Coevolving Representations in Joint Image-Feature Diffusion

Theodoros Kouzelis[1,4⋆], Spyros Gidaris[3], and Nikos Komodakis[1,2,5]

[1]Archimedes, Athena RC [2]University of Crete [3]valeo.ai
[4]National Technical University of Athens [5]IACM-Forth

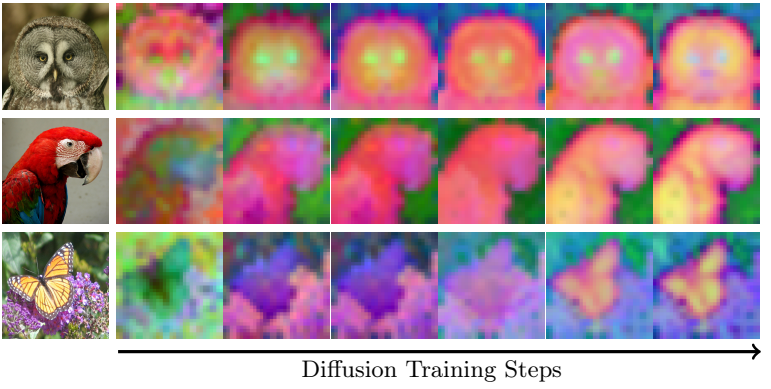

Fig. 1: Evolution of the representations throughout `CoReDi` training. As training progresses, the coevolving representations develop increasingly structured and semantically meaningful spatial organization.

Abstract. Joint image–feature generative modeling has recently emerged as an effective strategy for improving diffusion training by coupling low-level VAE latents with high-level semantic features extracted from pre-trained visual encoders. However, existing approaches rely on a fixed representation space, constructed independently of the generative objective and kept unchanged during training. We argue that the representation space guiding diffusion should itself adapt to the generative task. To this end, we propose Coevolving Representation Diffusion (`CoReDi`), a framework in which the semantic representation space evolves during training by learning a lightweight linear projection jointly with the diffusion model. While naïvely optimizing this projection leads to degenerate solutions, we show that stable coevolution can be achieved through a combination of stop-gradient targets, normalization, and targeted regularization that prevents feature collapse. This formulation enables the semantic space to progressively specialize to the needs of image synthesis, improving its complementarity with image latents. We apply `CoReDi` to both VAE latent diffusion and pixel-space diffusion, demonstrating that

⋆ Corresponding author: theodoros.kouzelis@athenarc.gr
Code is available at https://github.com/zelaki/CoReDi

adaptive semantic representations improve generative modeling across both settings. Experiments show that CoReDi achieves faster convergence and higher sample quality compared to joint diffusion models operating in fixed representation spaces.

1 Introduction

Diffusion models [11,17,35] have become the dominant paradigm for high-fidelity image synthesis. Most modern systems operate either in pixel space or in compressed VAE latent spaces, modeling low-level image statistics with remarkable precision. However, they do not explicitly leverage the rich semantic structure captured by large pretrained visual encoders.

Recent work has explored incorporating semantic priors into diffusion. Approaches include aligning pretrained representations with VAE latents or intermediate diffusion features [24, 40, 48, 49], replacing VAE latents with such pretrained representations [38, 42, 52], or *jointly modeling* low-level VAE latents and high-level semantic representations within a unified diffusion process [22, 32, 34, 47].

In this latter joint modeling paradigm, the semantic representation space serves as an auxiliary high-level space that complements the low-level VAE latents, forcing the generative model to capture both precise local details (via the VAE latents) and semantic structure (via the representation space). However, in existing joint approaches, the semantic representation space—over which the joint diffusion process operates—is constructed independently of the generative objective and remains fixed during training. In practice, PCA or lightweight autoencoders are used to project the typically high-dimensional semantic features into a more compact space (i.e., with fewer channels) [32]. The diffusion model is then trained to learn the distribution of this predefined projection space. Crucially, the representation space itself is not optimized for the generative objective.

This raises a fundamental question: *Should the representation space guiding diffusion remain fixed, or should it adapt jointly with the generative model?*

Coevolving Representation Diffusion. We introduce *Coevolving Representation Diffusion* (CoReDi), a framework in which the projection of pretrained visual features is learned jointly with the diffusion model. Instead of using a predefined mapping (e.g. PCA), we train a learnable projection from a frozen visual encoder whose output coevolves with the generative model under the joint diffusion objective. The representation space is therefore no longer an externally imposed target, but a learnable component optimized directly for image synthesis, as illustrated by the evolution of the learned representations throughout training in Fig. 1.

Naïvely optimizing the projection with the joint diffusion loss leads to degenerate solutions, since both the representation input and its denoising target

become trainable. Through systematic analysis, we identify three necessary ingredients for stable coevolution:

1. **Stop-gradient in the representation diffusion target**, preventing trivial minimization of the representation diffusion loss.
2. **Batch normalization after the projection**, which stabilizes feature scale, preserves the intended noise schedule, and avoids per-channel sample collapse.
3. **Explicit regularization against feature collapse.** Even with stop-gradient and batch normalization, we observe *feature collapse*, where multiple channels encode redundant information or fail to capture meaningful variation. To address this, we introduce explicit regularization terms that enforce diversity and information preservation in the projected space. We explore simple yet effective strategies, including feature-variance regularization, orthogonality constraints on the projection weights, and covariance regularization to discourage feature collapse. These regularizers play a central role in ensuring that the learned representation remains expressive and complementary to the image latents.

We demonstrate empirically that all three components are necessary for effective coevolution.

Beyond VAE Latents. Finally, we ask whether joint image–feature diffusion must rely on `VAE` latents at all. While `VAE` latents provide computational efficiency, they introduce a reconstruction bottleneck that may limit ultimate image fidelity. Since the auxiliary semantic space already enforces high-level structure, we show that `CoReDi` extends naturally to pixel-space diffusion, removing the reconstruction bottleneck imposed by `VAE` compression. Building on `DeCo` [29], we develop an efficient pixel-space variant in which pretrained visual features coevolve jointly with raw pixels during training, yielding substantial improvements over the baseline `DeCo` pixel diffusion model and demonstrating that adaptive semantic guidance remains beneficial beyond latent-based generation.

Contributions. In summary, our contributions are as follows:

- We propose `CoReDi`, a framework for jointly modeling images and semantic representations in which the representation space itself coevolves with the diffusion model.
- We identify and analyze three necessary ingredients for stable training, highlighting the critical role of explicit regularization in preventing feature collapse.
- We show that coevolving representations improve joint diffusion in both `VAE` latent space and pixel space (see Fig. 2).

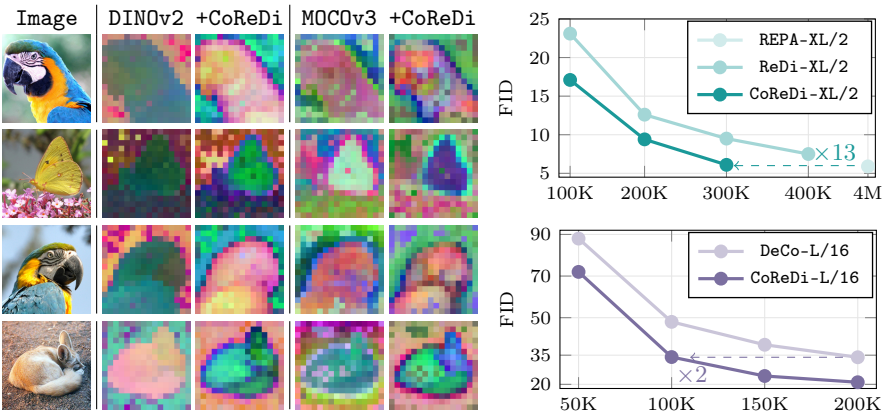

Fig. 2: (Left) Comparison of fixed PCA and learned `CoReDi` representations for `DINOv2` and `MOCOv3`. The learned projections yield cleaner, more structured representations with coherent spatial organization, while the fixed PCA projections produce noisier, less semantically meaningful activations. (Right) By jointly adapting the representation space alongside the generative model, `CoReDi` consistently speeds up convergence. In latent space (Top), `CoReDi` outperforms `ReDi` and, notably, converges $\sim 13\times$ faster than `REPA`. In pixel space (Bottom) it improves convergence by $\times 2$ over `DeCo`.

2 Related Work

Latent Diffusion Models. Latent Diffusion Models [28, 33, 35, 46, 53] operate in the compressed latent space of a variational autoencoder (`VAE`) [21, 35, 48], which reduces spatial dimensionality compared to pixel-space diffusion, significantly lowering computational cost and learning difficulty [35]. The Diffusion Transformer (`DiT`) [33] marked a significant architectural shift by replacing the U-Net [36] backbone with a transformer, and SiT [28] extended this framework to flow-based diffusion objectives.

Pixel Space Diffusion. Recently, there has been a surge of interest in pixel diffusion models, since they avoid the reconstruction bottleneck imposed by the `VAE`. Early approaches relied on multi-stage pipelines operating at progressively increasing resolutions to manage the high dimensionality of pixel space [7, 41], at the expense of more complex training and inference procedures. More recent work has explored alternative architectures to sidestep these issues, including transformer-based normalizing flows [51], fractal generative models [26], `DiT`-based models that predict neural field parameters per patch [45], and methods that predict the clean image directly to anchor generation to the low-dimensional data manifold [25]. `DeCo` [29] decouples the generation of high and low frequency components, leveraging a lightweight pixel decoder to reduce the complexity of direct pixel synthesis. Despite these advances, the integration of visual represen-

tations into pixel diffusion models to enhance generative performance remains largely unexplored.

Semantic Representations in Generative Models. Recent work has explored leveraging semantic representations [10, 29, 31, 43, 44] to enhance generative modeling [6, 19, 22, 32, 34, 38, 39, 42, 47, 49, 52]. REPA [49] aligns diffusion features with pretrained visual encoders, while REPA-E [24] enables end-to-end joint optimization of the VAE and diffusion model, and iREPA [40] improves the spatial structure of the representation used for alignment. Many recent works [6, 38, 39, 42, 52] replace the VAE with pretrained visual encoder representations, improving generation. However, by keeping the encoder frozen, these methods fall behind state-of-the-art VAEs in reconstruction quality [4]. REG [47] and ReDi [22] jointly model low-level VAE features and high-level semantic features from DINOv2 [31], with ReDi using PCA-compressed patch embeddings. Instead of relying on static representations, we propose allowing the input representation to evolve dynamically during training, further improving generation quality.

Preventing Representation Collapse in Self-Supervised Learning. Self-supervised approaches are often prone to representational collapse, and several mechanisms have been proposed to address this. Redundancy-reduction methods such as Barlow Twins [50], VICReg [2], and W-MSE [12] decorrelate features to avoid degenerate solutions, with VICRegL [3] extending this to local features. SimCLR [8] highlights batch normalization as an implicit collapse prevention mechanism, coupling examples within a batch to discourage trivial constant representations. Architectural approaches, BYOL [14], SimSiam [9], and DINO [5, 31] instead break gradient symmetry via stop-gradients, momentum encoders, or output centering.

3 Method

In this section, we describe the CoReDi framework for jointly modeling images and coevolving semantic representations under a diffusion objective. We begin by reviewing the preliminary setup of joint image-feature synthesis (Sec. 3.1). Next, we introduce our learnable projection, allowing the semantic representation space to evolve alongside the generative model, including stabilization techniques such as batch normalization and stop-gradient (Sec. 3.2). We then discuss explicit regularization strategies designed to prevent feature collapse and ensure diversity in the learned representations (Sec. 3.3). Finally, we present the overall training objective (Sec. 3.4) and describe a natural extension of CoReDi to pixel-space diffusion (Sec. 3.5).

3.1 Preliminary: Joint Image-Feature Synthesis

Joint Flow Matching Objective. The joint image–feature generation framework of [22] trains a single flow matching model [1, 13, 27] to jointly capture low-level image structure and high-level semantic information. Given an image latent

$\mathbf{x}_0 \sim p(\mathbf{x})$ and its corresponding visual representation $\mathbf{z}_0 = \texttt{VE}(\mathbf{x}_0) \in \mathbb{R}^{L \times D}$ extracted by a frozen pretrained encoder VE where L is the number of spatial tokens and D is the feature dimension. Since the dimensionality of the semantic features D greatly exceeds that of the image latents, [22] reduces it via a fixed PCA projection $\mathbf{P} \in \mathbb{R}^{D \times d}$, computed once prior to training, yielding $\tilde{\mathbf{z}}_0 = \mathbf{z}_0 \mathbf{P} \in \mathbb{R}^{L \times d}$. Using $\mathbf{x}_0, \tilde{\mathbf{z}}_0$ with $d \ll D$, the coupled interpolation process is defined as

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}_x, \quad \tilde{\mathbf{z}}_t = (1-t)\tilde{\mathbf{z}}_0 + t\boldsymbol{\epsilon}_z, \tag{1}$$

A network $\mathbf{v}_\theta(\mathbf{x}_t, \tilde{\mathbf{z}}_t, t)$ then predicts the velocities for both modalities via two heads, $\mathbf{v}^x_\theta(\mathbf{x}_t, \tilde{\mathbf{z}}_t, t)$ and $\mathbf{v}^z_\theta(\mathbf{x}_t, \tilde{\mathbf{z}}_t, t)$. Training minimizes the joint flow-matching objective:

$$\mathcal{L}_{\text{joint}}(\mathbf{x}_0, \tilde{\mathbf{z}}_0, t) = \underbrace{\|\mathbf{v}^x_\theta(\mathbf{x}_t, \tilde{\mathbf{z}}_t, t) - (\boldsymbol{\epsilon}_x - \mathbf{x}_0)\|^2}_{\mathcal{L}_{\text{image}}} + \lambda_z \underbrace{\|\mathbf{v}^z_\theta(\mathbf{x}_t, \tilde{\mathbf{z}}_t, t) - (\boldsymbol{\epsilon}_z - \tilde{\mathbf{z}}_0)\|^2}_{\mathcal{L}_{\text{rep}}}, \tag{2}$$

where λ_z balances the contribution of the representation loss $\mathcal{L}_{\text{rep}}$ relative to the image loss $\mathcal{L}_{\text{image}}$.

Merged Tokens Strategy. We adopt the merged tokens strategy [22] to fuse image and representation. Both modalities are embedded separately and summed channel-wise, $\mathbf{h}_t = \mathbf{x}_t \mathbf{W}^x_{\text{emb}} + \tilde{\mathbf{z}}_t \mathbf{W}^z_{\text{emb}}$, before being processed by the transformer. Separate predictions for each modality are then obtained via modality-specific decoding heads, $\mathbf{v}^x_\theta = \mathbf{o}_t \mathbf{W}^x_{\text{dec}}$ and $\mathbf{v}^z_\theta = \mathbf{o}_t \mathbf{W}^z_{\text{dec}}$ where $\mathbf{o}_t$ is the output of the diffusion transformer. This early fusion approach enables joint modeling of both modalities while preserving the original token count, incurring no additional computational overhead over the standard diffusion transformer.

3.2 Coevolving Representation Diffusion

While [22] reduces the dimensionality of the pretrained visual representation using a *fixed* PCA projection, we instead learn an adaptive projection $g_\phi(\cdot)$ *jointly* with the generative model. Concretely, given the frozen encoder output $\mathbf{z}_0 = \texttt{VE}(\mathbf{x}_0)$, we replace the fixed mapping with a learnable projection:

$$\tilde{\mathbf{z}}_0 = g_\phi(\mathbf{z}_0), \tag{3}$$

Unlike the fixed PCA projection, g_ϕ adapts throughout training, allowing the representation space to evolve alongside the generative model to better assist image synthesis. In this work, we explore g_ϕ as a simple trainable linear layer $g_\phi(\mathbf{z}_0) = \mathbf{z}_0 \mathbf{W}_\phi$ with $\mathbf{W}_\phi \in \mathbb{R}^{D \times d}$ where D is the feature dimension of VE.

Batch Normalization. Diffusion models are highly sensitive to input scale, as variations in feature statistics implicitly distort the intended noise schedule and destabilize training. To mitigate this, we apply batch normalization after the

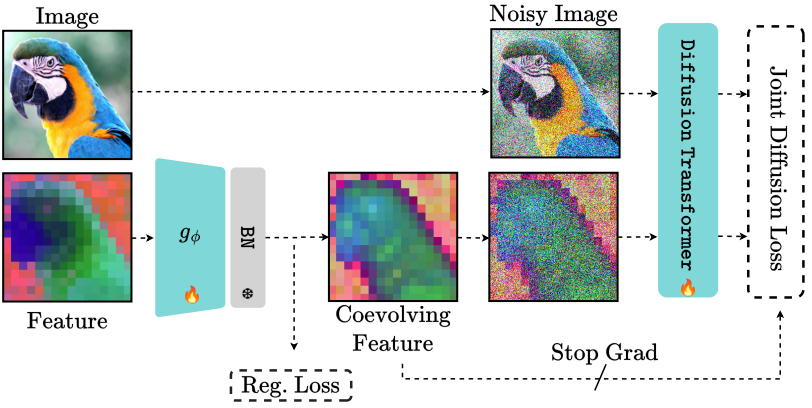

Fig. 3: Overview of CoReDi. Given an input image, a frozen pretrained visual encoder extracts semantic features, which are projected to a lower-dimensional space via a learnable projection g_ϕ, followed by batch normalization and a regularization loss to prevent collapse. Both the noisy image tokens and the noisy coevolving feature tokens are passed as input to a diffusion backbone, which jointly predicts the image and representation velocities. A stop-gradient is applied through the clean representation target in the representation loss, allowing the projection to coevolve with the generative model without degeneracy.

learnable projection using exponential moving average estimates of the mean and variance. Beyond scale stabilization, batch normalization acts as an implicit regularizer against sample collapse, enforcing a non-degenerate distribution over samples at each feature channel. We omit the standard trainable affine parameters (scale and shift), as the purpose of normalization here is solely to control input scale and prevent collapse, rather than to allow the network to rescale or shift the normalized features.

Stop-Gradient. Directly optimizing the projection g_ϕ via Eq. 2 leads to degenerate solutions since both the input to the model and the target are trainable. To stabilize training, we stop gradients through the *clean* projected target used in the representation velocity loss:

$$\mathcal{L}_{\text{rep}}(\mathbf{x}_0, \tilde{\mathbf{z}}_0, t) = \|\mathbf{v}_\theta^z(\mathbf{x}_t, \tilde{\mathbf{z}}_t, t) - (\boldsymbol{\epsilon}_z - \mathtt{sg}(\tilde{\mathbf{z}}_0))\|^2, \tag{4}$$

where $\mathtt{sg}(\cdot)$ denotes the stop-gradient operator. In this way, the diffusion model learns to jointly denoise image and representation tokens, while the representation space can coevolve without allowing the representation target itself to be trivially modified to reduce the loss.

3.3 Regularization Methods

Batch normalization implicitly prevents *sample collapse*, where different images or spatial locations are projected to the same feature point. However, we observe

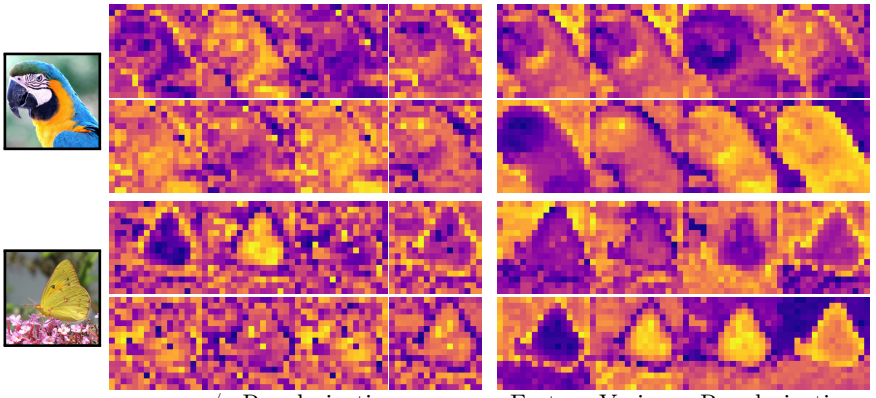

Fig. 4: Regularization Prevents Feature Collapse. Visualization of all 8 channels of the coevolving representation $\tilde{\mathbf{z}}_0$ at 200K steps, under two training configurations. Without regularization, the projected channels collapse, failing to capture diverse semantic information. The Feature Variance Regularization strategy successfully prevents collapse, yielding semantically meaningful channel activations.

that learned projections can still exhibit *feature collapse*, where individual feature channels fail to vary meaningfully across samples or fail to carry variation that is distinct from other channels (Fig. 4, top left). To address this, we explore the following regularization strategies.

Feature Variance Regularization. To prevent feature collapse, we encourage each channel of the representation to exhibit sufficient variation. Unlike `VICReg` [2], which enforces variance across the batch dimension, we instead consider each feature vector $\tilde{\mathbf{z}}_0^i$ (at location i) and penalize feature vectors whose standard deviation across the channel dimension falls below a threshold γ via a hinge loss:

$$\mathcal{L}_{\text{var}}(\tilde{\mathbf{z}}_0) = \frac{1}{L}\sum_{i=1}^{L} \max\left(0, \gamma - \sqrt{\text{Var}(\tilde{\mathbf{z}}_0^i) + \epsilon}\right),$$

where L is the number of spatial tokens, ϵ ensures numerical stability, and $\gamma = 1$ sets the minimum desired standard deviation. This encourages each channel to remain active and carry meaningful variation, preventing feature collapse and redundancy across channels.

Orthogonality Regularization. As an alternative to penalizing feature collapse at the feature level, we instead regularize the weight matrix $\mathbf{W}_\phi$ of the linear projection $\tilde{\mathbf{z}}_0 = \mathbf{z}_0\mathbf{W}_\phi$ directly. Concretely, we penalize the deviation of $\mathbf{W}_\phi^\top\mathbf{W}_\phi$ from the identity matrix:

$$\mathcal{L}_{\text{orth}} = \left\|\mathbf{W}_\phi^\top\mathbf{W}_\phi - \mathbf{I}\right\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. By enforcing orthonormality of the projection columns, this regularization structurally prevents feature redundancy, encouraging each projection direction to capture a distinct component of the representation space.

Covariance Regularization. Inspired by [2, 50] we penalize the off-diagonal entries of the channel covariance matrix of the projected representations $\tilde{\mathbf{z}}_0$. Concretely, letting $C(\tilde{\mathbf{z}}_0) \in \mathbb{R}^{d\times d}$ denote the normalized channel covariance matrix, we define:

$$\mathcal{L}_{\text{cov}}(\tilde{\mathbf{z}}_0) = \frac{1}{d}\sum_{i\neq j}[C(\tilde{\mathbf{z}}_0)]_{i,j}^2\,,$$

where d is the number of feature channels. This encourages the off-diagonal entries of $C(\tilde{\mathbf{z}}_0)$ to be close to zero, decorrelating the projected channels and preventing them from encoding redundant information.

3.4 Overall Training of CoReDi

The training is performed end-to-end by jointly optimizing the diffusion model parameters θ and the projection parameters ϕ as visualized in Fig. 3. The total training objective is:

$$\mathcal{L}(\theta,\phi) = \mathcal{L}_{\text{image}}(\theta,\phi) + \lambda_z\mathcal{L}_{\text{rep}}(\theta,\phi) + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}(\phi), \tag{5}$$

where $\mathcal{L}_{\text{image}}$ is the image flow-matching loss, $\mathcal{L}_{\text{rep}}$ is the representation flow-matching loss, and $\mathcal{L}_{\text{reg}}$ is a regularization term applied solely to the projection parameters ϕ. Specifically, $\mathcal{L}_{\text{reg}}$ can be any of the feature variance, orthogonality, or covariance regularizers, as described in Sec. 3.3: $\mathcal{L}_{\text{reg}} \in \{\mathcal{L}_{\text{var}}, \mathcal{L}_{\text{orth}}, \mathcal{L}_{\text{cov}}\}$. The hyperparameters λ_z and λ_{reg} control the relative contributions of the representation and regularization losses, respectively.

3.5 Coevolving Representations in Pixel Space

To extend `CoReDi` to pixel space, we build upon the encoder-decoder architecture of `DeCo` [29], in which a `DiT` encoder operates on downsampled images and a lightweight pixel decoder reconstructs full-resolution outputs. Given a noisy downsampled image $\hat{\mathbf{x}}_t$ and the noisy coevolving representation $\tilde{\mathbf{z}}_t$, the encoder processes both modalities jointly to produce the joint condition features $\mathbf{c}_{\text{joint}}$:

$$\mathbf{c}_{\text{joint}} = \text{Enc}_\theta(\hat{\mathbf{x}}_t, \tilde{\mathbf{z}}_t, t), \tag{6}$$

The full-resolution image velocity and representation velocity are then predicted by the pixel decoder and a lightweight linear projection head, respectively:

$$\mathbf{v}^x_{\text{pred}} = \text{Dec}_\theta(\mathbf{x}_t, \mathbf{c}_{\text{joint}}, t), \qquad \mathbf{v}^z_{\text{pred}} = \mathbf{W}_{\text{dec}}\,\mathbf{c}_{\text{joint}}. \tag{7}$$

This extension requires only minimal modifications to the DeCo architecture, enabling joint modeling of image and semantic representation tokens within a unified encoder-decoder framework.

Model	#Params	Iter.	FID↓
SiT-B/2	130M	400K	33.0
ReDi-B/2	130M	400K	21.4
CoReDi-B/2	130M	200K	24.7
CoReDi-B/2	130M	400K	16.4
SiT-XL/2	675M	7M	8.3
REPA-XL/2	675M	4M	5.9
ReDi-XL/2	675M	4M	3.3
CoReDi-XL/2	675M	2M	3.3

Table 1: Latent Diffusion Comparison without CFG. FID scores on ImageNet256 without Classifier-Free Guidance for SiT models of various sizes with REPA, ReDi and CoReDi.

Model	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Latent Diffusion Models						
LDM [35]	200	3.60	-	247.7	0.87	0.48
DiT-XL/2 [33]	1400	2.27	4.60	278.2	0.83	0.57
SiT-XL/2 [28]	1400	2.06	4.50	270.3	0.82	0.59
Leveraging Visual Representations						
REPA [49]	800	1.80	4.50	284.0	0.81	0.61
ReDi [22]	800	1.72	4.68	278.7	0.77	0.63
CoReDi	**400**	**1.58**	4.33	297.2	0.63	0.78

Table 2: Latent Diffusion Comparison with CFG. Quantitative evaluation on ImageNet256 with Classifier-Free Guidance. REPA, ReDi, and CoReDi all use SiT-XL/2 as the base model.

4 Experiments

Implementation Details. Training. For latent diffusion experiments, we follow the standard training setup of DiT [33] and SiT [28], training on ImageNet at 256×256 resolution with a batch size of 256. For pixel space diffusion, we experiment with DeCo-L/16 using a patch size of 16×16. For fair comparison with ReDi [22], we set $\lambda_z = 1$ and use a learnable projection with 8 output channels. For pixel space diffusion, we set $\lambda_z = 0.1$ and use a learnable projection with 16 output channels. The projection layer is initialized with a random orthogonal matrix. Unless noted otherwise, we use the Feature Variance regularization loss with $\lambda_{\texttt{reg}} = 1.0$. For the XL experiments, which are trained for more iterations, we optimize the projection using a cosine decay scheduler; see the appendix for additional details.

Sampling. For latent diffusion experiments, we employ the SDE Euler–Maruyama sampler. To ensure a fair comparison with ReDi the number of sampling steps is fixed at 250 across all experiments. For pixel diffusion experiments, we follow DeCo and use the Heun sampler and 50 sampling steps.

Evaluation. To benchmark generative performance, we report Frechet Inception Distance (FID) [16], sFID [30], Inception Score [37], Precision (Pre.) and Recall (Rec.) [23] using 50k samples and the ADM's TensorFlow evaluation suite [11].

4.1 Latent Space Diffusion

CoReDi improves the generative modeling performance. To demonstrate the effectiveness of our approach, we compare CoReDi against SiT, REPA,

VE	Method	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
DINOv2 [31]	ReDi	30.9	7.2	49.2	0.56	0.57
	CoReDi	24.7	6.7	57.4	0.60	0.61
MOCOv3 [10]	ReDi	38.2	7.6	37.4	0.54	0.58
	CoReDi	33.5	6.5	41.2	0.55	0.61
SigLIPv2 [43]	ReDi	36.2	7.4	41.5	0.55	0.59
	CoReDi	29.1	6.5	48.7	0.58	0.61
MAE [15]	ReDi	40.3	6.9	34.1	0.52	0.58
	CoReDi	37.1	6.6	36.8	0.52	0.62

Table 3: Representation Encoder Variation. Results with different Visual Encoders (VE). All models at 200K steps. CoReDi yields consistent performance improvements across different choices for the visual encoder used for joint training compared to fixed ReDi projection used in [22].

and ReDi across different model scales and evaluation settings in Tab. 1 and Tab. 2. Without classifier-free guidance, CoReDi consistently delivers better generative performance. For the B/2 architecture, CoReDi reaches an FID of 16.4 at 400K iterations, substantially outperforming ReDi (21.4 FID) and SiT (33.0 FID) at the same training budget. For the larger XL/2 architecture, CoReDi matches the best FID of 3.3 achieved by ReDi-XL/2, while requiring only 2M iterations instead of 4M, and clearly outperforms both REPA-XL/2 with 5.9 FID and the converged SiT-XL/2 baseline with 8.3 FID at 7M iterations.

With classifier-free guidance, CoReDi further establishes a stronger trade-off between generation quality and training cost, as shown in Tab. 2. CoReDi achieves an FID of 1.58 after only 400 epochs, improving over both REPA (1.80 at 800 epochs) and ReDi (1.72 at 800 epochs), while cutting the training steps by half.

Generalization Across Visual Encoders. To further validate the generality of our approach, we evaluate CoReDi against ReDi across four different visual encoders in Tab. 3. CoReDi consistently improves over the fixed PCA projection of ReDi across all encoders, demonstrating that the benefits of our adaptive representation learning are not specific to any particular choice of visual encoder. Notably, improvements are observed across all encoders, with the largest gains for DINOv2 (30.9 → 24.7 FID) and SigLIPv2 (36.2 → 29.1 FID), suggesting that the learned projection can extract more generative-relevant information for a variety of semantic features.

4.2 Pixel Space Diffusion

Improved pixel space generation. For pixel space diffusion, we evaluate CoReDi on top of DeCo-L/16 in Tab. 4. At 100K iterations, CoReDi-L/16 achieves

Model	Iter.	Param.	FID↓
DeCo-L/16*	100K	426M	46.0
DeCo-L/16	200K	426M	31.3
CoReDi-L/16	100K	426M	31.5
CoReDi-L/16	200K	426M	21.5

Table 4: Pixel Diffusion FID Comparisons. FID scores on ImageNet 256×256 without Classifier-Free Guidance for DeCo-L/16 and CoReDi-L/16. *Denotes our reproduction.

λ_z	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
0.04	34.4	9.6	43.0	0.50	0.63
0.08	32.4	9.1	45.6	0.51	0.62
0.10	31.5	8.9	47.0	0.52	0.63
0.14	33.5	10.2	45.5	0.50	0.62

Table 5: Pixel Space λ_z Ablation Results at 100K steps for pixel CoReDi-L/16 under different representation loss weights λ_z.

a comparable FID of 31.5 to DeCo-L/16 at 200K iterations, indicating a $\times 2$ acceleration in convergence. With further training, the performance continues to improve, reaching an FID of 21.5 at 200K iterations. These results demonstrate that coevolving representations provide consistent benefits beyond latent-based generation, yielding notable performance improvements in the pixel diffusion setting as well.

Representation Loss Weight in Pixel Space. In pixel space, images and semantic representations operate at very different dimensionalities, unlike the latent space setting where VAE latents and representations are of comparable scale. This discrepancy makes a naive transfer of the latent space hyperparameter λ_z suboptimal. Tab. 5 ablates the effect of λ_z in the pixel space setting, showing that larger values that are effective in latent space diffusion lead to degraded performance here. Performance peaks at $\lambda_z = 0.1$, achieving an FID of 31.5, which we adopt as our default for all pixel space experiments.

4.3 Analysis

Necessity of Batch Normalization and Stop-Gradient. Tab. 7 validates the necessity of both batch normalization and stop-gradient in CoReDi. Removing the stop-gradient leads to a significant performance drop, with FID deteriorating from 24.7 to 50.8, as the projection trivially minimizes the representation loss by collapsing the target rather than learning a meaningful representation space. Removing batch normalization causes complete training collapse, with FID reaching 223.9 and near-zero recall, confirming that the unbounded scale of the learned projection severely distorts the noise schedule and destabilizes training. These results confirm that both components are necessary for stable coevolution.

Reg.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
CoReDi	24.7	6.7	57.4	0.60	0.61
w/o SG	50.8	8.1	29.5	0.49	0.53
w/o BN	223.9	150.6	3.6	0.06	0.01

Table 7: Stabilization Methods. Results at 200K steps for latent CoReDi-B/2 without Stop-Gradient (SG) and without Batch Normalization (BN).

λ_{reg}	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
0.5	25.9	6.6	55.2	0.59	0.60
1.0	24.7	6.7	57.4	0.60	0.61
1.5	24.3	6.6	58.9	0.61	0.61

Table 8: Regularization Weight. Results for CoReDi-B/2 at 200K under different regularization weights λ_{reg} for Feature Variance Loss.

Effect of different Regularization Methods. In Tab. 6 we present the effect of the different regularization strategies on generative performance. Without any regularization, the learned projection suffers from feature collapse, yielding an FID of 37.2, worse than the static PCA projection of ReDi, confirming that naive joint optimization of the projection leads to degenerate solutions. This can also be qualitatively observed in Fig. 4 for Feature Variance Loss. All three regularization strategies successfully prevent collapse and recover strong performance: orthogonality regularization achieves an FID of 25.6, covariance regularization 25.9 FID, and Feature Variance regularization 24.7 FID. These results demonstrate that explicit regularization is a necessary ingredient for stable coevolution, and that Feature Variance regularization provides the best results among the strategies explored.

Reg.	FID↓	sFID↓	IS↑
ReDi	30.9	7.2	49.2
-	37.2	7.2	39.8
Ortho	25.6	6.8	57.0
Cov	25.9	7.0	54.8
VF	24.7	6.7	57.4

Table 6: Effect of Regularization. Results at 200K steps for CoReDi-B/2 under different regularization strategies: orthogonality (Ortho), covariance (Cov), feature variance (VF), and no regularization (-). The fixed PCA projection of ReDi [22] is included as a reference (gray).

Effect of Feature Variance Regularization Weight. In Tab. 8 we ablate the effect of the variance regularization weight λ_{reg}. All three values yield strong performance, demonstrating robustness to this hyperparameter. Performance improves slightly with larger weights, with $\lambda_{\text{reg}} = 1.5$ achieving the best FID of 24.3.

4.4 Coevolving Representations Develop Spatial Structure

Recent work has shown that the spatial structure of visual representations, rather than their global semantic quality, is the primary driver of generation performance in representation-guided diffusion models [40]. Motivated by this finding, we investigate whether the coevolving representations in CoReDi develop stronger spatial structure over the course of training. We measure this using three spatial self-similarity metrics proposed in [40]: LDS [18], which measures the contrast

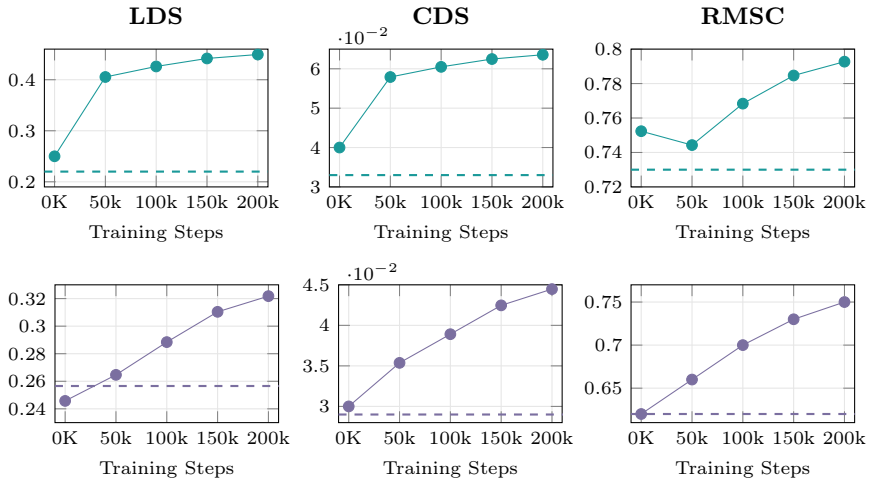

Fig. 5: Spatial structure of coevolving representations during training for `CoReDi` with `DINOv2` and `MOCOv3` as measured by `LDS`, `CDS`, and `RMSC`. All three metrics improve consistently as training progresses. Dashed horizontal lines indicate the fixed PCA projections used in `ReDi` [22].

between the average similarity of nearby versus distant patch pairs, `CDS` [40], which captures how quickly patch similarity decays with spatial distance, and `RMSC` [40], which measures the overall spatial diversity of patch features. (See Appendix for more details).

In Fig. 5, we illustrate the evolution of all three metrics throughout `CoReDi` training. We observe that all three improve consistently as training progresses, suggesting that the adaptive projection naturally evolves toward representations with stronger spatial organization. Furthermore, the learned projections achieve higher spatial structure scores than the fixed `PCA` projection used in `ReDi`, suggesting that the coevolving representation space captures richer spatial information than a static linear projection. This suggests a potential explanation for the generative improvements observed with `CoReDi`: the joint optimization of the projection with the generative objective encourages the learned representation space to develop spatial structure that is more beneficial for image synthesis.

5 Conclusion

We introduced `CoReDi`, a framework for joint image-feature diffusion in which the semantic representation space coevolves with the generative model during training. Unlike prior work that relies on fixed, predetermined representations to assist generation, `CoReDi` learns an adaptive projection of the representation space jointly with the diffusion objective, allowing the representation space to develop structure that is directly beneficial for image synthesis. Through systematic analysis, we identified three necessary ingredients for stable coevolution

— stop-gradient stabilization, batch normalization, and explicit regularization against feature collapse — and demonstrated empirically that all three are essential. We further showed that the coevolving representations develop stronger spatial structure over the course of training, providing a potential explanation for the observed generative improvements. Finally, we demonstrated that `CoReDi` extends naturally beyond `VAE` latent spaces to pixel-space diffusion, yielding consistent improvements across both settings. We hope that this work motivates further exploration of adaptive representation spaces as a tool for improving generative modeling.

Acknowledgements

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. Hardware resources were granted with the support of GRNET. Also, this work was partially conducted using EuroHPC resources (Project ID e-dev-2026d01-087) and HPC resources from GENCI-IDRIS (Grant AD011016639).

References

1. Albergo, M., Boffi, N.M., Vanden-Eijnden, E.: Stochastic interpolants: A unifying framework for flows and diffusions. Journal of Machine Learning Research **26**(209), 1–80 (2025) 5
2. Bardes, A., Ponce, J., LeCun, Y.: Vicreg: Variance-invariance-covariance regularization for self-supervised learning. arXiv preprint arXiv:2105.04906 (2021) 5, 8, 9
3. Bardes, A., Ponce, J., LeCun, Y.: Vicregl: Self-supervised learning of local visual features. Advances in Neural Information Processing Systems **35**, 8799–8810 (2022) 5
4. Black Forest Labs: FLUX.2: Analyzing and enhancing the latent space of FLUX – representation comparison (2025), https://bfl.ai/research/representation-comparison 5
5. Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9630–9640 (2021), https://api.semanticscholar.org/CorpusID:233444273 5
6. Chen, B., Bi, S., Tan, H., Zhang, H., Zhang, T., Li, Z., Xiong, Y., Zhang, J., Zhang, K.: Aligning visual foundation encoders to tokenizers for diffusion models. ICLR (2026) 5
7. Chen, S., Ge, C., Zhang, S., Sun, P., Luo, P.: Pixelflow: Pixel-space generative models with flow. arXiv preprint arXiv:2504.07963 (2025) 4
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PmLR (2020) 5
9. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021) 5

10. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9620–9629 (2021), https://api.semanticscholar.org/CorpusID:233024948 5, 11
11. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021) 2, 10, 21
12. Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: International conference on machine learning. pp. 3015–3024. PMLR (2021) 5
13. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024) 5
14. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020) 5
15. 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) 11
16. 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) 10, 21
17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 2
18. Huang, J., Kumar, R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 762–768 (1997), https://api.semanticscholar.org/CorpusID:9317935 13
19. Karypidis, E., Kakogeorgiou, I., Gidaris, S., Komodakis, N.: Dino-foresight: Looking into the future with dino. arXiv preprint arXiv:2412.11673 (2024) 5
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 21
21. Kouzelis, T., Kakogeorgiou, I., Gidaris, S., Komodakis, N.: EQ-VAE: Equivariance regularized latent space for improved generative image modeling. In: Forty-second International Conference on Machine Learning (2025), https://openreview.net/forum?id=UWhW5YYLo6 4
22. Kouzelis, T., Karypidis, E., Kakogeorgiou, I., Gidaris, S., Komodakis, N.: Boosting generative image modeling via joint image-feature synthesis. arXiv preprint arXiv:2504.16064 (2025) 2, 5, 6, 10, 11, 13, 14, 19
23. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in neural information processing systems **32** (2019) 10, 22
24. Leng, X., Singh, J., Hou, Y., Xing, Z., Xie, S., Zheng, L.: Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. arXiv preprint arXiv:2504.10483 (2025) 2, 5
25. Li, T., He, K.: Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720 (2025) 4
26. Li, T., Sun, Q., Fan, L., He, K.: Fractal generative models. arXiv preprint arXiv:2502.17437 (2025) 4

27. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022) 5
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) 4, 10, 19, 20
29. Ma, Z., Wei, L., Wang, S., Zhang, S., Tian, Q.: Deco: Frequency-decoupled pixel diffusion for end-to-end image generation. arXiv preprint arXiv:2511.19365 (2025) 3, 4, 5, 9, 20
30. Nash, C., Menick, J., Dieleman, S., Battaglia, P.W.: Generating images with sparse representations. arXiv preprint arXiv:2103.03841 (2021) 10, 22
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) 5, 11
32. Pan, Y., Feng, R., Dai, Q., Wang, Y., Lin, W., Guo, M., Luo, C., Zheng, N.: Semantics lead the way: Harmonizing semantic and texture modeling with asynchronous latent diffusion. arXiv preprint arXiv:2512.04926 (2025) 2, 5
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) 4, 10
34. Petsangourakis, G., Sgouropoulos, C., Psomas, B., Giannakopoulos, T., Sfikas, G., Kakogeorgiou, I.: Reglue your latents with global and local semantics for entangled diffusion. arXiv preprint arXiv:2512.16636 (2025) 2, 5
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) 2, 4, 10
36. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 4
37. 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) 10, 22
38. Shi, M., Wang, H., Zheng, W., Yuan, Z., Wu, X., Wang, X., Wan, P., Zhou, J., Lu, J.: Latent diffusion model without variational autoencoder. arXiv preprint arXiv:2510.15301 (2025) 2, 5
39. Shi, M., Wang, H., Zheng, W., Yuan, Z., Wu, X., Wang, X., Wan, P., Zhou, J., Lu, J.: Latent diffusion model without variational autoencoder. ICLR **abs/2510.15301** (2026), https://api.semanticscholar.org/CorpusID:282203316 5
40. Singh, J., Leng, X., Wu, Z., Zheng, L., Zhang, R., Shechtman, E., Xie, S.: What matters for representation alignment: Global information or spatial structure? arXiv preprint arXiv:2512.10794 (2025) 2, 5, 13, 14, 22
41. Teng, J., Zheng, W., Ding, M., Hong, W., Wangni, J., Yang, Z., Tang, J.: Relay diffusion: Unifying diffusion process across resolutions for image synthesis. arXiv preprint arXiv:2309.03350 (2023) 4
42. Tong, S., Zheng, B., Wang, Z., Tang, B., Ma, N., Brown, E., Yang, J., Fergus, R., LeCun, Y., Xie, S.: Scaling text-to-image diffusion transformers with representation autoencoders. arXiv preprint arXiv:2601.16208 (2026) 2, 5

43. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025) 5, 11
44. Venkataramanan, S., Pariza, V., Salehi, M., Knobel, L., Gidaris, S., Ramzi, E., Bursuc, A., Asano, Y.M.: Franca: Nested matryoshka clustering for scalable visual representation learning. arXiv preprint arXiv:2507.14137 (2025) 5
45. Wang, S., Gao, Z., Zhu, C., Huang, W., Wang, L.: Pixnerd: Pixel neural field diffusion. arXiv preprint arXiv:2507.23268 (2025) 4
46. Wang, S., Tian, Z., Huang, W., Wang, L.: Ddt: Decoupled diffusion transformer. arXiv preprint arXiv:2504.05741 (2025) 4
47. Wu, G., Zhang, S., Shi, R., Gao, S., Chen, Z., Wang, L., Chen, Z., Gao, H., Tang, Y., Yang, J., et al.: Representation entanglement for generation: Training diffusion transformers is much easier than you think. arXiv preprint arXiv:2507.01467 (2025) 2, 5
48. 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) 2, 4
49. 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. In: International Conference on Learning Representations (2025) 2, 5, 10, 19
50. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: International conference on machine learning. pp. 12310–12320. PMLR (2021) 5, 9
51. Zhai, S., Zhang, R., Nakkiran, P., Berthelot, D., Gu, J., Zheng, H., Chen, T., Bautista, M.A., Jaitly, N., Susskind, J.: Normalizing flows are capable generative models. arXiv preprint arXiv:2412.06329 (2024) 4
52. Zheng, B., Ma, N., Tong, S., Xie, S.: Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690 (2025) 2, 5
53. Zheng, K., Chen, Y., Mao, H., Liu, M.Y., Zhu, J., Zhang, Q.: Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. arXiv preprint arXiv:2409.02908 (2024) 4

Appendix

A Additional Results and Ablations

Model	#Iters.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-XL/2 [28]	7M	8.3	6.3	131.7	0.68	0.67
REPA-XL/2 [49]	50K	52.3	31.2	24.3	0.45	0.53
ReDi-XL/2 [22]	50K	56.1	18.9	23.8	0.44	0.47
CoReDi-XL/2	50K	40.9	9.74	32.1	0.53	0.56
REPA-XL/2 [49]	100K	19.4	6.1	67.4	0.64	0.61
ReDi-XL/2 [22]	100K	23.1	5.9	61.5	0.64	0.57
CoReDi-XL/2	100K	19.0	5.6	69.3	0.65	0.59
REPA-XL/2 [49]	200K	11.1	5.0	100.4	0.69	0.64
ReDi-XL/2 [22]	200K	12.6	5.7	97.3	0.69	0.61
CoReDi-XL/2	200K	9.2	4.7	110.0	0.71	0.62
REPA-XL/2 [49]	400K	7.9	5.1	122.6	0.70	0.65
ReDi-XL/2 [22]	400K	7.5	5.1	129.5	0.72	0.62
CoReDi-XL/2	400K	6.1	4.6	136.1	0.73	0.64
REPA-XL/2 [49]	4M	5.9	5.7	157.8	0.70	0.69
ReDi-XL/2 [22]	4M	3.3	4.8	188.9	0.74	0.68
CoReDi-XL/2	2M	3.3	4.4	176.8	0.74	0.66

Table 9: Detailed evaluation for SiT-XL/2, CoReDi-XL/2, ReDi-XL/2, and REPA-XL/2. All results are reported without Classifier-Free Guidance.

Detailed Quantitative Comparison. In Table 9, we present detailed results for CoReDi alongside REPA and ReDi. We observe that CoReDi converges significantly faster than both baselines and matches ReDi's converged generative performance at 4M iterations with only 2M iterations.

VAE-only Classifier-Free Guidance. Following ReDi [22], we apply Classifier-Free Guidance exclusively to the VAE latents rather than across both the image latents and the features, as this strategy consistently yields superior generation quality and greater robustness to CFG weight variations (see Section 4.4 in [22] for more details). In our ablation as presented in Fig. 6, we find that a CFG weight of 1.8 achieves optimal performance.

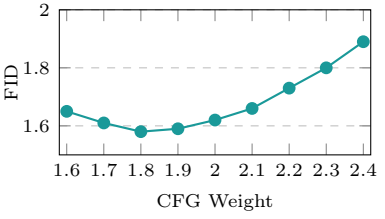

Fig. 6: FID score as a function of CFG weight.

Spatial Structure of Coevolving Representations in Pixel Diffusion. Fig. 7 shows the evolution of spatial structure metrics throughout CoReDi training in pixel space. All three metrics — LDS, CDS, and RMSC — improve consistently as training progresses. This validates our observation that coevolving representations develop increasingly structured spatial organization during training, and further demonstrates that this phenomenon is not specific to latent space diffusion but holds in the pixel space setting as well.

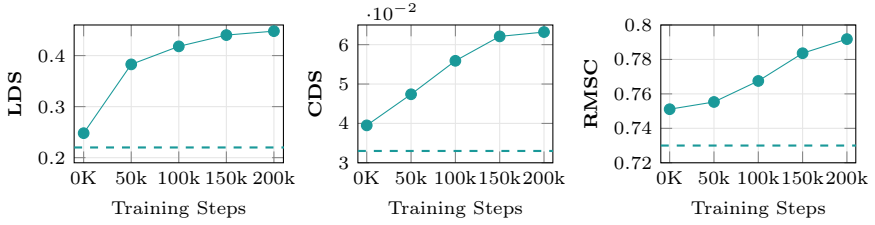

Fig. 7: Spatial structure of coevolving representations in pixel space for CoReDi-L/16 with DINOv2 as measured by LDS, CDS, and RMSC. All three metrics improve consistently as training progresses. Dashed horizontal lines indicate the fixed PCA projections.

B Additional Implementation Details

B.1 Architecture settings

Latent Space Diffusion. We follow the SiT configurations from [28]. SiT-B/2 (130M parameters) uses 12 transformer blocks with embedding dimension 768 and 12 attention heads. SiT-XL/2 (675M parameters) uses 28 blocks with embedding dimension 1152 and 16 heads. In all latent diffusion experiments, images are encoded with SD-VAE-FT-EMA, and we use a 2×2 patch size.

We observe that applying a cosine decay schedule to the projection yields more stable optimization over a longer training horizon and results in better generative performance. In particular, for the XL experiments, we use a cosine decay schedule that reduces the projection learning rate to 0 by 400k iterations. We emphasize that this schedule is applied only to the learnable projection and not to the backbone DiT, which is trained with the standard constant learning rate $1e-4$.

Pixel Space Diffusion. We follow the DeCo configuration from [29]. DeCo-L/16 uses 22 transformer blocks in the encoder and 3 MLP blocks in the pixel decoder, with embedding dimension 1024 and 16 attention heads. In pixel-space experiments, we set $\lambda_z = 0.1$ and use a learnable projection to 16 channels.

	Pixel	Latent	
CoReDi			
λ_z	0.1	1	1
λ_{reg}	1	1	1
Proj. Channels	16	8	8
Architecture			
Backbone	`DeCo-L`	`SiT-B`	`SiT-XL`
DiT Depth	22	12	28
Hidden Dim.	1024	768	1152
Heads	16	12	16
Params	426M	130M	675M
Decoder Depth	3	-	-
Decoder Hidden Dim.	32	-	-
Patch Size	16	2	2
Optimization			
Optimizer	AdamW	AdamW	AdamW
Batch Size	256	256	256
Learning Rate	1e-4	1e-4	1e-4
LR Schedule	constant	constant	constant
LR Schedule Proj.	constant	constant	cos. decay
Weight Decay	0	0	0
EMA Decay	0.9999	0.9999	0.9999
Time Sampler	logit normal	uniform	uniform
Sampling			
Steps	50	250	250
CFG Scale	-	-	1.8

Table 10: Configurations of pixel space and latent space experiments.

B.2 Optimization Settings

We optimize all models using AdamW [20] with a constant learning rate of 1×10^{-4}, momentum parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, and a batch size of 256. To accelerate training for latent diffusion we pre-compute the image latents. Full optimization details are provided in Table 10.

C Evaluation Metrics

Generative Performance Evaluation Metrics. We follow the evaluation setup of ADM [11], generating 50K samples and computing metrics using the official TensorFlow evaluation toolkit [11]. All evaluations are conducted on NVIDIA A100 GPUs. We report the following metrics:

- **FID** [16] measures the distance between the feature distributions of real and generated images, computed using `Inception-v3` under a multivariate Gaussian assumption.

- **sFID** [30] computes FID using spatial feature maps from intermediate `Inception-v3` layers, better capturing the spatial structure of generated images.
- **IS** [37] evaluates generated images using `Inception-v3`, assigning higher scores to outputs that are both classifiable with high confidence and diverse across categories.
- **Precision & Recall** [23] measure realism and diversity in feature space. Precision reflects the fraction of generated images that appear realistic, while recall measures coverage of the real data distribution.

C.1 Spatial Structure Evaluation Metrics

In this section, we briefly describe each spatial self-similarity metric used to analyze the coevolving representations in `CoReDi`, following the definitions introduced in [40].

Local vs. Distant Similarity (LDS). LDS measures the average similarity contrast between spatially close and distant patch pairs:

$$\mathrm{LDS}(\mathbf{X}) = \mathbb{E}[K_{\mathbf{X}}(t,t') \mid d(t,t') < r_{\mathrm{near}}] - \mathbb{E}[K_{\mathbf{X}}(t,t') \mid d(t,t') \geq r_{\mathrm{far}}],$$

where $K_{\mathbf{X}}$ is the cosine similarity between patch tokens and $d(\cdot,\cdot)$ is the Manhattan distance. Larger values indicate stronger spatial organization, where nearby patches are more similar to each other than distant ones.

Correlation Decay Slope (CDS). CDS measures how quickly patch similarity decays with spatial distance. Given the spatial correlogram $g_{\mathbf{X}}(\delta) = \mathbb{E}[K_{\mathbf{X}}(t,t') \mid d(t,t') = \delta]$, fit a least-squares line $\hat{g}_{\mathbf{X}}(\delta) \approx \alpha + \beta\delta$ and define:

$$\mathrm{CDS}(\mathbf{X}) = -\hat{\beta},$$

where $\hat{\beta}$ is the fitted slope. Larger values indicate faster similarity decay with distance, reflecting stronger spatial organization.

RMS Spatial Contrast (RMSC). RMSC measures the spatial diversity of patch token representations. Given normalized patch features $\hat{\mathbf{x}}_t = \mathbf{x}_t / \|\mathbf{x}_t\|_2$, it is defined as:

$$\mathrm{RMSC}(\mathbf{X}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{x}}_t - \bar{\mathbf{x}}\|_2^2},$$

where $\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{x}}_t$ is the mean normalized feature. Higher values indicate greater spatial diversity, reflecting preserved spatial structure, while lower values indicate more uniform, spatially uninformative representations.

D Additional Qualitative Results

We provide qualitative results of both generated images and visual representations in Fig. 8. Further, in Fig. 9, we visually compare our learned representations with static PCA for all visual encoders examined in the paper.

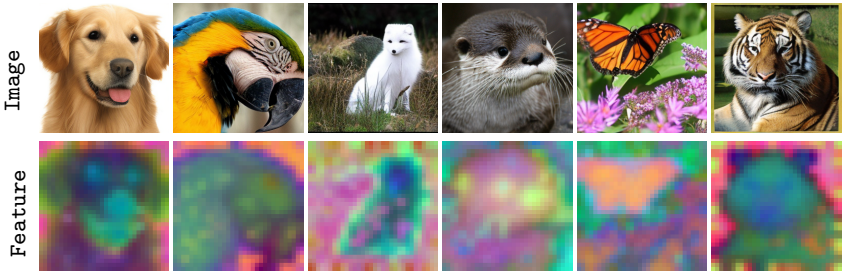

Fig. 8: Selected samples from our `CoReDi-XL/2` trained for 1M steps on ImageNet 256×256. Images and visual representations are jointly generated by our model. We use Classifier-Free Guidance with $w = 4.0$.

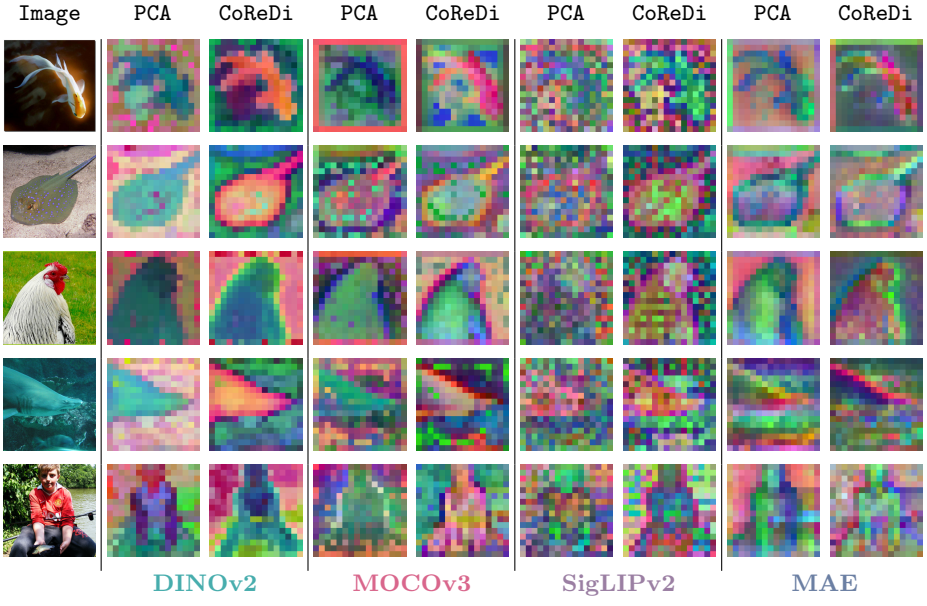

Fig. 9: Qualitative comparison of feature visualizations. For each image, we show PCA visualization of DINOv2, MOCOv3, SigLIPv2 and MAE features. For each feature, we visualize PCA vs `CoReDi`'s learned projection.