729 (2023)
33. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023)
34. Pernias, P., Rampas, D., Richter, M.L., Pal, C.J., Aubreville, M.: Würstchen: An efficient architecture for large-scale text-to-image diffusion models. arXiv preprint arXiv:2306.00637 (2023)
35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
36. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29** (2016)

37. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
38. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. pmlr (2015)
39. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019)
40. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)
41. Wang, S., Gao, Z., Zhu, C., Huang, W., Wang, L.: Pixnerd: Pixel neural field diffusion. arXiv preprint arXiv:2507.23268 (2025)
42. Wang, S., Tian, Z., Huang, W., Wang, L.: Ddt: Decoupled diffusion transformer. arXiv preprint arXiv:2504.05741 (2025)
43. Wang, Z., Zhao, W., Zhou, Y., Li, Z., Liang, Z., Shi, M., Zhao, X., Zhou, P., Zhang, K., Wang, Z., et al.: Repa works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. arXiv preprint arXiv:2505.16792 (2025)
44. Wu, H., Wu, D., He, T., Guo, J., Ye, Y., Duan, Y., Bian, J.: Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. arXiv preprint arXiv:2507.07982 (2025)
45. Yao, J., Yang, B., Wang, X.: Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15703–15712 (2025)
46. Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940 (2024)
47. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
48. Zhang, X., Liao, J., Zhang, S., Meng, F., Wan, X., Yan, J., Cheng, Y.: Videorepa: Learning physics for video generation through relational alignment with foundation models. arXiv preprint arXiv:2505.23656 (2025)
49. Zheng, B., Ma, N., Tong, S., Xie, S.: Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690 (2025)