RAC: RECTIFIED FLOW AUTO CODER

Anonymous authors
Paper under double-blind review

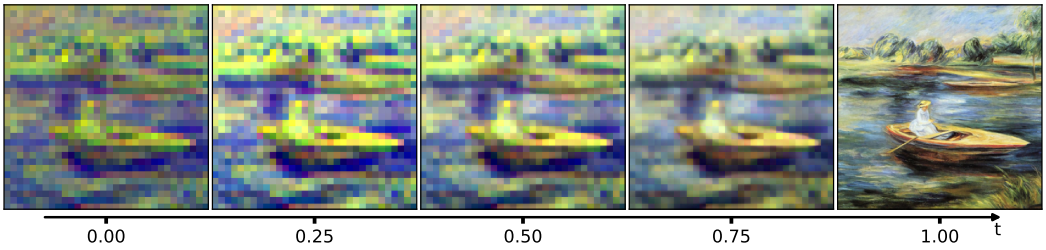


Figure 1: **The trajectory demonstration of RAC:** Make the reconstruction task a condition generation task; Make the decoder the encoder; Make the single-step decoding and encoding a multi-step decoding and encoding.

ABSTRACT

In this paper, we propose a **R**ectified Flow **A**uto **C**oder (**RAC**) inspired by Rectified Flow to replace the traditional VAE: **1.** It achieves multi-step decoding by applying the decoder to flow timesteps. Its decoding path is straight and correctable, enabling step-by-step refinement. **2.** The model inherently supports bidirectional inference, where the decoder serves as the encoder through time reversal (hence *Coder* rather than encoder or decoder), reducing parameter count by nearly 41%. **3.** This generative decoding method improves generation quality since the model can correct latent variables along the path, partially addressing the reconstruction–generation gap. Experiments show that RAC achieves a Pareto improvement over SOTA VAEs, where even a 10× parameter-reduced decoder exceeds full-scale VAE performance in both reconstruction and generation quality, validating the effectiveness of our approach.

1 INTRODUCTION

In the field of generative models, over the past many years, there has been a problem of inconsistent generation and reconstruction results. For instance, when using the same generation method and the same VAE, the generation results are often inferior to the reconstruction results: **(1)** Some studies consider it to be a problem of representation Zheng et al. (2026); Wang et al. (2026), and attempt to solve it by optimizing the latent space; **(2)** Some studies consider it to be a problem of training strategy Leng et al. (2025), and use the generation backbone and VAE for joint training optimization; **(3)** Some studies consider it to be a problem of the method Zhuang et al. (2025), and propose a native unified method to try to solve it, etc.

Under the same conditions of using VAE Kingma & Welling (2013), one very obvious and unique variable that distinguishes generation from reconstruction is - the variables used during generation are provided by the generation framework (such as Unet, DiT, etc.). Therefore, we hypothesize that part of the reason is that the latent variables output by the generation method are problematic. **It easily misses the uneven manifold space of traditional VAE**, resulting in unstable generation effects and being less accurate than the encoder provided by VAE itself[1].

[1]From another perspective, Text2Img (prediction, information increase) is not as precise as Img2Latent (encoding compression, information reduction), which can also serve as a reason for the disparity in the generation and reconstruction capabilities.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

So, in the case of minimal modifications, the method to enhance the decoder's performance during generation is to incorporate the decoder into the generation process. As shown in Fig. 1, we introduce the concepts of time steps and velocity fields into the decoder, **enabling it to gradually calibrate the input variables or the behavior during encoding**, and theoretically, it inherently supports bidirectional transformation. The decoder of traditional VAE is an important part of the generation process. Generation frameworks (such as Unet Ronneberger et al. (2015), DiT Peebles & Xie (2023), etc.) can perform multi-step reasoning, but the end VAE is actually a one-step generation. This is actually not very reasonable: Consider a navigator who, despite having a correctable route, is forced to teleport directly to the destination in one step — losing all opportunity to adjust course along the way.

Specifically, we take a unified view of continuous-time generation and representation learning, and elevate the classic VAE decoder from a pointwise mapping to a *path-level* generative process: a time-conditioned velocity field defines an integrable flow from latent variables to the image space, and explicitly models a correctable, multi-step decoding trajectory. This design provides a structured generation path while making the same model naturally act as an encoder under time reversal, enabling parameter sharing and bidirectional consistency. For training, we center on path consistency, latent alignment, and reconstruction constraints, yielding a unified optimization framework that balances invertibility and generation quality. Our contributions are threefold:

- We propose RAC, which generalizes VAE decoding from a *single-step map* to a *continuous-time, integrable path*, establishing a **unified flow-based autoencoding paradigm** for generation and representation.
- We design a **structured bidirectional mechanism** where the same velocity-field model performs decoding and encoding in forward and reverse time, achieving parameter sharing, path reversibility, and generation–reconstruction consistency.
- We present a stable training objective and implementation recipe (**path consistency + latent alignment + reconstruction constraints**), delivering strong reconstruction and generation performance at comparable model scale.

2 METHODOLOGY

Figure 2 illustrates the RAC pipeline. We replace the single-step VAE decoder with a continuous-time velocity field $\mathbf{v}_\theta$, which integrates a full-resolution *state* tensor $\mathbf{s} \in \mathbb{R}^{C_s \times H \times W}$ from a latent-derived initialization to the target image; time-reversal of the same model performs encoding. To prevent latent space collapse, we freeze the VAE encoder and train only the decoder under a joint objective covering reconstruction, path consistency, and latent alignment.

2.1 OVERVIEW AND GOAL

The core idea is to replace the single-step VAE decoder with a *time-conditioned velocity field* $\mathbf{v}_\theta(\mathbf{s}, t)$ that integrates a *state* $\mathbf{s}$ from a latent-derived initialization $\mathbf{s}_0$ to a target image state $\mathbf{s}^*$; running the same field in reverse serves as encoding. This formulation targets the fundamental mismatch in VAE-style models: reconstruction uses a data-conditioned encoder, while generation relies on latents that may not respect the decoder's learned manifold. By making decoding a multi-step, correctable process, generation can actively refine latent variables along the path rather than committing to a one-shot projection.

Let $\mathbf{x} \in [0,1]^{3\times H\times W}$ be an image and $\tilde{\mathbf{x}} = 2\mathbf{x} - 1 \in [-1,1]^{3\times H\times W}$ its normalized version. We define a state tensor by padding channels:

$$\mathbf{s} = \mathrm{pad}(\tilde{\mathbf{x}}) \in \mathbb{R}^{C_s \times H \times W}, \quad C_s \geq 3. \tag{1}$$

A KL-VAE teacher provides a latent target $\mathbf{z}_T \in \mathbb{R}^{C\times h\times w}$. We learn a continuous-time flow from latent state to image state and reuse the same flow in reverse for encoding. This gives us a single model that *(i)* decodes with step-wise correction, *(ii)* supports bidirectional inference, and *(iii)* aligns generation and reconstruction within one mechanism. This formulation is intentionally generic: any KL-regularized VAE can serve as the teacher, which makes the method broadly compatible with existing pipelines.

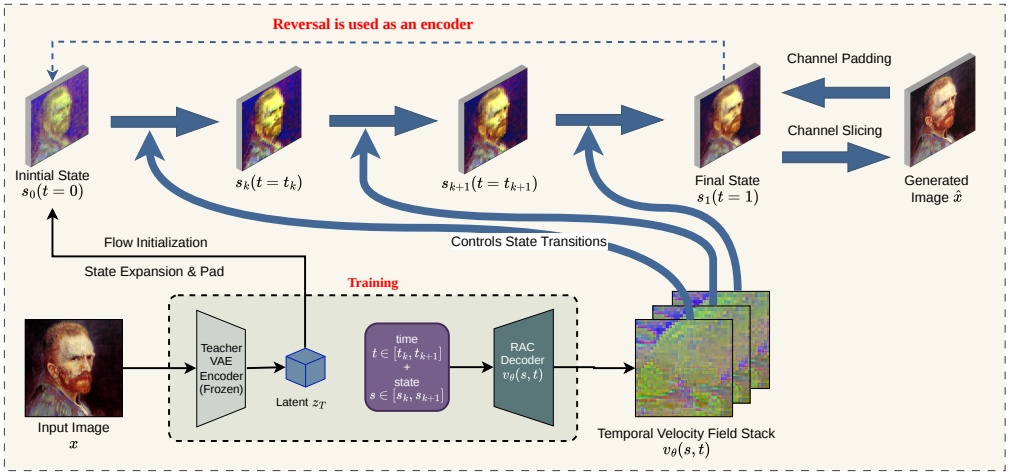


Figure 2: **Method Overview.** **(i) Training.** To prevent latent space collapse, we freeze the VAE encoder and train only the RAC decoder; reverse-time inference then serves as encoding. **(ii) State Construction.** Extra channels beyond RGB are padded with 0.5, keeping the velocity field shape constant and ensuring bidirectional consistency. **(iii) RAC Input.** RAC takes time t and the current state as input, driving the transition from latent initialization to the target image.

2.2 Time-Conditioned Rectified Flow Decoder

We model the decoder as a time-conditioned velocity field:

$$\frac{d\mathbf{s}(t)}{dt} = \mathbf{v}_\theta(\mathbf{s}(t), t), \quad t \in [0, 1]. \tag{2}$$

The velocity field is predicted by a lightweight VAE backbone Rombach et al. (2022). To stabilize and structure the flow, the network uses the current state, an explicit time channel, and optional relative positional encoding to return a full-resolution velocity field: $\mathbf{v}_\theta(\mathbf{s}, t) = f_\theta\Big(\text{down}(\mathbf{s}),\ t\mathbf{1},\ \text{pos}(\mathbf{s})\Big) \in \mathbb{R}^{C_s \times H \times W}$.

We build a latent state by padding the teacher latent and spatially expanding it:

$$\mathbf{s}_0 = \text{expand}\big(\text{pad}(\mathbf{z}_T)\big) \in \mathbb{R}^{C_s \times H \times W}. \tag{3}$$

Decoding is then performed by integrating the flow from $t = 0$ to $t = 1$:

$$\mathbf{s}_K = \mathbf{s}_0 + \int_0^1 \mathbf{v}_\theta(\mathbf{s}(t), t)\, dt, \tag{4}$$

and the predicted RGB image is the projection $\hat{\mathbf{x}} = \Pi(\mathbf{s}_K)$ onto the first three channels. Crucially, the *same* model yields an encoder by reversing time:

$$\mathbf{s}_0 = \mathbf{s}_K - \int_0^1 \mathbf{v}_\theta(\mathbf{s}(t), t)\, dt. \tag{5}$$

In practice we use K Euler steps with optional random time grids to encourage robustness:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \Delta t_k\, \mathbf{v}_\theta(\mathbf{s}_k, t_k), \quad k = 0, \ldots, K-1, \tag{6}$$

and inject small Gaussian noise per step to prevent brittle dynamics. This step-wise integration explicitly exposes intermediate states, enabling both diagnostic visualization and targeted regularization of the decoding path.

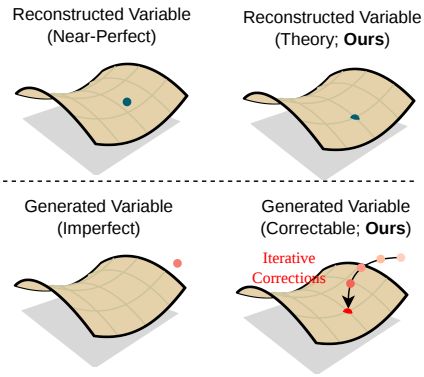


Figure 3: **Reconstruction is Condition Generation:** The previous reconstructions were more accurate because they could relatively approach the manifold. The previous generations predicted variables that were often some distance away from the manifold. This is part of the reason for the past differences in the performance of generation and reconstruction. However, our method theoretically aims for perfect reconstruction, and the multi-step decoding can correct the potential variables provided by Unet/DiT. Therefore, both reconstruction and generation are significantly superior to traditional VAEs.

As shown in Figure 3, RAC first learns image-state reconstruction, then organizes the latent space through generation, and finally refines decoding through multi-step calibration. This training objective directly bridges generation and reconstruction.

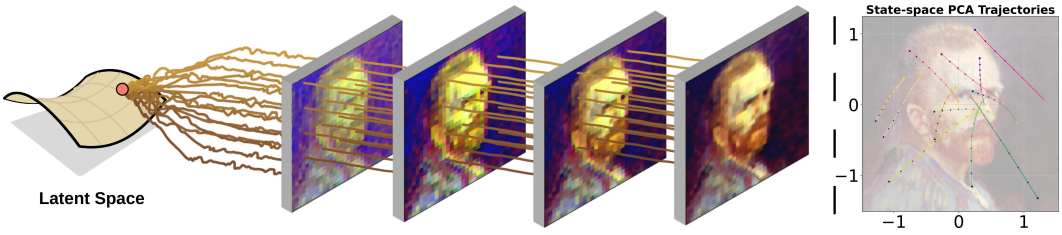


Figure 4: **Conceptual and empirical views of RAC generation trajectories.** Left: a conceptual illustration of trajectory-based generation from latent space to image space. Right: sampled state trajectories projected into a 2D PCA space.

2.3 State Construction and Transition

To bridge the resolution gap between latent space and image space, we introduce explicit *state construction* operators. The KL-VAE latent is padded and spatially expanded into a full-resolution latent state, $\mathbf{s}_0 = \mathrm{expand}\big(\mathrm{pad}(\mathbf{z}_T)\big) \in \mathbb{R}^{C_s \times H \times W}$. The target image state is defined by padding the normalized image: $\mathbf{s}^* = \mathrm{pad}(\tilde{\mathbf{x}})$. The transition $\mathbf{s}_0 \to \mathbf{s}^*$ is performed by integrating the rectified flow, producing a trajectory $\{\mathbf{s}_k\}_{k=0}^{K}$:

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \Delta t_k \, \mathbf{v}_\theta(\mathbf{s}_k, t_k), \quad k = 0, \ldots, K-1. \tag{7}$$

The inverse transition is obtained by time reversal using the same velocity field, enabling a single-model encoder–decoder pair. This explicit state transition view makes the generation path interpretable and allows step-wise correction in both directions, which is particularly important when the generated latent deviates from the encoder manifold. The encoded full-resolution state is downsampled back to the teacher latent size via a lightweight learned projector or average pooling. When a teacher latent channel count differs from C_s, a channel projection maps it to the RAC state width before spatial expansion.

As shown in Fig. 4, RAC generation can be viewed as a trajectory-based process rather than a single-step mapping. The left panel provides an intuitive illustration of this perspective, where a latent representation evolves through a sequence of intermediate image states. The right panel offers an empirical counterpart by projecting sampled state trajectories into a 2D PCA space. These projected paths suggest that RAC follows structured and coordinated routes in its shared state space during generation. For more engineering details, please refer to Section B.

2.4 Training Objectives

Our losses are designed to enforce three properties: *accurate reconstruction, rectified paths, and latent consistency*. Let $\mathbf{s}^* = \mathrm{pad}(\tilde{\mathbf{x}})$ be the target image state and $\{\mathbf{s}_k\}_{k=0}^{K}$ the decoded trajectory. All squared norms below denote mean squared error averaged over batch and tensor elements. We define a standard reconstruction loss: $\mathcal{L}_{\text{recon}} = \big\|\mathbf{s}_K - \mathbf{s}^*\big\|_2^2$. To explicitly encourage a *uniform, correctable* path, we penalize deviation from linear interpolation:

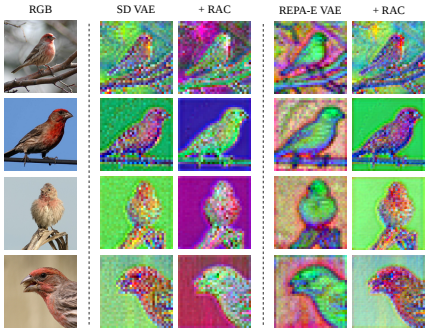
$$\mathcal{L}_{\text{path}}^{\text{dec}} = \frac{1}{K-1} \sum_{k=1}^{K-1} \Big\| \mathbf{s}_k - \mathbf{s}_0 - t_k(\mathbf{s}^* - \mathbf{s}_0) \Big\|_2^2. \tag{8}$$

For encoding, we reverse the flow to obtain $\hat{\mathbf{s}}_0$, downsample it to latent $\hat{\mathbf{z}}$, and align it with the teacher latent: $\mathcal{L}_{\text{latent}} = \|\hat{\mathbf{z}} - \mathbf{z}_T\|_2^2$. We further require that the teacher decoder applied to $\hat{\mathbf{z}}$ matches the input image: $\mathcal{L}_{\text{pixel}} = \|\mathrm{Dec}_T(\hat{\mathbf{z}}) - \mathbf{x}\|_2^2$. Finally, we enforce round-trip consistency so that encoding followed by decoding returns to the same image state: $\mathcal{L}_{\text{rt}} = \big\|\mathrm{Flow}_\theta(\mathrm{Flow}_\theta^{-1}(\mathbf{s}^*)) - \mathbf{s}^*\big\|_2^2$.

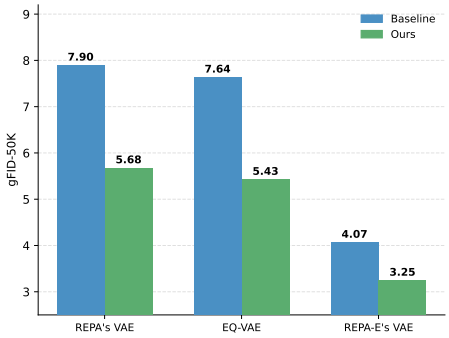
The final objective is:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{path}}\mathcal{L}_{\text{path}}^{\text{dec}} + \lambda_{\text{latent}}\mathcal{L}_{\text{latent}} + \lambda_{\text{pixel}}\mathcal{L}_{\text{pixel}} + \lambda_{\text{rt}}\mathcal{L}_{\text{rt}} \tag{9}$$

These objectives jointly enforce accurate endpoints, rectified intermediate paths, and latent alignment, which is crucial for maintaining consistency between reconstruction and generation.



(a) PCA Analysis on VAE Latent Space Structure. RAC yields cleaner, more organized latent structure.



(b) Performance Improvements with Baseline (400K Steps)

Figure 5: **The State Space of RAC is a More Organized Representation Space.** (a) We used PCA to conduct a visual analysis of the state space that we encoded. This state space represents the state before the final output of the encoder. We can clearly observe that the training of RAC has made the original Latent representation more orderly and clean. (b) Comparison of gFID-50K on ImageNet 256×256 Deng et al. (2009) across different VAE configurations. We report results for the baseline methods (REPA Yu et al. (2024), REPA-E Leng et al. (2025); Rombach et al. (2022) and EQ-VAE Kouzelis et al. (2025)), comparing each against our approach trained under identical settings. Lower is better. RAC consistently outperforms the corresponding baselines across all VAE settings.

Table 1: **VAE architecture and decoder scale on ImageNet-1K.** Left: generation quality across VAE backbones and training configurations. Right: reconstruction quality (rFID) and computation (GFLOPs) across RAC decoder scales. RAC consistently improves quality while reducing parameters or computation. Default settings are highlighted in gray. See App. C.3 for generation.

Autoencoder	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑	ΔParams
SD-VAE Rombach et al. (2022)	24.1	6.25	55.7	0.62	0.60	0.0%
+REPA-E Leng et al. (2025)	16.3	5.69	75.0	0.68	0.60	0.0%
+RAC (Ours)	**14.7**	**5.51**	**78.4**	**0.70**	**0.61**	**-40.8%**
IN-VAE Rombach et al. (2022); Leng et al. (2025)	22.7	5.47	56.0	0.62	0.62	0.0%
+REPA-E Leng et al. (2025)	12.7	5.57	84.0	0.69	0.62	0.0%
+RAC (Ours)	**11.2**	**5.21**	**87.5**	**0.71**	**0.63**	**-40.7%**
VA-VAE Yao et al. (2025)	12.8	6.47	83.8	0.71	0.58	0.0%
+REPA-E Leng et al. (2025)	11.1	5.31	88.8	0.72	0.61	0.0%
+RAC (Ours)	**9.8**	**5.08**	**91.4**	**0.73**	**0.62**	**-40.7%**

Decoder	rFID	GFLOPs
SD-VAE-0.1x	0.44	94.3
SD-VAE-0.2x	0.39	152.4
SD-VAE-0.3x	0.35	254.5
SD-VAE	0.62	310.4

Scalable Decoders: improve rFID while remaining much more efficient than VAEs. 0.1x indicates that the parameter being used is one-tenth of the SD.

3 Experiments

In this section, we present a comprehensive evaluation of RAC, including reconstruction quality comparison (§3.1), generation quality comparison (§3.2), bidirectional consistency and parameter efficiency (§3.3), generation–reconstruction gap analysis (§3.4), ablation study (§3.5), and visualization evaluation (§3.5).

Setup. All the datasets used in our experiments are the same as those of the comparison models. The detailed experimental settings will be explained in the corresponding sections.

3.1 Reconstruction Quality

Reconstruction Quality Comparison. As shown in Figure 5(a), RAC-induced latent representations exhibit significantly cleaner, more coherent structures than the standard SD-VAE, which stem from the path consistency and latent alignment objectives jointly regularizing the state space during training. This structural regularity not only benefits generation—as reflected by the consistent gFID-50K improvements across all VAE configurations in Figure 5(b)—but also yields more faithful image reconstruction, since a well-organized state space provides smoother and more stable ODE trajectories during the reverse decoding process. Quantitatively, RAC achieves competitive reconstruction quality while improving generative performance, demonstrating that the two objectives are complementary rather than conflicting within our flow-based autoencoding framework.

Table 1 (left) presents a comprehensive comparison of generation quality across three VAE backbones—SD-VAE, IN-VAE, and VA-VAE—each evaluated under three training configurations: the vanilla baseline, +REPA-E, and our proposed +RAC. Across all metrics including gFID, sFID, IS, Precision, and Recall, RAC consistently achieves the best performance within each VAE group.



Figure 6: **Qualitative Results on Imagenet 256 $\times$ 256** using RAC. It can achieve significant results in the state reconstruction mode with just 30k steps.

For SD-VAE, RAC reduces gFID from 24.1 (baseline) to 14.7, outperforming the +REPA-E variant at 16.3, while also improving IS from 75.0 to 78.4; a similar trend holds for IN-VAE. For the strongest backbone VA-VAE, RAC further pushes gFID down to 9.8 from 11.1, with sFID improving from 5.31 to 5.08 and IS reaching 91.4. Finally, we presented the parameter changes of our RAC compared to the original VAE. It can be observed that the parameters have generally decreased by approximately 41%. These gains across all three VAE configurations demonstrate that RAC provides a complementary improvement to any pretrained VAE backbone, reinforcing the effectiveness of our flow-based autoencoding framework as a plug-in enhancement for latent training.

Table 1 (right) studies the effects of decoder scale on reconstruction quality (rFID) and computational efficiency (GFLOPs) in RAC. We evaluate three scaled variants—SD-VAE-0.1x, SD-VAE-0.2x, and SD-VAE-0.3x—whose factors denote parameter fractions relative to the full SD-VAE decoder used as reference. Despite retaining only a fraction of its parameters, all three achieve substantially better rFID. This counterintuitive but important result shows that, within RAC, a significantly smaller decoder both reduces computation (by approximately 70%) and improves reconstruction fidelity. It suggests that continuous flow-based decoding alleviates the representational burden on decoder capacity, enabling high-quality reconstruction under aggressive parameter reduction.

Visual Reconstruction Showcase. Figure 6 shows high-fidelity RAC reconstructions across animals, textures, architecture, and everyday objects on ImageNet 256×256 after only 30k training steps. RAC preserves fine-grained details—dog fur, snake scales, woven slippers, and ornate carpet patterns—that are challenging for conventional VAE decoders under aggressive spatial compression. It also maintains global structure and color, evident in the robin's sharp silhouette, the beer bottle's vivid green, and the Florence Cathedral's dome geometry.

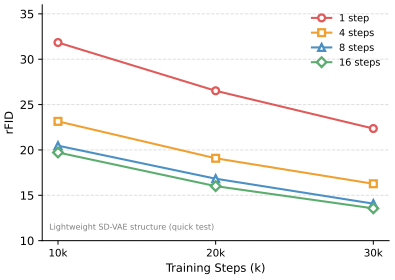


Figure 7: **rFID vs. Training Steps under Different Inference Steps.** Results on a lightweight SD-VAE structure for 1, 4, 8, and 16 decoding steps. Lower is better.

Decoding Steps vs. Reconstruction Quality. Figure 7 investigates how the number of inference steps in the RAC decoding process affects reconstruction quality (rFID) over the course of training. We conduct this analysis on a lightweight SD-VAE variant as a controlled ablation, evaluating four inference step configurations: 1-16 steps, measured at 10-30k training steps. As expected, increasing the number of decoding steps consistently improves rFID across all training checkpoints, with the gap between 1-step and 16-step decoding being most pronounced at early training stages. By 30k steps, rFID improves from 22.36 at 1 step down to 13.56 at 16 steps, demonstrating that the flow-based decoder benefits substantially from finer ODE integration, with clear separation between curves. The ordering remains consistent across checkpoints, confirming that additional decoding steps improve reconstruction quality.



Figure 8: **Generative Decoding under Ultra-short Training:** To visually demonstrate the benefits of our generative decoding on the results, we only trained for 1k steps (this is merely 1/10k of the full training duration in other works), and then we systematically observed the number of decoding inferences impacts performance. It's clearly see that during the initial training phase, multi-step decoding significantly contributed to the improvement in quality compared to single-step baselines.

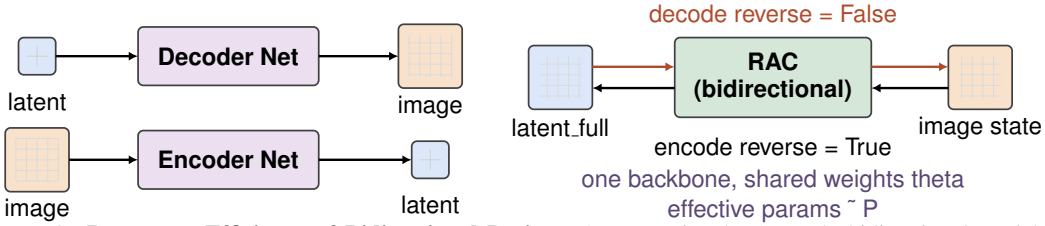


Figure 9: **Parameter Efficiency of Bidirectional Design.** A conventional VAE-style bidirectional model uses separate encoder and decoder backbones to map between compact latent and image spaces. In contrast, RAC uses a shared bidirectional model in the full-resolution state space, achieving encoding and decoding by reversing the flow direction rather than duplicating the backbone.

3.2 Generation Quality

Generated Image Samples. Figure 8 illustrates generative decoded image samples under ultra-short training (1k steps, 1/10k of full training duration), comparing 1-8 step generative decoding across diverse ImageNet categories. Even at this early training stage, the multi-step decoding of RAC progressively refines image quality: 1-step outputs capture coarse structure and semantics, while 4-step and 8-step inferences recover finer textures, sharper edges, and more faithful color rendition. This behavior demonstrates that RAC's generative decoder functions as an iterative refinement mechanism, effectively compensating for the limited encoder training by leveraging the diffusion prior to denoise and enhance latent representations at inference time.

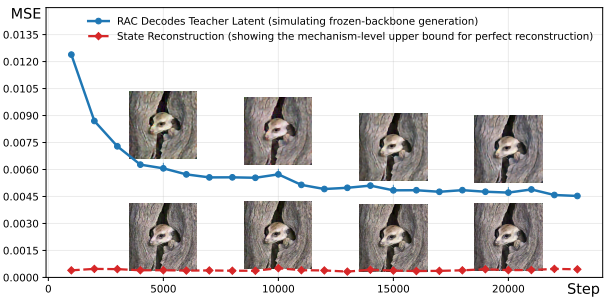
Generation Quality Comparison. Table 2 presents a quantitative comparison of RAC against baseline SiT models Ma et al. (2024) and REPA-E Leng et al. (2025). RAC consistently achieves lower gFID and sFID scores while improving Inception Score (IS) and Precision across all scales, demonstrating that the rectified autoencoder with consistency path provides complementary benefits across representation-aligned generative frameworks. Specifically, RAC reduces gFID by 2.7, 1.6, and 1.6 points over REPA-E for SiT-B, SiT-L, and SiT-XL respectively, while maintaining competitive Recall. These results confirm that RAC's improvements are not model-capacity-specific but scale consistently, making it a broadly applicable enhancement to rectified flow-based generative frameworks.

Table 2: **Model scale comparison on ImageNet 256×256.** Our RAC improves generation quality across all scales, achieving consistent gFID reductions over both the vanilla baseline and REPA-E.

Diff. Model	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B (130M)	49.5	7.00	27.5	0.46	**0.59**
+REPA-E	34.8	6.31	39.1	0.57	0.59
+RAC (Ours)	**32.1**	**6.18**	**41.3**	**0.59**	**0.60**
SiT-L (458M)	24.1	6.25	55.7	0.62	**0.60**
+REPA-E	16.3	5.69	75.0	0.68	0.60
+RAC (Ours)	**14.7**	**5.51**	**78.4**	**0.70**	**0.61**
SiT-XL (675M)	19.4	6.06	67.4	0.64	**0.61**
+REPA-E	12.8	5.04	88.8	0.71	0.58
+RAC (Ours)	**11.2**	**4.87**	**92.5**	**0.73**	**0.61**



Figure 10: **Encoding and Decoding Trajectory Symmetry.** Left: RAC decodes the full-resolution state to the image through a sequence of intermediate states. Right: the trajectories of sampled spatial positions are projected to a 2D PCA space, revealing that the decoding process follows structured and coordinated paths in the learned state space rather than independent pixel-wise changes.





Our learning mechanism is unique: With only 100 training steps, the state reconstruction achieves extremely high quality.

Figure 11: **Mechanism-level ability of RAC.** Left: reconstruction errors during training. Decoding teacher latents with RAC simulates frozen-backbone generation, while state reconstruction measures the upper bound of reconstruction within RAC's shared full-resolution state space. Right: qualitative state reconstruction examples.

3.3 Bidirectional Consistency

Parameter Efficiency of Bidirectional Design. As illustrated in Fig. 9, a conventional VAE-style bidirectional design relies on two separate networks: a decoder that maps a compact latent to image space and an encoder that maps the image back to the compact latent. This design achieves bidirectionality by duplicating the backbone across two directions. In contrast, RAC uses a single shared bidirectional model operating in the full-resolution state space. Encoding and decoding are both realized by the same model, with the transformation direction controlled by reversing the flow direction. This shared design avoids the need for a dedicated encoder-decoder pair and therefore provides a substantially more parameter-efficient bidirectional formulation.

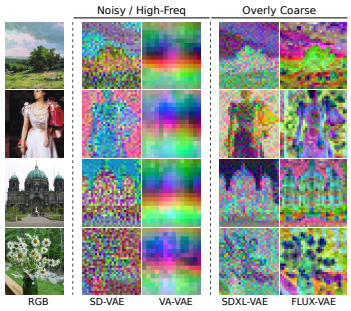
Encoding and Decoding Trajectory Symmetry. As shown in Fig. 10, RAC does not transform the representation by a single abrupt jump from latent-like initialization to the final image. Instead, the model evolves the sample through a sequence of full-resolution intermediate states, producing a smooth decoding trajectory in the shared state space. The left panel visualizes this evolution directly in image space, where the representation gradually acquires semantic structure and local detail over time. The right panel further projects the trajectories of sampled spatial positions into a 2D PCA space. Although each sampled position corresponds to a fixed spatial location, its feature vector follows a coherent trajectory rather than remaining static or changing independently. Our model does not require two unrelated transformations for encoding and decoding, but instead traverses a common representation manifold in opposite directions.

3.4 Generation–Reconstruction Gap Analysis

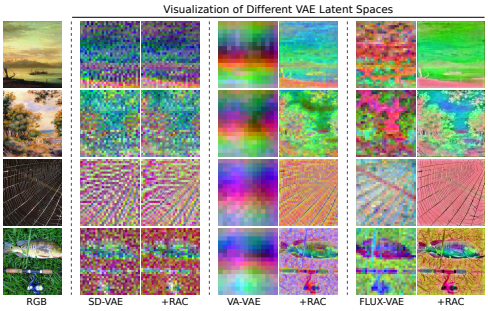
Generation–Reconstruction Gap Analysis. As shown in Figure 11, RAE Zheng et al. (2026) and Flux vae Labs (2024); Labs et al. (2025) achieve better results by increasing the number of channels and enhancing the expression ability of variables. However, if the state representation is used for training, it can learn a high-dimensional space without information loss. At this point, it is almost equivalent to fixed reconstruction learning generation. We can clearly see in the line graph that the generation ability is weaker than the

Table 3: **Training and decoding-step ablation:** We compare training configurations and decoding steps under 100K iterations of end-to-end training from scratch, evaluating their contributions to generation quality.

Method	gFID↓	sFID↓	IS↑	Prec.↑	Rec.↑
100K Iterations (20 Epochs)					
REPA Yu et al. (2024)	19.40	6.06	67.4	0.64	**0.61**
REPA-E (scratch)	14.12	7.87	83.5	0.70	0.59
RAC (scratch, 1-step)	14.05	7.83	83.9	0.70	0.61
RAC (scratch, 4-step)	**13.47**	**7.31**	**86.2**	**0.71**	0.61



(a) Latent artifacts in pretrained VAEs (Rombach et al., 2022; Podell et al., 2024; Labs, 2024; Yao et al., 2025)



(b) RAC yields cleaner backgrounds, richer structural details, and more stable colors, indicating improved spatial coherence in the learned latent representations.

Figure 12: Baseline latent manifolds show architecture-dependent anisotropy, with principal components dominated by high-frequency texture or coarse low-frequency blocks. RAC reallocates variance to spatially coherent components, yielding cleaner, more content-aligned latent geometry while preserving scene identity.

reconstruction. This directly confirms our assumption in the Introduction and indicates that we have discovered in the mechanism how to unify generation and reconstruction with a training objective.

Quantitative Gap between Generation and Reconstruction. As shown in Figure 11 (Right), with just 100 steps, our state reconstruction is nearly perfect. In practice, using encoded state condensed variables equates to a traditional variable-size compression paradigm. This robust behavior occurs because our training methodology constructs a high-dimensional continuous space mapped directly from the raw image, which subsequently stabilizes near the original pixel dimension. Consequently, even though this latent space is imperfect, it does not degrade reconstruction performance.

3.5 Ablation Study & Latent Space Analysis

Ablation Study on RAC Components. Table 3 suggests that even minimal RAC consistency regularization preserves latent quality. Increasing the integration from 1 to 4 steps further improves gFID, sFID, IS, and Precision, showing that the trajectory contributes measurable refinement beyond the shared parameterization alone. The gains arise under the same training budget and initialization, indicating that additional flow steps expose a more effective correction path for the learned latent space.

Latent Space PCA Visualization. Figure 12 highlights two complementary observations. First, different pretrained VAE architectures produce latent spaces with distinct but systematic defects: some are dominated by noisy high-frequency patterns, whereas others collapse fine details into coarse, low-information structures. Second, RAC-enhanced representations substantially regularize these latent distributions. Their spatially coherent principal components align with the consistent gFID and sFID improvements reported across the VAE backbones, linking the visual structure of the latent space to measurable generation benefits. This cross-architecture regularity shows that RAC provides a reliable refinement mechanism for pretrained representations.

4 Conclusion

This paper introduces RAC (Rectified Flow Auto Coder), an automatic encoding framework based on rectified flow, aiming to fundamentally solve the problem of inconsistency between generation and reconstruction in traditional VAEs. Unlike traditional VAEs which treat decoding as a single-step mapping, RAC models the decoding process as a continuous-time velocity field integration, enabling the model to gradually correct latent variables along the decoding path, thereby significantly reducing the performance gap between generation and reconstruction. Thanks to the time reversal mechanism, RAC naturally supports bidirectional inference, that is, the same velocity field model can simultaneously perform encoding and decoding functions, reducing the parameter quantity by nearly 41%. Experimental results show that RAC outperforms existing SOTA VAEs in terms of both reconstruction and generation performance, even under aggressive 10× parameter reduction.

AI USE STATEMENT

In this work, generative AI tools were used to polish the manuscript prose and assist with preparing explanatory illustrations. Generative AI tools were not used to design the model. All AI-assisted text and visual materials were reviewed and edited by the authors. The authors take responsibility for the final content of this paper, including all text, claims, and artifacts produced with AI assistance.

REFERENCES

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. FlexTok: Resampling images into 1D token sequences of flexible length. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2241–2292. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bachmann25a.html.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Yinbo Chen, Rohit Girdhar, Xiaolong Wang, Sai Saketh Rambhatla, and Ishan Misra. Diffusion autoencoders are scalable image tokenizers. *arXiv preprint arXiv:2501.18593*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021.

Sen Fang, Chen Chen, Lei Wang, Ce Zheng, Chunyu Sui, and Yapeng Tian. Signllm: Sign language production large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 6622–6634, October 2025a.

Sen Fang, Chunyu Sui, Yanghao Zhou, Xuedong Zhang, Hongbin Zhong, Yapeng Tian, and Chen Chen. Signdiff: Diffusion model for american sign language production. In *2025 IEEE 19th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–11, 2025b. doi: 10.1109/FG61629.2025.11099440.

Sen Fang, Yalin Feng, Chunyu Sui, Hongbin Zhong, Hongwei Yi, and Dimitris N. Metaxas. Signx: Continuous sign recognition in compact pose-rich latent space, 2026a. URL https://arxiv.org/abs/2504.16315.

Sen Fang, Yalin Feng, Hongbin Zhong, Yanxin Zhang, and Dimitris N. Metaxas. Stable signer: Hierarchical sign language generative model, 2026b. URL https://arxiv.org/abs/2512.04048.

Sen Fang, Hongbin Zhong, Yalin Feng, Yanxin Zhang, and Dimitris N. Metaxas. Streamflow: Theory, algorithm, and implementation for high-efficiency rectified flow generation, 2026c. URL https://arxiv.org/abs/2511.22009.

Yuchao Gu, Xintao Wang, Yixiao Ge, Ying Shan, and Mike Zheng Shou. Rethinking the objectives of vector-quantized tokenizers for image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7631–7640, June 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. *arXiv preprint arXiv:2502.09509*, 2025.

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

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers, 2025. URL https://arxiv.org/abs/2504.10483.

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

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023. doi: 10.1109/ICCV51070.2023.00387.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10619–10629, June 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 1690–1712, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.89. URL https://aclanthology.org/2025.findings-emnlp.89/.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19667–19679. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e3b21256183cf7c2c7a66be163579d37-Paper.pdf.

Théophane Vallaeys, Jakob Verbeek, and Matthieu Cord. Ssdd: Single-step diffusion decoder for efficient image tokenization. *arXiv preprint arXiv:2510.04961*, 2025.

Mengmeng Wang, Dengyang Jiang, Liuzhuozheng Li, Yucheng Lin, Guojiang Shen, Xiangjie Kong, Yong Liu, Guang Dai, and Jingdong Wang. Vae-repa: Variational autoencoder representation alignment for efficient diffusion training, 2026. URL https://arxiv.org/abs/2601.17830.

Xin Wen, Bingchen Zhao, Ismail Elezi, Jiankang Deng, and Xiaojuan Qi. "principal components" enable a new language of images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16641–16651, October 2025.

Chenqian Yan, Songwei Liu, Hongjian Liu, Xurui Peng, Xiaojian Wang, Fangmin Chen, Lean Fu, and Xing Mei. Hybrid sd: Edge-cloud collaborative inference for stable diffusion models, 2024. URL https://arxiv.org/abs/2408.06646.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Shuangfei Zhai, Ruixiang ZHANG, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2uheUFcFsM.

Long Zhao, Sanghyun Woo, Ziyu Wan, Yandong Li, Han Zhang, Boqing Gong, Hartwig Adam, Xuhui Jia, and Ting Liu. Epsilon-VAE: Denoising as visual decoding. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 77740–77759. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhao25w.html.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0u1LigJaab.

Xianwei Zhuang, Yuxin Xie, Yufan Deng, Liming Liang, Jinghan Ru, Yuguo Yin, and Yuexian Zou. Vargpt: Unified understanding and generation in a visual autoregressive multimodal large language model, 2025. URL https://arxiv.org/abs/2501.12327.

A Related Work

A.1 Flow-Based Generative Models

Normalizing flow models Zhai et al. (2025) learn invertible mappings between complex data distributions and simple priors. Early works such as Real NVP Dinh et al. (2017) and Glow Kingma & Dhariwal (2018) establish the foundation of flow-based generation by enabling tractable likelihood estimation through invertible transformations. Recent studies extend flow models toward diffusion-inspired formulations. Flow Matching (FM) Lipman et al. (2023) and Rectified Flow (RF) Liu et al. (2023) reformulate diffusion training as direct vector field regression, enabling deterministic sampling and efficient training. These methods bridge continuous normalizing flows Chen et al. (2018); Fang et al. (2026c); Song et al. (2023) and diffusion models Rombach et al. (2022); Salimans & Ho (2022), achieving competitive generation quality with reduced sampling steps.

A.2 Variational Autoencoders and Representation Learning

Variational Autoencoders (VAEs) Kingma & Welling (2013) provide a principled framework for learning latent-variable generative models via variational inference. However, standard VAEs often suffer from posterior collapse and limited reconstruction fidelity. To alleviate these issues, Representations Autoencoders (RAEs) Zheng et al. (2026) replace KL regularization with explicit latent constraints. Sparse Autoencoders (SAEs) Shu et al. (2025) enforce sparsity priors to promote compact and interpretable representations.

Recent advances focus on improving latent expressiveness and generative performance. REPA-E Leng et al. (2025) introduces residual posterior alignment for end-to-end latent diffusion training. Rethinking VQ Tokenizers Gu et al. (2024) identifies a competing relationship between reconstruction detail and semantic compression, and improves decoding while preserving the learned semantic latent space. VQGAN Esser et al. (2021) adopts vector-quantized autoencoders for efficient latent generative modeling. Hybrid diffusion-autoencoder frameworks Yan et al. (2024) further improve inference and generation consistency. NVAE Vahdat & Kautz (2020) proposes hierarchical latent variables for scalable high-resolution synthesis. These approaches demonstrate that carefully designed latent regularization and architectural constraints are crucial for balancing reconstruction quality and generative diversity.

A.3 Unified Modeling of Generation and Reconstruction

Recent research explores unifying generation and reconstruction within a single probabilistic framework. Latent diffusion models Rombach et al. (2022) combine autoencoding architectures with diffusion processes for efficient high-resolution synthesis.

Several works interpret autoencoding and sampling as two sides of the same stochastic process. Score-based models Song et al. (2021) and flow matching methods Lipman et al. (2023) naturally support both data reconstruction and generation via conditional sampling. Consistency models Song et al. (2023) further enable fast bidirectional mappings.

Recent end-to-end latent learning frameworks Leng et al. (2025); Wang et al. (2026); Fang et al. (2025a;b; 2026b;a) explicitly couple inference and generation modules, improving representation alignment and robustness. Such unified designs facilitate efficient downstream adaptation and controllable synthesis. Our work follows this line of research by jointly modeling reconstruction and generation within a unified flow-based latent framework, achieving improved fidelity and sampling efficiency.

A.4 Generative Decoders and Diffusion Tokenizers

A closely related line of work replaces deterministic autoencoder decoding with an iterative generative process. Epsilon-VAE Zhao et al. (2025) treats visual decoding as conditional denoising, while DiTo Chen et al. (2025) scales diffusion-based image tokenizers under a unified diffusion objective. SSDD Vallaeys et al. (2025) develops a diffusion decoder and distills it into an efficient single-step model. Earlier Diffusion Autoencoders Preechakul et al. (2022) combine a semantic encoder with diffusion-based reconstruction, whereas FlexTok Bachmann et al. (2025) and SEMANTICIST Wen

Table 4: **Mechanism-level comparison with generative-decoder and diffusion-tokenizer methods.** *Full sharing* means that the complete encoding and decoding transformations use one parameter set; *full reversal* means that the complete encoded state is obtained by reversing the same field used for decoding.

Method	Generative decoder	Independent encoder	Full sharing	Full reversal	Explicit round-trip loss	Representation / compatibility
Epsilon-VAE Zhao et al. (2025)	Conditional diffusion	Yes	No	No	No	Encoder-conditioned continuous latent
DiTo Chen et al. (2025)	Diffusion decoder	Yes	No	No	No	Jointly learned continuous tokenizer
SSDD Vallaeys et al. (2025)	Diffusion + distillation	Yes	No	No	No	KL-VAE replacement
Diffusion Autoencoders Preechakul et al. (2022)	DPM/DDIM	Semantic encoder	Partial	Stochastic code only	No	Two-part semantic/ stochastic code
FlexTok Bachmann et al. (2025)	Rectified flow	Yes	No	No	No	Variable-length 1D tokens
SEMANTICIST Wen et al. (2025)	Conditional diffusion	Yes	No	No	No	Structured 1D causal tokens
RAC (ours)	**Rectified-flow VAE decoder**	**Teacher only during training**	**Yes**	**Yes**	**Yes**	**Pretrained KL-VAE latent spaces**

et al. (2025) use generative decoders for structured one-dimensional tokenization. These works establish generative decoding as an effective alternative to a deterministic decoder, but retain an independent encoder for all or part of the representation.

RAC converts a lightweight VAE decoder into a bidirectional velocity field whose complete encoding and decoding paths share parameters, and explicitly regularizes their reversal through round-trip consistency. The frozen teacher encoder supplies latent supervision during training. At inference, reverse-time integration uses the same RAC network for encoding. This design preserves compatibility with existing KL-VAE latent spaces while keeping the teacher encoder fixed.

B Implementation Details

B.1 Algorithm and Implementation Details

We use a KL-VAE teacher for latent supervision, AdamW for optimization, and mixed precision to reduce memory. The same parameters are shared for forward and reverse flows, which keeps the model compact while enforcing bidirectional structure. A random time grid and step noise are used to avoid overfitting to fixed integration schedules.

Architectural details. The learned latent projector is a four-layer convolutional network with a base channel width of 40 (approximately 100K parameters) that downsamples the encoded full-resolution state to the teacher latent size. The optional relative positional encoding is computed on normalized spatial coordinates $x, y \in [-1, 1]$ as $\mathrm{pos}(x, y) = 0.01[\cos(\pi x), \cos(\pi y)]$ and concatenated with the downsampled state and time channel.

Algorithm 1 RAC Training (one iteration)

Require: image batch $\mathbf{x}$, teacher AE, steps K
$\tilde{\mathbf{x}} \leftarrow 2\mathbf{x} - 1$
$\mathbf{z}_T \leftarrow \mathrm{Enc}_T(\tilde{\mathbf{x}})$
$\mathbf{s}_0 \leftarrow \mathrm{expand}(\mathrm{pad}(\mathbf{z}_T))$
$\{\mathbf{s}_k\}_{k=0}^{K} \leftarrow \mathrm{Flow}_\theta(\mathbf{s}_0)$
$\mathbf{s}^* \leftarrow \mathrm{pad}(\tilde{\mathbf{x}})$
compute $\mathcal{L}_{\text{recon}}, \mathcal{L}_{\text{path}}^{\text{dec}}$
$\hat{\mathbf{s}}_0 \leftarrow \mathrm{Flow}_\theta^{-1}(\mathbf{s}^*)$
$\hat{\mathbf{z}} \leftarrow \mathrm{down}(\hat{\mathbf{s}}_0)$
compute $\mathcal{L}_{\text{latent}}, \mathcal{L}_{\text{pixel}}, \mathcal{L}_{\text{rt}}$
$\mathcal{L} \leftarrow$ Eq. equation 9; update θ with AdamW

Default settings. Unless otherwise noted, as shown in Algorithm 1, we use K=4 Euler steps, C_s=4 state channels, AdamW with learning rate 3×10^{-4} and $(\beta_1, \beta_2) = (0.9, 0.99)$, input resolution 256, and a small step noise ($\sigma = 0.05$). We enable random time grids during training and use mixed precision when available. The weights are typically set to $\lambda_{\text{path}} = 0.1$, $\lambda_{\text{latent}} = 1.0$, $\lambda_{\text{pixel}} = 1.0$, and $\lambda_{\text{rt}} = 1.0$.

Teacher and latent protocol. The pretrained KL-VAE teacher encoder remains frozen throughout RAC training. During training, $\mathbf{z}_T = \mathrm{Enc}_T(\tilde{\mathbf{x}})$ provides a data-conditioned latent target. During generation, the attached backbone (e.g., SiT, DiT, or UNet) supplies $\mathbf{z}_T$ and determines its

distribution. For a 256×256 image with an SD-VAE teacher, the latent size is typically $4 \times 32 \times 32$. Reported RAC training-step counts measure optimization of the velocity field and, when used, the lightweight latent projector; the teacher encoder remains fixed.

Operator and reversal details. The pad operator appends channels filled with 0.5, placing the image and latent-derived endpoints in the same C_s-channel state space; with the default $C_s{=}4$, an RGB image receives one padding channel. The expand operator repeats each compact latent position over its corresponding spatial block, preserving all latent values in the full-resolution state. The down operator maps a reverse-integrated state back to the teacher latent size through the projector described above, and Π selects the first three state channels as RGB. At every Euler step, the velocity field has the same $C_s \times H \times W$ shape as the evolving state. Encoding uses exactly the same parameters as decoding and reverses only the direction of Euler integration.

Why the teacher encoder is frozen. Joint optimization of an unconstrained encoder and RAC from scratch may learn an approximately invertible RGB shortcut that yields a weak latent for downstream generation. The high-dimensional intermediate trajectory can still carry information when the latent size is reduced. Freezing the pretrained encoder preserves an established latent space and isolates improvements due to RAC. This choice prioritizes representation stability and compatibility. Multi-step reverse integration requires more computation than one teacher-encoder pass.

Scaled variants and parameter accounting. The $0.1\times$, $0.2\times$, and $0.3\times$ decoder variants uniformly adjust the channel widths of all layers to reach 10%, 20%, and 30% of the full decoder parameter count, respectively, while preserving depth, upsampling stages, kernel sizes, and overall topology. This construction isolates decoder capacity, and each scale factor denotes the resulting total-parameter ratio relative to the full decoder. The reported 40.82% reduction compares the parameter count of a conventional teacher encoder–decoder pair with RAC's single shared bidirectional model. The reduction concerns model parameters and deployment storage. Encoding latency is reported separately because reverse integration uses multiple steps.

C Additional Experiments

C.1 Comparison of Released Generative Decoders

We compare RAC with generative-decoder and tokenizer systems for which released pretrained checkpoints can be evaluated through their native reconstruction pipelines. All methods receive the same held-out inputs and use identical image preprocessing and metric implementations. Because each released system retains its native latent representation, architecture, and inference schedule, Table 5 compares complete released systems under their native architectures, latent representations, and inference schedules.

Table 5: **Pixel-fidelity comparison using released checkpoints.** All methods use the same held-out inputs and evaluation code. Decoder and total counts distinguish decoder parameters from the complete system; RAC shares one network across both directions.

Method	Decoder params.	Total params.	Decode NFE	PSNR ↑	SSIM ↑
FlexTok-d12-d12 Bachmann et al. (2025)	172.45M	341.65M	20	17.152	0.4598
SEMANTICIST-L Wen et al. (2025)	457.05M	609.76M	250	22.199	0.6521
SSDD-S-F8C4 Vallaeys et al. (2025)	14.06M	48.22M	8	19.863	0.5126
RAC-Tiny (ours)	**1.46M**	**1.46M**	**4**	**23.917**	**0.7063**

Despite using only 1.46M shared parameters, RAC-Tiny obtains the strongest PSNR and SSIM among the evaluated generative-decoder systems, improving PSNR over SSDD-S, FlexTok, and SEMANTICIST by 4.054, 6.765, and 1.718 dB, respectively. This performance is achieved with 33–417$\times$ fewer total parameters than the compared generative-decoder systems.

Table 6: **Robustness to relative Gaussian perturbations of each method's native latent.** PSNR gains and degradation differences are paired across the same images. The final column reports the reduction in output-drift MSE achieved by RAC relative to TAESD.

Relative σ	TAESD PSNR ↑	RAC-4 PSNR ↑	RAC paired gain ↑	TAESD ΔPSNR	RAC-4 ΔPSNR	RAC robustness gain ↑	Output-drift reduction ↑
0.0	24.157	24.694	$+0.537 \pm 0.060$	0.000	0.000	0.000	–
0.1	23.675	24.355	$+0.680 \pm 0.054$	-0.482	-0.339	$+0.143 \pm 0.011$	80.2%
0.2	22.469	23.441	$+0.972 \pm 0.047$	-1.688	-1.253	$+0.435 \pm 0.026$	70.1%
0.3	20.927	22.154	$+1.226 \pm 0.045$	-3.229	-2.540	$+0.690 \pm 0.037$	60.8%
0.5	17.812	19.141	$+1.329 \pm 0.050$	-6.344	-5.552	$+0.792 \pm 0.048$	44.6%

C.2 Robustness to Perturbed Latents

To directly test whether multi-step decoding is robust to inaccurate generation latents, we compare RAC with the single-step TAESD baseline under controlled latent perturbations. TAESD uses the latent produced by its frozen encoder, whereas RAC obtains its latent by reversing the shared velocity field for four steps and applying the learned compact projector. Both representations have size $4 \times 32 \times 32$. For each method, we perturb its native clean latent as

$$\mathbf{z}_\sigma = \mathbf{z}_{\text{native}} + \sigma \,\text{std}(\mathbf{z}_{\text{native}})\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{10}$$

We evaluate 1,000 ImageNet images at 256×256 resolution, using the same four decoding steps as in RAC training. Differences are computed pairwise for each image before reporting their mean and 95% confidence interval. Output drift is the MSE between the decoded outputs from perturbed and clean latents.

At $\sigma = 0$, RAC-4 improves PSNR by 0.537 dB. As the latent shift increases, its paired advantage grows to 1.329 dB. More importantly, RAC degrades less at every nonzero perturbation level and reduces output drift by 44.6–80.2%, showing that multi-step decoding is less sensitive to latent error and can progressively correct deviations along its trajectory.

C.3 Scaled-Decoder Generation Quality and Efficiency

We first evaluate the scaled RAC decoders with the frozen DiT-XL backbone reported by RAE Zheng et al. (2026). Every configuration uses the same backbone checkpoint trained for 80 epochs, the same sampling and evaluation protocol, and no classifier-free guidance. The SD-VAE baseline is the single-step result reported by RAE; each RAC variant changes only the decoder under this matched setting.

All three scaled RAC decoders improve gFID and IS over the single-step SD-VAE while using fewer decoder GFLOPs. The $0.1\times$ model reduces gFID from 7.13 to 6.98 with 94.3 GFLOPs, and the $0.3\times$ model reaches the strongest overall generation result with 6.74 gFID, 129.2 IS, and 0.66 recall. These controlled results directly show that the reconstruction gains from decoder scaling extend to end-to-end generation quality.

Table 7: **Scaled RAC generation with frozen DiT-XL.** All settings use 80 epochs without CFG; GFLOPs measure decoder cost.

Decoder	gFID↓	IS↑	Prec.↑	Rec.↑	GFLOPs↓
SD-VAE-0.1×	6.98	126.4	0.70	0.65	**94.3**
SD-VAE-0.2×	6.86	127.9	**0.71**	0.65	152.4
SD-VAE-0.3×	**6.74**	**129.2**	**0.71**	**0.66**	254.5
SD-VAE Zheng et al. (2026)	7.13	124.9	0.70	0.65	310.4

We next measure the complete compact-latent-to-image decoding process on one Quadro RTX 8000 at FP16 and 256×256 resolution, with 20 warm-up runs followed by 100 CUDA-event timed runs. Table 8 reports decoder-local cost under the default $K = 4$ setting.

Table 8: **Decoder-local efficiency under matched hardware and precision.** Batch-8 latency is reported per image.

Method	NFE	Total params.	B1 latency	B8 latency/image	B1 peak memory
Standard SD-VAE	1	83.65M	29.059 ± 0.117 ms	31.144 ms	319.2 MiB
RAC (default)	4	49.51M	106.421 ± 0.284 ms	125.659 ms	319.4 MiB

The four-step RAC decoder has $3.66\times$ the Batch-1 latency of the single-step SD-VAE decoder, while reducing total autoencoder parameters from 83.65M to 49.51M (40.82%) and retaining nearly identical peak activation memory. The integration step count provides explicit control over this quality–latency trade-off.

At the pipeline level, decoder computation remains small. SiT-XL/2 requires approximately 118.6 GFLOPs per forward pass and 29,650 GFLOPs over 250 sampling steps. The lightweight RAC variants require 94.3–254.5 GFLOPs, only 0.32%–0.86% of the backbone's sampling cost. Thus, multi-step decoding raises decoder-local latency but adds little end-to-end computation.

C.4 Additional Experimental Protocol Clarifications

Several evaluated works introduce a coupled system consisting of a generative backbone and its associated autoencoder or decoder. Generation metrics therefore characterize the complete backbone–decoder configuration. Our tables expose individual components in the method labels so that each comparison identifies the axis being varied. Consequently, the same system-level result can appear in more than one table when it is the shared intersection of two analyses. For example, the SD-VAE configuration in Table 1 uses the default SiT-L backbone and therefore coincides with the SiT-L configuration in Table 2, which uses SD-VAE. Likewise, the SiT-XL REPA baseline shared by Tables 2 and 3 refers to the same 100K-iteration (20-epoch) configuration. These repeated entries preserve a common experimental anchor across VAE, model-scale, and training analyses.

Within every VAE block in Table 1, the first row defines the common vanilla system. The rows prefixed by "+" are mutually exclusive branches from this shared baseline. The "+REPA-E" branch uses the corresponding REPA-E update, while the "+RAC" branch uses a single RAC decoder trained with supervision from the frozen baseline encoder. Thus, "+RAC" replaces the decoder branch associated with the baseline configuration; the resulting system contains one RAC decoder, with REPA-E and RAC evaluated separately. The host generative backbone, dataset, resolution, training budget, sampling schedule, and evaluation protocol are matched within each comparison. Published baseline entries follow the configurations reported by their originating studies, and RAC is evaluated by changing the designated autoencoder/decoder component under the matched host setup.

This protocol also explains how a frozen teacher encoder is compatible with improved generation metrics. The frozen encoder supplies stable target latents while RAC is trained; the attached SiT, DiT, or UNet supplies latents during generation, and the RAC trajectory maps those latents into images. Metrics such as gFID, sFID, IS, precision, and recall are computed from the final decoded images and therefore reflect the coupled behavior of the generative backbone and decoder. A stronger decoder can improve these image-space metrics while the teacher encoder remains fixed. Equal values in an individual metric across distinct configurations can occur independently; the complete metric vector and the stated backbone–decoder configuration identify each result.