COMPACT SEMANTIC REPRESENTATIONS ENABLE NORMALIZING FLOWS FOR IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Normalizing flows (NFs) provide exact likelihoods and deterministic invertible sampling, but have historically lagged behind diffusion models for large-scale image generation. We identify a key obstacle: NFs are required to learn a single invertible transport over the full ambient space, making them highly sensitive to high-dimensional representations. This leads to a semantic-capacity mismatch in modern visual representation spaces, where semantic information is compact but encoded in overcomplete features. We propose **SRC-Flow**, which introduces a Semantic Representation Compressor (SRC) to compact high-dimensional RAE features into a low-dimensional semantic space before flow modeling and preserve reconstruction through the frozen RAE decoder. This compact space reduces the modeling burden of NFs and enables effective likelihood-based generation in semantic representation space. We further adopt constant noise regularization tailored to the fixed unconditional bijection learned by flows. On ImageNet 256×256 and 512×512, SRC-Flow achieves state-of-the-art generation quality among normalizing flow methods, with gFID scores of 1.65 and 2.07 under classifier-free guidance, while retaining exact likelihood computation in the compact semantic representation space and deterministic invertible sampling at the flow level.

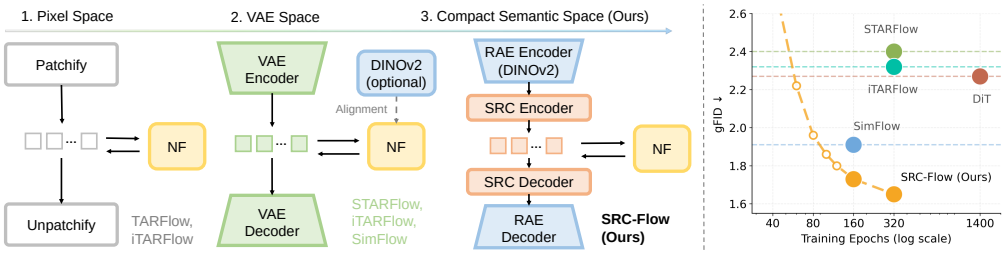


Figure 1: SRC-Flow moves NF modeling from pixels or VAE latents to compact semantic representations produced by the SRC, achieving state-of-the-art generation quality among normalizing flow methods.

1 INTRODUCTION

Normalizing flows (NFs) (Rezende & Mohamed, 2015; Dinh et al., 2017) are generative models with exact likelihood computation and deterministic invertible sampling. These properties make NFs theoretically appealing, but their image generation quality has historically lagged behind diffusion models (Ho et al., 2020; Song et al., 2021b;a; Karras et al., 2022). Recent Transformer-based flows have narrowed this gap: TARFlow (Zhai et al., 2025) revisited pixel-space autoregressive flows, STARFlow (Gu et al., 2025) scaled NFs in VAE latent space, SimFlow (Zhao et al., 2025) simplified latent NF training, and iTARFlow (Chen et al., 2026b) improved sampling through iterative denoising. These advances suggest that NFs remain promising, but also raise a central question: *what representation space is suitable for normalizing flows?*

This choice of representation space is especially important for NFs because their exact likelihood objective requires an invertible transport over the full ambient space. Unlike diffusion or rectified-flow models, which can redistribute learning across noise levels through time-dependent denoising, an NF learns a single fixed bijection between data and prior. Thus, every modeled dimension contributes to the likelihood and log-determinant. Pixel-space NFs suffer from extreme dimensionality, while VAE-latent NFs reduce dimensionality but operate in latents with limited semantic structure.

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.

Representation Autoencoders (RAEs) (Zheng et al., 2026a) provide an appealing alternative by pairing a frozen pretrained vision encoder, such as DINOv2 (Oquab et al., 2024; Caron et al., 2021), with a trained decoder. Their representations are semantically rich and structurally organized, but they are also high-dimensional and overcomplete. This leads to a *semantic-capacity mismatch*: the semantic information needed for generation is compact, whereas an NF must model the full high-dimensional representation through exact invertible transport. Diffusion models can partially absorb this mismatch through dimension-dependent noise schedule shifts (Esser et al., 2024), but NFs have no timestep or noise schedule mechanism. Consequently, applying NFs to full RAE representations forces the bijection to model every channel under exact invertibility, including redundant or weakly informative ones, increasing the burden of likelihood training.

We propose **SRC-Flow**, a normalizing-flow framework built on compact semantic representations. Instead of directly modeling full RAE tokens, SRC-Flow introduces a **Semantic Representation Compressor (SRC)** that compresses high-dimensional semantic features from dimension n to a compact dimension $d \ll n$ before flow modeling, while preserving reconstruction through the frozen RAE decoder. The motivation is supported by the compact structure of pretrained visual representations: a small number of channels preserves most of the semantic variation in RAE features. By reducing ambient dimensionality while retaining semantic content, SRC enables NFs to benefit from pretrained representation spaces without being overwhelmed by their overcomplete structure.

We further find that the per-example noise strategy inherited from RAE decoder training is suboptimal for flow learning. Because an NF learns a single unconditional bijection, modeling a mixture of differently perturbed distributions increases the difficulty of likelihood training. We therefore adopt a constant noise regularization strategy, which better matches the transport learned by NFs.

Our contributions are threefold. First, we identify a semantic-capacity mismatch between high-dimensional pretrained representations and exact-likelihood normalizing flows, verified with a naive full-representation baseline. Second, we propose SRC, which extracts compact semantic representations from frozen RAE features while preserving decoder compatibility. Third, SRC-Flow achieves state-of-the-art quality among NF methods on ImageNet 256×256 and 512×512, with gFID scores of 1.65 and 2.07 under classifier-free guidance, while retaining exact likelihood computation in the compact semantic representation space and deterministic invertible sampling at the flow level.

2 Preliminaries

Representation Autoencoders. A Representation Autoencoder (RAE) (Zheng et al., 2026a) pairs a frozen pretrained visual encoder with a trained image decoder. We use DINOv2-B (Oquab et al., 2024) as the encoder E. Given an image $x \in \mathbb{R}^{3\times H\times W}$, E produces patch tokens as follows:

$$z_{\text{raw}} = E(x) \in \mathbb{R}^{N\times n}, \qquad z = \frac{z_{\text{raw}} - \mu_{rae}}{\sigma_{rae}}, \tag{1}$$

where $N = \frac{H}{p} \times \frac{W}{p}$ is the number of spatial tokens, n is the channel dimension, and $(\mu_{rae}, \sigma_{rae})$ are precomputed channel-wise statistics. The normalized feature $z \in \mathbb{R}^{N\times n}$ is the semantic representation used in this work. A frozen RAE decoder D reconstructs images from denormalized features. Compared with VAE latents, RAE representations inherit richer semantic structure from pretrained models, but their high ambient dimension poses a challenge for exact-likelihood normalizing flows.

Normalizing flows. A normalizing flow (Rezende & Mohamed, 2015; Dinh et al., 2017) defines an invertible mapping $f_\theta : \mathcal{Y} \to \mathcal{U}$ from a data distribution $p_{\mathcal{Y}}$ to a simple prior $p_{\mathcal{U}} = \mathcal{N}(0, I)$, typically composed as $f_\theta = f_{K-1} \circ f_{K-2} \circ \cdots \circ f_0$. For a modeling variable $y \in \mathcal{Y}$ and its corresponding prior variable $u = f_\theta(y)$, the exact log-likelihood is given by the change-of-variables formula:

$$\log p_{\mathcal{Y}}(y) = \log p_{\mathcal{U}}(f_\theta(y)) + \log \left| \det \frac{\partial f_\theta}{\partial y} \right|. \tag{2}$$

Generation samples $u \sim \mathcal{N}(0, I)$ and applies the inverse mapping $y = f_\theta^{-1}(u)$. Unlike diffusion models, which learn time-dependent denoising processes, NFs learn a single deterministic bijection and optimize exact likelihood over the full modeled space.

Transformer autoregressive flow. We instantiate f_θ with the Transformer Autoregressive Flow (TAF) used in recent flow-based image generation methods (Zhai et al., 2025; Gu et al., 2025; Zhao et al., 2025). TAF consists of K autoregressive affine blocks. For the i-th token in block k, a causal Transformer predicts shift and log-scale parameters (μ_i^k, α_i^k) from previous tokens and an optional class label. The forward data-to-prior mapping and reverse sampling step are as follows:

$$y_i^{k+1} = (y_i^k - \mu_i^k) \odot \exp(-\alpha_i^k), \qquad y_i^k = y_i^{k+1} \odot \exp(\alpha_i^k) + \mu_i^k. \tag{3}$$

The autoregressive structure yields a tractable triangular Jacobian, and generation proceeds sequentially in the inverse direction. Following SimFlow (Zhao et al., 2025), we use a *deep-shallow* design with $K = 6$ blocks: the first $K - 1$ blocks are shallow, while the last block is deep, and enable classifier-free guidance by randomly dropping class labels during training. In SRC-Flow, the variable y is the compact semantic representation produced by the Semantic Representation Compressor.

3 Method

We present **SRC-Flow**, a framework that enables normalizing flows to model compact semantic representation spaces. SRC-Flow first compresses high-dimensional pretrained visual features into a compact representation, and then trains an exact-likelihood normalizing flow in this space. We begin by identifying the semantic-capacity mismatch between representation spaces and normalizing flows (Section 3.1), then introduce the Semantic Representation Compressor (SRC) (Section 3.2), the noise regularization strategy (Section 3.3), and the full training and inference pipeline (Section 3.4).

3.1 Semantic-Capacity Mismatch in Representation-Space Flows

Modern pretrained visual representations provide rich semantic structure, but their high ambient dimensionality poses a challenge for NFs. Let $z \in \mathbb{R}^{N \times n}$ denote the normalized RAE representation in Eq. (1), where N is the number of spatial tokens and n is the token channel dimension.

Although the effective semantic information is compact, the ambient dimension Nn is large and overcomplete. For NFs, every modeled channel contributes to the likelihood objective and the log-determinant, forcing the flow to learn an exact invertible transport over the full representation space.

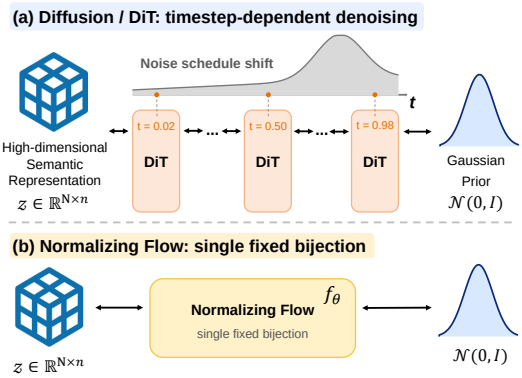


Figure 2: Diffusion adapts through timestep-dependent noise schedule shifts, while NFs learn a single fixed bijection over full representation space.

Diffusion and rectified-flow models, in contrast, learn time-dependent fields and can redistribute learning across noise levels; RAE-based diffusion, for example, calibrates training for high-dimensional representations with a dimension-dependent schedule shift $t' = \eta t/(1 + (\eta - 1)t)$ (Zheng et al., 2026a). An NF has no such mechanism: it learns a *single fixed bijection* $f_\theta : \mathcal{Y} \to \mathcal{U}$ without a timestep or noise schedule (Figure 2), and must therefore transport the entire distribution of $z \in \mathbb{R}^{N \times n}$ to the Gaussian prior. This leads to a *semantic-capacity mismatch*: semantic content is compact, but the flow is forced to spend capacity modeling the full ambient representation.

We empirically verify this mismatch with a naive full-representation baseline. We train the Transformer autoregressive flow directly on the full normalized RAE tokens.

Table 1: Naive baseline.

CFG	Hidden Dim.	gFID↓
✗	1152	11.53
✗	2048	11.46
✓	1152	3.58
✓	2048	3.54

Since each token requires predicting $2n$ affine parameters for shift and log-scale, we compare the default hidden dimension 1152 with an enlarged hidden dimension 2048. As shown in Table 1, increasing hidden width brings almost no improvement under either guided or unguided sampling. This suggests that direct full-space modeling is ill-suited for current flow architectures. The exact compact-space NLL is similarly insensitive to flow width (Appendix D.4). Rather than further scaling the flow, we construct a compact semantic representation that preserves the effective information in RAE features while reducing the ambient dimension seen by the flow.

3.2 Semantic Representation Compressor

Compact structure of semantic representations. The analysis above shows that directly modeling the full RAE token space is ill-suited for normalizing flows. However, pretrained visual representations are not uniformly informative across all channels: they are ambiently high-dimensional but semantically compressible.

We verify this by performing PCA on normalized RAE features across ImageNet. As shown in Figure 3, the cumulative explained variance rises rapidly and saturates early: the first 32 principal components already explain 99.06% of the total variance. This indicates that most semantic variation can be captured by tens of dimensions rather than the full token dimension n. We therefore seek a learnable compressor that preserves semantic information while reducing the ambient dimensionality modeled by the flow.

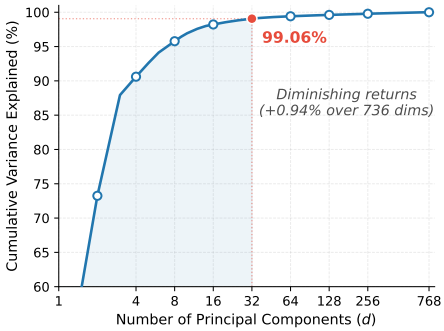


Figure 3: PCA of normalized RAE features. The first 32 components explain 99.06% variance.

SRC architecture. We introduce a learnable Semantic Representation Compressor (SRC) between the frozen RAE encoder and the normalizing flow. The SRC contains an encoder C_{enc} and a decoder C_{dec}. Given a normalized RAE representation $z \in \mathbb{R}^{N\times n}$, the SRC first compresses it to a compact representation and then reconstructs the original representation dimension:

$$\tilde{z}_c = C_{\text{enc}}(z) \in \mathbb{R}^{N\times d}, \qquad \hat{z} = C_{\text{dec}}(\tilde{z}_c) \in \mathbb{R}^{N\times n}. \tag{4}$$

The reconstructed representation $\hat{z}$ is denormalized and decoded by the frozen RAE decoder.

A key design choice is to use Transformer blocks for compression. RAE tokens come from a Vision Transformer encoder and contain long-range semantic correlations across spatial positions. Token-wise projections compress each token independently, while convolutional compressors primarily capture local interactions, and they are both empirically less effective in our ablations.

In contrast, Transformer blocks aggregate global token information through self-attention like pretrained vision encoders, which is important because the flow operates only on the compressed representation.

Concretely, the SRC encoder applies L Transformer blocks followed by a projection layer from n to d channels, while the decoder mirrors this structure with a projection layer from d back to n followed by L Transformer blocks, as shown in Figure 4.

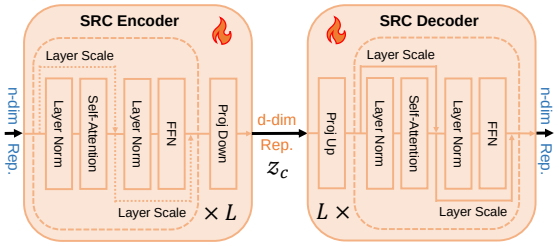


Figure 4: Semantic Representation Compressor (SRC). The encoder compresses RAE tokens from n to d channels, and the decoder restores the original representation dimension.

SRC training and compact representation. The SRC is trained before the normalizing flow while keeping both the RAE encoder E and decoder D frozen. Given an image x, we obtain $z = \text{norm}(E(x))$, reconstruct it as $\hat{z} = C_{\text{dec}}(C_{\text{enc}}(z))$, and decode $\hat{z}$ through the frozen RAE decoder. The SRC is optimized with the same reconstruction objective used for RAE decoder training, including pixel, perceptual, and adversarial losses. Following the RAE protocol, we apply noise augmentation to encoder outputs during SRC training so that the compact representation remains compatible with the noise-tolerant frozen decoder.

After SRC training, we freeze C_{enc} and C_{dec}. The normalizing flow is trained on the re-normalized compact semantic representation as follows:

$$z_c = \text{norm}_2(C_{\text{enc}}(z)), \qquad z_c \in \mathbb{R}^{N\times d}, \tag{5}$$

where $\text{norm}_2(\cdot)$ denotes channel-wise normalization using statistics computed over compact SRC features. The compact dimension d controls the trade-off between reconstruction fidelity and flow modeling difficulty. We use $d = 32$ by default, which preserves 99.06% PCA variance and gives the best full-scale generation performance in our ablation study.

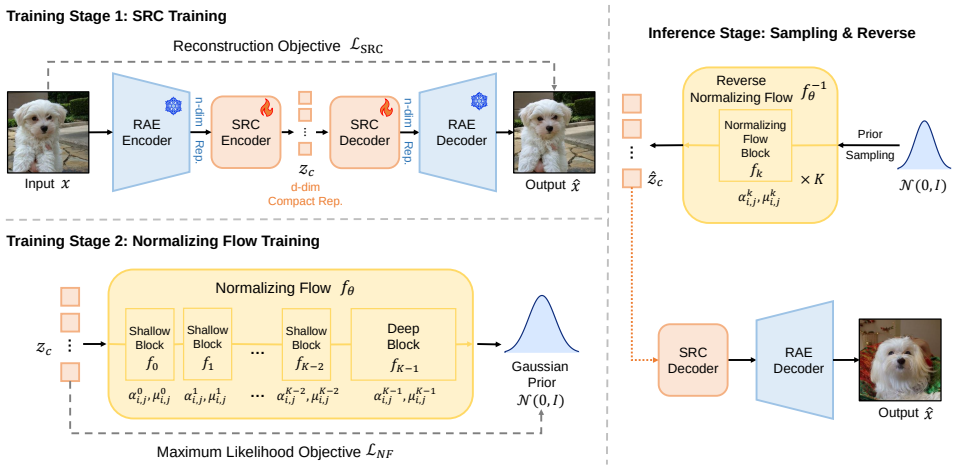


Figure 5: Overview of SRC-Flow. Stage 1 trains SRC with frozen RAE. Stage 2 trains an NF on compact semantic representations. Inference maps Gaussian samples through the inverse NF and decoders.

3.3 Noise Regularization

Gaussian noise is a standard regularization technique for normalizing flows (Dinh et al., 2017; Zhai et al., 2025): it smooths the empirical data distribution and helps the flow cover the Gaussian prior more uniformly. In SRC-Flow, we inject noise into RAE encoder outputs before normalization.

Table 2: Noise regularization results.

Schedule	w/o cfg	w/ cfg
Per-sample σ^i_{flow}	11.06	1.94
Constant $\sigma_{\text{flow}} = 0.4$	8.40	1.65

The key design choice is how the noise level is assigned. The RAE decoder training protocol samples a different perturbation strength for each example, i.e., $\sigma^i_{\text{flow}} \sim \mathcal{U}(0, \sigma_{\text{flow}})$, to improve decoder robustness. This is suboptimal for NFs: a flow learns a single unconditional bijection that is not conditioned on the perturbation level, so per-sample noise forces it to model a mixture of differently perturbed distributions with one fixed mapping. We instead use a constant noise level for all samples:

$$z_c = \text{norm}_2(C_{\text{enc}}(\text{norm}(E(x) + \epsilon_{\text{flow}}))), \qquad \epsilon_{\text{flow}} \sim \mathcal{N}(0, \sigma^2_{\text{flow}} I). \tag{6}$$

Constant noise substantially improves both unguided and guided generation (Table 2). We use $\sigma_{\text{flow}} = 0.4$, which performs best under the training schedule (Figure 8).

3.4 Overall Pipeline

SRC-Flow is trained in two stages, as illustrated in Figure 5. We denote the frozen RAE encoder and decoder as E and D, the SRC encoder and decoder as C_{enc} and C_{dec}, and the normalizing flow as f_θ.

Stage 1: SRC training. With E and D frozen, we train only the SRC using the RAE decoder objective (pixel, perceptual, and adversarial losses) and per-example noise augmentation, following the RAE decoder protocol:

$$\hat{x} = D(\text{denorm}\left(C_{\text{dec}}\left(C_{\text{enc}}\left(\text{norm}(E(x) + \epsilon_{\text{src}})\right)\right)\right)), \epsilon_{\text{src}} \sim \mathcal{N}(0, \sigma^2_{\text{src}} I),\ \sigma_{\text{src}} \sim \mathcal{U}(0, 0.8). \tag{7}$$

After this stage, both C_{enc} and C_{dec} are frozen.

Stage 2: Flow training. The flow maps the compact representation $z_c \in \mathbb{R}^{N \times d}$ in Eq. (6) to a Gaussian prior variable $u = f_\theta(z_c)$ and is trained by maximum likelihood:

$$\mathcal{L}_{\text{NF}} = \frac{1}{2}\|f_\theta(z_c)\|_2^2 + \sum_{k=0}^{K-1}\sum_{i=0}^{N-1}\sum_{j=0}^{d-1}\alpha^k_{i,j}, \tag{8}$$

where the first term is the negative Gaussian log-density up to a constant and the second is the negative log-determinant accumulated over all flow blocks.

Inference. Starting from $u \sim \mathcal{N}(0, I)$, we invert each stage in turn:

$$\hat{x} = D\left(\text{denorm}\left(C_{\text{dec}}\left(\text{denorm}_2\left(f_\theta^{-1}(u)\right)\right)\right)\right). \tag{9}$$

For class-conditional generation, we use the classifier-free guidance of STARFlow (Gu et al., 2025).

Table 3: Class-conditional image generation on ImageNet 256 × 256. SRC-Flow achieves the best result among normalizing flow methods. ‡SimFlow continued to 320 epochs with its official configuration (trained by us).

Method	Epochs	#Params	rFID↓	w/o guidance				w/ guidance			
				gFID↓	IS↑	Prec.↑	Rec.↑	gFID↓	IS↑	Prec.↑	Rec.↑
Pixel Space											
ADM (Dhariwal & Nichol, 2021)	400	554M	-	10.94	101.0	0.69	0.63	3.94	215.8	0.83	0.53
RIN (Jabri et al., 2023)	480	410M	-	3.42	182.0	-	-	-	-	-	-
PixelFlow (Chen et al., 2025)	320	677M	-	-	-	-	-	1.98	282.1	0.81	0.60
PixNerd (Wang et al., 2026a)	320	700M	-	-	-	-	-	1.93	298.0	0.80	0.60
SiD2 (Hoogeboom et al., 2025)	1280	-	-	-	-	-	-	1.38	-	-	-
TARFlow (Zhai et al., 2025)	320	1.4B	-	-	-	-	-	4.69	-	-	-
iTARFlow (Chen et al., 2026b)	320	2.2B	-	-	-	-	-	3.32	-	-	-
JetFormer (Tschannen et al., 2025b)	500	2.8B	-	-	-	-	-	6.64	-	0.69	0.56
FARMER (Zheng et al., 2026b)	320	1.9B	-	-	-	-	-	3.60	269.2	0.81	0.51
Latent Autoregressive											
VAR (Tian et al., 2024)	350	2.0B	-	1.92	323.1	0.82	0.59	1.73	350.2	0.82	0.60
MAR (Li et al., 2024)	800	943M	0.53	2.35	227.8	0.79	0.62	1.55	303.7	0.81	0.62
xAR (Ren et al., 2025)	800	1.1B	0.53	-	-	-	-	1.24	301.6	0.83	0.64
Latent Diffusion											
DiT (Peebles & Xie, 2023)	1400	675M	0.61	9.62	121.5	0.67	0.67	2.27	278.2	0.83	0.57
MaskDiT (Zheng et al., 2024)	1600	675M	0.61	5.69	177.9	0.74	0.60	2.28	276.6	0.80	0.61
SiT (Ma et al., 2024)	1400	675M	0.61	8.61	131.7	0.68	0.67	2.06	270.3	0.82	0.59
MDTv2 (Gao et al., 2023)	1080	675M	0.61	-	-	-	-	1.58	314.7	0.79	0.65
REPA (Yu et al., 2025)	800	675M	0.61	5.78	158.3	0.70	0.68	1.29	306.3	0.79	0.64
VA-VAE (Yao et al., 2025)	800	675M	0.28	2.17	205.6	0.77	0.65	1.35	295.3	0.79	0.65
DDT (Wang et al., 2026b)	400	675M	0.61	6.27	154.7	0.68	0.69	1.26	310.6	0.79	0.65
REPA-E (Leng et al., 2025)	800	675M	0.28	1.69	219.3	0.77	0.67	1.12	302.9	0.79	0.66
RAE (Zheng et al., 2026a)	800	839M	0.57	1.51	242.9	0.79	0.63	1.13	262.6	0.78	0.67
Latent Normalizing Flows											
BackFlow (Chen et al., 2026c)	320	1.4B	0.61	-	-	-	-	4.18	240.8	-	-
STARFlow (Gu et al., 2025)	320	1.4B	2.73	-	-	-	-	2.40	-	-	-
iTARFlow (Chen et al., 2026b)	320	770M	2.73	-	-	-	-	2.32	-	-	-
SimFlow (Zhao et al., 2025)	160	1.4B	1.08	10.13	124.7	0.71	0.61	1.91	284.4	0.82	0.60
SimFlow‡	320	1.4B	1.08	9.37	125.6	-	-	1.90	285.2	0.82	0.60
SRC-Flow (Ours)	160	1.4B	0.62	9.28	125.1	-	-	1.73	288.9	0.82	0.61
SRC-Flow (Ours)	320	1.4B	0.62	8.40	127.4	0.72	0.61	1.65	289.7	0.82	0.61

4 Experiments

4.1 Experimental Setup

Dataset, metrics, and baselines. We evaluate class-conditional generation on ImageNet (Deng et al., 2009) at 256 × 256 and 512 × 512. Generation quality is measured by gFID (Heusel et al., 2017), Inception Score (IS) (Salimans et al., 2016), Precision, and Recall (Kynkäänniemi et al., 2019), using 50k generated samples and ADM statistics (Dhariwal & Nichol, 2021). Reconstruction quality is measured by rFID on the ImageNet validation set. We compare SRC-Flow with pixel-space methods, latent autoregressive models, latent diffusion models, and latent normalizing flows. Among NF baselines, TARFlow (Zhai et al., 2025) operates in pixel space; BackFlow (Chen et al., 2026c) aligns reverse-pass features in a latent TARFlow variant; STARFlow (Gu et al., 2025), iTARFlow (Chen et al., 2026b), and SimFlow (Zhao et al., 2025) improve latent-space flow modeling through architectural, denoising, or training modifications.

Implementation details. The SRC uses compact dimension $d = 32$ and $L = 4$ Transformer layers. We train SRC for 16 epochs with batch size 256 using AdamW (Loshchilov & Hutter, 2019), learning rate 2×10^{-4}, cosine decay after 1 warm epoch. The flow follows the SimFlow (Zhao et al., 2025) architecture and is trained for 320 epochs with batch size 256, AdamW, learning rate 1×10^{-4}, cosine decay from epoch 160, EMA decay 0.9999, and constant noise $\sigma_{\text{flow}} = 0.4$.

4.2 Main Results

Results on ImageNet 256 × 256**.** Among normalizing flows, SRC-Flow achieves the best guided gFID of 1.65 (Table 3), outperforming pixel-space NFs such as TARFlow (4.69), FARMER (3.60),

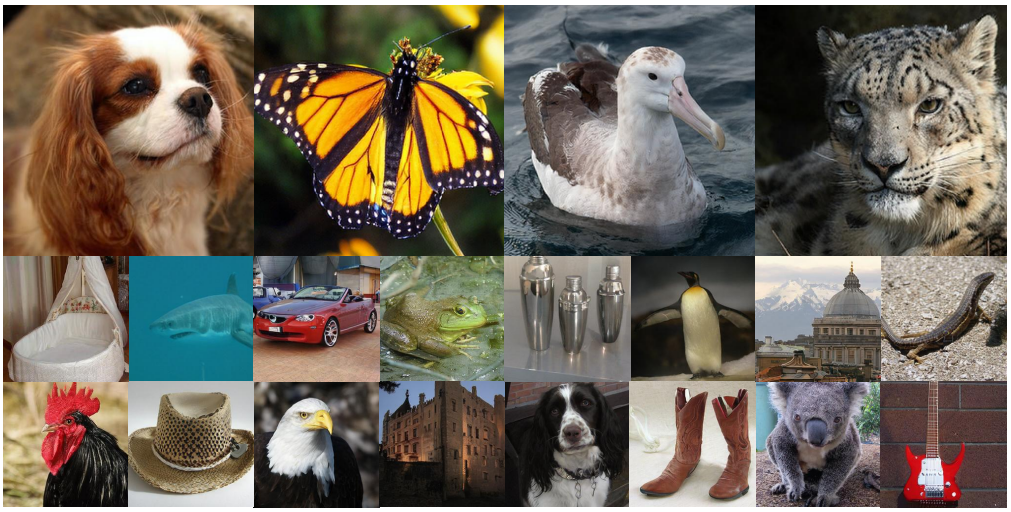


Figure 6: Class-conditional SRC-Flow samples on ImageNet: 512×512 (top row) and 256×256 (other rows).

and iTARFlow (3.32), and latent-space NFs such as BackFlow (4.18), STARFlow (2.40), latent iTARFlow (2.32), and SimFlow (1.91); without guidance, it improves the best NF result from 10.13 (SimFlow) to 8.40. Since SimFlow reports 160 epochs, we additionally continue training SimFlow to 320 epochs with its official configuration and report SRC-Flow at 160 epochs.

At matched budgets, SRC-Flow improves guided gFID from 1.91 to 1.73 at 160 epochs and from 1.90 to 1.65 at 320 epochs, whereas longer training leaves SimFlow essentially unchanged, and the gFID ranges over 10 independent 50k evaluation sets do not overlap (Appendix D.2).

Although SRC reduces the modeled channel dimension from n to 32, its rFID of 0.62 remains close to the 0.57 of the original RAE, indicating that SRC preserves decoder-relevant semantic information while substantially reducing the ambient dimensionality faced by the flow.

Table 4: ImageNet 512×512 w/ guidance. SRC-Flow achieves best result among normalizing flow methods.

Method	#Params	gFID↓	IS↑
BigGAN-deep (Brock et al., 2019)	158M	8.43	177.9
StyleGAN-XL (Sauer et al., 2022)	-	2.41	267.8
VAR (Tian et al., 2024)	2.3B	2.63	303.2
MAGVIT-v2 (Yu et al., 2024)	307M	1.91	324.3
MAR (Li et al., 2024)	481M	1.73	279.9
xAR (Ren et al., 2025)	608M	1.70	281.5
ADM (Dhariwal & Nichol, 2021)	731M	3.85	221.7
SiD2 (Hoogeboom et al., 2025)	-	1.50	-
DiT (Peebles & Xie, 2023)	674M	3.04	240.8
SiT (Ma et al., 2024)	674M	2.62	252.2
DiffiT (Hatamizadeh et al., 2024)	-	2.67	252.1
REPA (Yu et al., 2025)	675M	2.08	274.6
DDT (Wang et al., 2026b)	675M	1.28	305.1
EDM2 (Karras et al., 2024)	1.5B	1.25	-
RAE (Zheng et al., 2026a)	839M	1.13	259.6
STARFlow (Gu et al., 2025)	1.4B	3.00	-
SimFlow (Zhao et al., 2025)	1.4B	2.74	304.9
SRC-Flow	1.4B	2.07	305.7

Results on ImageNet 512×512. SRC-Flow achieves a gFID of 2.07 and an IS of 305.7 (Table 4), substantially outperforming STARFlow (3.00) and SimFlow (2.74), while SRC reconstruction remains close to the RAE tokenizer (rFID 0.54 vs. 0.53).

Qualitative samples on ImageNet 256×256 and 512×512. Figure 6 presents class-conditional samples generated by SRC-Flow on ImageNet at both resolutions with classifier-free guidance.

4.3 Analysis

Compact representation dimension. We first study the compact dimension d. As shown in Table 5, reconstruction fidelity improves as d increases, since larger compact representations preserve more decoder-relevant information. Semantic preservation follows the same trend: at $d = 32$, Top-5 accuracy of reconstructions remains 89.8% (vs. 96.6% for the full RAE), although Top-1 continues to improve with larger d, so not every fine-grained cue is preserved; DINO and CLIP similarities show the same saturation (Appendix D.5). However, generation quality is not monotonic: small d loses useful semantic and visual information, while large d increases the ambient dimensionality that the flow must model under exact likelihood. The best balance is at $d = 32$ (gFID 1.65, rFID 0.62 vs. 0.57 for the RAE), supporting our claim that the full RAE space is unnecessarily overcomplete for flow modeling; Figure 7 shows that its reconstructions remain visually close to the RAE.



Figure 7: Reconstructions across d. The $d = 32$ SRC preserves visual details close to the original RAE.

Table 5: Effect of compact dimension d on ImageNet 256×256. Top-1/Top-5 is the accuracy of a separately trained, frozen MLP classifier on reconstructed images, measuring semantic preservation. $d = 768$ denotes the full RAE representation. DINO and CLIP similarities are reported in Appendix D.5.

	Reconstruction & semantics		gFID↓	
d	rFID↓	Top-1/Top-5↑	w/o cfg	w/ cfg
4	1.13	7.3/19.6	15.94	3.53
8	0.75	20.7/43.2	7.23	2.12
16	0.64	47.3/70.4	7.84	1.69
32	0.62	69.1/89.8	8.40	1.65
64	0.60	74.7/93.8	9.78	1.97
128	0.60	78.9/95.1	10.65	2.21
768 (RAE)	0.57	82.6/96.6	11.53	3.58

Table 6: Compressor architecture. All variants compress RAE tokens to $d = 32$. Performance improves as token interaction becomes stronger.

Comp.	rFID↓	gFID↓	IS↑
PCA	1.08	3.31	263.8
Linear	0.94	2.86	274.5
Conv	0.70	2.14	282.8
Trans (SRC)	0.62	1.65	289.7

Table 7: SRC-Flow with different RAE encoders at $d = 16/32/64$, 160 epochs, $\sigma_{\text{flow}} = 0.4$.

Encoder	rFID↓	w/cfg gFID↓
DINOv2	0.64 / 0.62 / 0.60	1.82 / **1.73** / 2.01
SigLIP2	0.97 / 0.91 / 0.87	2.12 / **2.05** / 2.32
MAE	0.58 / 0.43 / 0.36	3.72 / **3.36** / 3.37

Compressor architecture and depth. Table 6 compares compression designs. Non-learned PCA already improves on the full RAE (3.31 vs. 3.58) without training a new tokenizer, but it is not optimized for the frozen decoder or the flow. Learnable compressors perform better, and stronger token interaction helps further: token-wise linear projection is weaker than local convolution, and the Transformer-based SRC performs best by aggregating information across spatial tokens with global self-attention. Semantic compression thus benefits from modeling global token dependencies rather than being mere dimensionality reduction. A noise-robust VA-VAE trained with the same NF and schedule reaches gFID 1.89, still behind SRC (Appendix D.3). For depth (Table 8), increasing L from 1 to 4 improves both reconstruction and generation, while deeper compressors only slightly improve rFID with negligible generation gains; we use $L = 4$ by default.

Noise regularization. Training the flow without noise converges poorly (Figure 8), confirming that noise regularization is essential in compact semantic space, and $\sigma_{\text{flow}} = 0.4$ performs best. In terms of reconstruction robustness, the $d = 32$ SRC stays close to the full RAE at low noise and even achieves lower rFID under strong perturbations, suggesting that compression suppresses redundant, noise-sensitive directions and acts as an implicit denoising projection.

Transfer across pretrained encoders. To test whether the compact regime is specific to DINOv2, we train SRC and the same flow on the released SigLIP2- (Tschannen et al., 2025a) and MAE-based (He et al., 2022) RAEs (Zheng et al., 2026a), with a matched 160-epoch schedule and $\sigma_{\text{flow}} = 0.4$ without retuning (Table 7). All three encoders show a compact optimum or saturation near $d = 32$: increasing to $d = 64$ improves rFID but not generation. Notably, MAE reconstructs best yet generates worst, again showing that reconstruction fidelity alone does not determine NF tractability. Absolute quality and the precise optimum remain encoder-dependent, so we do not claim that $d = 32$ is universal (full results in Appendix D.7).

5 Related Work

Normalizing Flows for Image Generation. Normalizing flows are exact-likelihood generative models built from tractable invertible transformations. Coupling flows such as NICE (Dinh et al.,

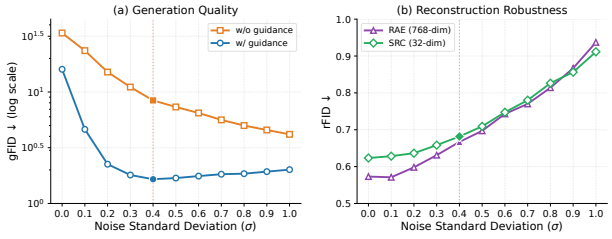


Figure 8: Noise regularization. $\sigma_{\text{flow}} = 0.4$ gives the best gFID, and the $d = 32$ SRC improves high-noise robustness.

Table 8: SRC depth. Increasing depth improves SRC quality up to $L = 4$. Deeper compressors slightly improve rFID but bring negligible gains in gFID and IS.

L	rFID↓	gFID↓	IS↑
1	0.77	2.34	282.6
2	0.68	1.87	291.4
4	0.62	1.65	289.7
8	0.60	1.65	289.1
16	0.59	1.66	289.6

2015), RealNVP (Dinh et al., 2017), and Glow (Kingma & Dhariwal, 2018) enabled likelihood modeling with triangular Jacobians, and autoregressive (Papamakarios et al., 2017; Kingma et al., 2016), continuous (Chen et al., 2018; Grathwohl et al., 2019), residual (Behrmann et al., 2019; Chen et al., 2019), and spline flows (Durkan et al., 2019), as well as Flow++ (Ho et al., 2019), further improved expressiveness and training. Nevertheless, NFs were largely surpassed in image synthesis by diffusion (Ho et al., 2020; Song et al., 2021b;a; Karras et al., 2022) and autoregressive models (van den Oord et al., 2016; Esser et al., 2021; Ramesh et al., 2021). Transformer-based NFs have recently renewed this direction: TARFlow (Zhai et al., 2025) generates images directly in pixel space with masked autoregressive flows, STARFlow (Gu et al., 2025) moves NFs to VAE latent space with a deep-shallow architecture, SimFlow (Zhao et al., 2025) jointly optimizes the autoencoder and flow, and iTARFlow (Chen et al., 2026b) addresses the noise dilemma with multi-noise training and iterative denoising. Related systems include JetFormer (Tschannen et al., 2025b), FARMER (Zheng et al., 2026b), and BackFlow (Chen et al., 2026c), which aligns reverse-pass NF features with pretrained visual representations. In contrast, SRC-Flow focuses on the modeling space itself, placing normalizing flows in a compact semantic representation space.

Latent Spaces for Generative Models. Latent Diffusion Models (Rombach et al., 2022) popularized the two-stage VAE-based paradigm (Kingma & Welling, 2014), now widely adopted in high-resolution systems (Podell et al., 2024; Ramesh et al., 2022; Esser et al., 2024). Diffusion Transformers (Peebles & Xie, 2023) and interpolant or rectified-flow models (Lipman et al., 2023; Liu et al., 2023; Ma et al., 2024) further improved scalability, although these "flow" objectives generally do not provide exact likelihood through a change-of-variables formula. Latent generation has also been improved by representation alignment (Yu et al., 2025), VAE optimization (Yao et al., 2025), equivariant regularization (Kouzelis et al., 2025), end-to-end VAE tuning (Leng et al., 2025), and decoupled diffusion architectures (Wang et al., 2026b). Representation Autoencoders (RAEs) (Zheng et al., 2026a) replace VAE encoders with pretrained representation models such as DINOv2 (Oquab et al., 2024), SigLIP (Tschannen et al., 2025a), and other self-supervised or language-supervised encoders (Dosovitskiy et al., 2021; Caron et al., 2021; He et al., 2022; Radford et al., 2021), whose features are semantically rich but high-dimensional and overcomplete. AlignTok (Chen et al., 2026a), FlatDINO (Calvo-González & Fleuret, 2026), LV-RAE (Liu et al., 2026), and PS-VAE (Zhang et al., 2026) adapt such features for reconstruction or diffusion-based generation, whereas SRC-Flow studies how they should be structured for exact-likelihood NFs.

6 Conclusion

We presented SRC-Flow, which identifies a semantic-capacity mismatch between the compact semantic information in pretrained representations and the full ambient space that NFs must model through a single exact invertible transport. Our Semantic Representation Compressor (SRC), together with constant noise regularization, reduces this modeling burden and achieves state-of-the-art generation quality among NF methods, with exact likelihood in the compact semantic representation space and deterministic invertible sampling at the flow level. **Limitations and future work.** Exact likelihood holds for the compact representation rather than for images, since the SRC and RAE decoders are not invertible. Our generation experiments are class-conditional on ImageNet, so fine-grained text-conditioned control remains untested. A gap also remains to the strongest diffusion and autoregressive models, the autoregressive flow limits inference throughput, and the frozen RAE decoder upper-bounds reconstruction fidelity. Future work includes parallel or non-autoregressive flows, stronger semantic compression objectives, and text-conditional or higher-resolution generation.

AI USE STATEMENT

We used large language model (LLM) assistants to aid and polish the writing of this paper, improving its clarity, readability, and overall coherence. During the preparation of this manuscript, we used ChatGPT to assist with language polishing and editing, specifically within the Abstract, Introduction, and Method sections. The primary goal of using these tools was to enhance the flow, readability, and coherence of the text. We confirm that all mathematical formulations, theoretical derivations, and principle expositions were written entirely by authors. Furthermore, all experimental designs, methodology proposals, and code execution were conducted manually by the authors without the use of generative AI, and all reported numbers come from our own experiments. The authors take full responsibility for the content and integrity of this publication.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A specify the model and the exact maximum-likelihood objective. Appendix B describes representation extraction, normalization, the SRC architecture, and both training stages, with all architectural and optimization hyperparameters listed in Tables 9 and 10. Evaluation follows the standard 50k-sample ImageNet protocol with ADM reference statistics (Section 4.1); Appendix D.2 reports per-seed results and the exact protocol used to quantify evaluation variance, and Appendices D.4–D.8 describe the likelihood, semantic-preservation, diversity, cross-encoder, and COCO evaluations. Stage-wise training cost and sampling throughput are reported in Appendices D.9 and D.10. All datasets and pretrained RAE checkpoints used in this work are publicly available, and we will release code and trained models upon publication.

ETHICS STATEMENT

This work studies generative modeling on public benchmarks (ImageNet and COCO) and does not involve human subjects, crowdsourcing, or the collection of new data. As with other image generators, improvements in generation quality could be misused to produce misleading content, although the risk is limited here because our models are class-conditional on ImageNet categories and have no text-conditioning interface. Generated samples may also inherit biases present in the training data, such as uneven coverage of object categories, scenes, and regions, and should not be treated as representative of the real world. Any release of models will be accompanied by a model card documenting intended use, limitations, and usage terms. We use existing datasets and pretrained models in accordance with their licenses and cite their original sources.

REFERENCES

Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *ICML*, 2019.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

Ramón Calvo-González and François Fleuret. Laminating representation autoencoders for efficient diffusion. *arXiv preprint arXiv:2602.04873*, 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Bowei Chen, Sai Bi, Hao Tan, He Zhang, Tianyuan Zhang, Zhengqi Li, Yuanjun Xiong, Jianming Zhang, and Kai Zhang. Aligning visual foundation encoders to tokenizers for diffusion models. In *ICLR*, 2026a.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

Ricky T. Q. Chen, Jens Behrmann, David K. Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. In *NeurIPS*, 2019.

Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. PixelFlow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025.

Tianrong Chen, Jiatao Gu, David Berthelot, Joshua Susskind, and Shuangfei Zhai. Normalizing flows with iterative denoising. In *ICML*, 2026b.

Yang Chen, Xiaowei Xu, Shuai Wang, Chenhui Zhu, Ruxue Wen, Xubin Li, Tiezheng Ge, and Limin Wang. Flowing backwards: Improving normalizing flows via reverse representation alignment. In *AAAI*, 2026c.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, 2021.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. In *ICLR Workshop*, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *NeurIPS*, 2019.

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

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. MDTv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *ICLR*, 2019.

Jiatao Gu, Tianrong Chen, David Berthelot, Huangjie Zheng, Yuyang Wang, Ruixiang Zhang, Laurent Dinh, Miguel Angel Bautista, Josh Susskind, and Shuangfei Zhai. STARFlow: Scaling latent normalizing flows for high-resolution image synthesis. In *NeurIPS*, 2025.

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. DiffiT: Diffusion vision transformers for image generation. In *ECCV*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

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

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *ICML*, 2019.

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

Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 FID on ImageNet512 with pixel-space diffusion. In *CVPR*, 2025.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *ICML*, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

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

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1×1 convolutions. In *NeurIPS*, 2018.

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

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *NeurIPS*, 2016.

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. EQ-VAE: Equivariance regularized latent space for improved generative image modeling. In *ICML*, 2025.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019.

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

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *NeurIPS*, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Siyu Liu, Chujie Qin, Hubery Yin, Qixin Yan, Zheng-Peng Duan, Chen Li, Jing Lyu, Chun-Le Guo, and Chongyi Li. Improving reconstruction of representation autoencoder. *arXiv preprint arXiv:2602.08620*, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *NeurIPS*, 2017.

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

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 *ICLR*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-X prediction for autoregressive visual generation. In *ICCV*, 2025.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

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

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

Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *SIGGRAPH*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.

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

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024.

Michael Tschannen, Alexey A. Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier J. Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025a.

Michael Tschannen, André Susano Pinto, and Alexander Kolesnikov. JetFormer: An autoregressive generative model of raw images and text. In *ICLR*, 2025b.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion. In *ICLR*, 2026a.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. In *CVPR*, 2026b.

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

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation. In *ICLR*, 2024.

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

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. In *ICML*, 2025.

Shilong Zhang, He Zhang, Zhifei Zhang, Chongjian Ge, Shuchen Xue, Shaoteng Liu, Mengwei Ren, Soo Ye Kim, Yuqian Zhou, Qing Liu, Daniil Pakhomov, Kai Zhang, Zhe Lin, and Ping Luo. Both semantics and reconstruction matter: Making representation encoders ready for text-to-image generation and editing. In *ICML*, 2026.

Qinyu Zhao, Guangting Zheng, Tao Yang, Rui Zhu, Xingjian Leng, Stephen Gould, and Liang Zheng. SimFlow: Simplified and end-to-end training of latent normalizing flows. *arXiv preprint arXiv:2512.04084*, 2025.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026a.

Guangting Zheng, Qinyu Zhao, Tao Yang, Fei Xiao, Zhijie Lin, Jie Wu, Jiajun Deng, Yanyong Zhang, and Rui Zhu. FARMER: Flow autoregressive transformer over pixels. In *CVPR*, 2026b.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

A Derivation of the SRC-Flow Objective

This section derives the maximum likelihood objective used for training SRC-Flow. We first review the change-of-variables theorem, then instantiate it with a composed normalizing flow and the autoregressive affine transformation used in our compact semantic representation space.

A.1 Change of Variables

We begin with the one-dimensional case. Let u be a random variable with a known density $\pi(u)$, and let y be another random variable related to u by an invertible transformation,

$$u = f(y), \qquad y = f^{-1}(u). \tag{10}$$

The density of y must preserve probability mass. For an infinitesimal interval, we have,

$$p_{\mathcal{Y}}(y)\, dy = \pi(u)\, du. \tag{11}$$

Since $u = f(y)$, this gives,

$$p_{\mathcal{Y}}(y) = \pi(f(y)) \left| \frac{df(y)}{dy} \right|. \tag{12}$$

Intuitively, the derivative term accounts for how the transformation locally expands or contracts volume. The multivariate case follows the same principle. Let $y \in \mathbb{R}^m$ and $u = f(y) \in \mathbb{R}^m$, where f is invertible and differentiable. The probability density transforms as,

$$p_{\mathcal{Y}}(y) = p_{\mathcal{U}}(f(y)) \left| \det \frac{\partial f(y)}{\partial y} \right|. \tag{13}$$

Taking the logarithm yields the standard change-of-variables formula,

$$\log p_{\mathcal{Y}}(y) = \log p_{\mathcal{U}}(f(y)) + \log \left| \det \frac{\partial f(y)}{\partial y} \right|. \tag{14}$$

This is the basic likelihood identity used by normalizing flows.

A.2 Normalizing Flow Composition

A normalizing flow constructs the transformation f_θ as a composition of invertible mappings. Let $h^0 = y$ denote the data variable and $h^K = u$ denote the prior variable. The flow is defined as,

$$h^{k+1} = f_k(h^k), \qquad k = 0, \ldots, K-1, \tag{15}$$

so that,

$$u = f_\theta(y) = f_{K-1} \circ f_{K-2} \circ \cdots \circ f_0(y). \tag{16}$$

Applying Eq. (14) recursively gives,

$$\log p_{\mathcal{Y}}(y) = \log p_{\mathcal{U}}(u) + \sum_{k=0}^{K-1} \log \left| \det \frac{\partial h^{k+1}}{\partial h^k} \right|. \tag{17}$$

Thus, likelihood computation is tractable when each transformation is invertible and its Jacobian determinant is easy to evaluate.

For generation, we sample from the prior and apply the inverse transformations in reverse order,

$$y = f_\theta^{-1}(u) = f_0^{-1} \circ f_1^{-1} \circ \cdots \circ f_{K-1}^{-1}(u), \qquad u \sim \mathcal{N}(0, I). \tag{18}$$

A.3 Compact Semantic Representation

In SRC-Flow, the modeled variable is not the pixel image or the full RAE representation, but the compact semantic representation. Given an image x, we obtain,

$$z_c = \mathrm{norm}_2\left(C_{\mathrm{enc}}\left(\mathrm{norm}(E(x) + \epsilon_{\mathrm{flow}})\right)\right), \qquad \epsilon_{\mathrm{flow}} \sim \mathcal{N}(0, \sigma_{\mathrm{flow}}^2 I). \tag{19}$$

Here, E is the frozen RAE encoder, C_{enc} is the SRC encoder, $\mathrm{norm}(\cdot)$ is the RAE feature normalization, and $\mathrm{norm}_2(\cdot)$ is the second-stage normalization computed over compact SRC features. The

compact representation satisfies $z_c \in \mathbb{R}^{N\times d}$, where N is the number of spatial tokens and d is the compact channel dimension. In our main experiments, $d = 32$ and $\sigma_{\text{flow}} = 0.4$.

The normalizing flow maps this compact representation to a standard Gaussian prior,

$$u = f_\theta(z_c), \qquad p_{\mathcal{U}}(u) = \mathcal{N}(0, I). \tag{20}$$

Applying Eq. (17) with $y = z_c$ gives,

$$\log p_{\mathcal{Z}_c}(z_c) = \log p_{\mathcal{U}}(f_\theta(z_c)) + \sum_{k=0}^{K-1} \log \left| \det \frac{\partial h^{k+1}}{\partial h^k} \right|. \tag{21}$$

Since the prior is a standard Gaussian over Nd dimensions, we have,

$$\log p_{\mathcal{U}}(f_\theta(z_c)) = -\frac{1}{2} \left\| f_\theta(z_c) \right\|_2^2 - \frac{Nd}{2} \log(2\pi). \tag{22}$$

The second term is constant with respect to model parameters and is omitted during optimization.

A.4 Autoregressive Affine Flow

We instantiate each flow block f_k as an autoregressive affine transformation. Let $h_i^k \in \mathbb{R}^d$ denote the i-th token at block k. A causal Transformer predicts the shift and log-scale parameters from previous tokens and an optional class condition c,

$$\mu_i^k, \alpha_i^k = g_\theta^k(h_{<i}^k, c). \tag{23}$$

The forward data-to-prior transformation and the inverse prior-to-data transformation are,

$$h_i^{k+1} = (h_i^k - \mu_i^k) \odot \exp(-\alpha_i^k), \qquad h_i^k = h_i^{k+1} \odot \exp(\alpha_i^k) + \mu_i^k. \tag{24}$$

This matches the affine transformation used in the main paper.

Because μ_i^k and α_i^k depend only on previous tokens $h_{<i}^k$, the Jacobian of the transformation is triangular under the autoregressive ordering. For token i and channel j, the diagonal derivative is,

$$\frac{\partial h_{i,j}^{k+1}}{\partial h_{i,j}^k} = \exp(-\alpha_{i,j}^k). \tag{25}$$

Therefore, the log-determinant of block k is,

$$\log \left| \det \frac{\partial h^{k+1}}{\partial h^k} \right| = -\sum_{i=0}^{N-1} \sum_{j=0}^{d-1} \alpha_{i,j}^k. \tag{26}$$

Summing over all flow blocks gives,

$$\sum_{k=0}^{K-1} \log \left| \det \frac{\partial h^{k+1}}{\partial h^k} \right| = -\sum_{k=0}^{K-1} \sum_{i=0}^{N-1} \sum_{j=0}^{d-1} \alpha_{i,j}^k. \tag{27}$$

A.5 SRC-Flow Negative Log-Likelihood

Substituting Eq. (22) and Eq. (27) into Eq. (21), we obtain,

$$\log p_{\mathcal{Z}_c}(z_c) = -\frac{1}{2} \left\| f_\theta(z_c) \right\|_2^2 - \sum_{k=0}^{K-1} \sum_{i=0}^{N-1} \sum_{j=0}^{d-1} \alpha_{i,j}^k - \frac{Nd}{2} \log(2\pi). \tag{28}$$

Thus, minimizing the negative log-likelihood is equivalent to optimizing,

$$\mathcal{L}_{\text{NF}} = \frac{1}{2} \left\| f_\theta(z_c) \right\|_2^2 + \sum_{k=0}^{K-1} \sum_{i=0}^{N-1} \sum_{j=0}^{d-1} \alpha_{i,j}^k, \tag{29}$$

where the constant Gaussian normalization term is omitted.

For a training dataset $\mathcal{D}$, the empirical objective is,

$$\mathcal{L}_{\text{NF}}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{x\in\mathcal{D}} \left[\frac{1}{2} \left\| f_\theta(z_c(x)) \right\|_2^2 + \sum_{k=0}^{K-1} \sum_{i=0}^{N-1} \sum_{j=0}^{d-1} \alpha_{i,j}^k \right]. \tag{30}$$

This objective is the exact negative log-likelihood in the compact semantic representation space.

A.6 GENERATION

During generation, we sample $u \sim \mathcal{N}(0, I)$ and apply the inverse flow f_θ^{-1} autoregressively over tokens. After obtaining the compact semantic sample, we invert the second-stage normalization, decode it through the SRC decoder, invert the first-stage normalization, and reconstruct the image with the frozen RAE decoder,

$$\hat{x} = D\left(\text{denorm}\left(C_{\text{dec}}\left(\text{denorm}_2\left(f_\theta^{-1}(u)\right)\right)\right)\right). \tag{31}$$

The likelihood derived above is exact for z_c, because the normalizing flow transformation f_θ is bijective in the compact semantic representation space.

B ADDITIONAL IMPLEMENTATION DETAILS

This section provides additional implementation details for SRC-Flow, including representation extraction, normalization, SRC architecture, training objectives, normalizing flow training, and classifier-free guidance.

B.1 REPRESENTATION EXTRACTION AND NORMALIZATION

Given an input image x, the frozen RAE encoder E produces raw semantic tokens,

$$z_{\text{raw}} = E(x) \in \mathbb{R}^{N \times n}, \tag{32}$$

where N is the number of spatial tokens and n is the original RAE channel dimension. We apply channel-wise normalization using statistics $(\mu_{rae}, \sigma_{rae})$ precomputed over the ImageNet training set,

$$z = \text{norm}(z_{\text{raw}}) = \frac{z_{\text{raw}} - \mu_{rae}}{\sigma_{rae}}. \tag{33}$$

This normalization approximately standardizes each RAE feature channel and produces the semantic representation z used by the SRC.

After the SRC encoder is trained and frozen, we compute a set of channel-wise statistics over compact SRC features on training set. The compact representation modeled by the normalizing flow is,

$$z_c = \text{norm}_2\left(C_{\text{enc}}(z)\right), \qquad z_c \in \mathbb{R}^{N \times d}. \tag{34}$$

The second-stage normalization standardizes the compact semantic representation and stabilizes flow training. During flow training, noise is injected before the first normalization.

During inference, the inverse operations are applied in reverse order. Given a Gaussian sample $u \sim \mathcal{N}(0, I)$, the generated image is obtained through,

$$f_\theta^{-1}(u) \xrightarrow{\text{denorm}_2} \tilde{z}_c \xrightarrow{C_{\text{dec}}} \hat{z} \xrightarrow{\text{denorm}} \hat{z}_{\text{raw}} \xrightarrow{D} \hat{x}. \tag{35}$$

Here, $\text{denorm}_2(\cdot)$ and inverts the compact-space normalization, and $\text{denorm}(\cdot)$ inverts the original RAE feature normalization, respectively.

B.2 SEMANTIC REPRESENTATION COMPRESSOR

The Semantic Representation Compressor (SRC) consists of an encoder C_{enc} and a decoder C_{dec}. The encoder maps the normalized RAE representation to a compact semantic representation,

$$C_{\text{enc}} : \mathbb{R}^{N \times n} \to \mathbb{R}^{N \times d}, \tag{36}$$

while the decoder maps the compact representation back to the original RAE channel dimension,

$$C_{\text{dec}} : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times n}. \tag{37}$$

The SRC encoder applies L Transformer blocks followed by a projection layer from n channels to d channels. Our normalizing flow model is trained in this space. The decoder mirrors this structure: a projection layer first maps the compact representation from d channels back to n channels, followed by L Transformer blocks. The encoder and decoder do not share weights.

Each Transformer block contains LayerNorm, multi-head self-attention, a feed-forward network with GELU activation, and LayerScale for training stability. The projection layers perform channel compression and expansion at each spatial token, while Transformer self-attention enables global token interaction before compression or after expansion. Unless otherwise specified, we use $d = 32$ and $L = 4$. The default SRC configuration is summarized in Table 9.

Table 9: Default SRC configuration. We use a symmetric Transformer-based encoder-decoder compressor.

Component	Setting
Input representation	Normalized RAE tokens $z \in \mathbb{R}^{N\times n}$
Compact representation	$z_c \in \mathbb{R}^{N\times d}$
Compact dimension	$d = 32$
Transformer blocks	$L = 4$ encoder blocks and $L = 4$ decoder blocks
Channel projection	1×1 projection
Attention	Multi-head self-attention
Normalization	LayerNorm
Activation	GELU
Stabilization	LayerScale
Weight sharing	No encoder-decoder sharing
Frozen modules	RAE encoder and RAE decoder

B.3 SRC Training

During the SRC training stage, both the RAE encoder E and decoder D are frozen, and only C_{enc} and C_{dec} are optimized. Given an image x, the reconstruction path is,

$$\hat{x} = D\left(\text{denorm}\left(C_{\text{dec}}\left(C_{\text{enc}}\left(\text{norm}(E(x) + \epsilon_{\text{src}})\right)\right)\right)\right), \qquad \epsilon_{\text{src}} \sim \mathcal{N}(0, \sigma_{\text{src}}^2 I). \tag{38}$$

Following the RAE decoder training protocol, the SRC training noise level is sampled independently for each example in the training batch,

$$\sigma_{\text{src}} \sim \mathcal{U}(0, 0.8). \tag{39}$$

This exposes the SRC decoder to perturbed representations and helps maintain compatibility with the frozen noise-tolerant RAE decoder.

The SRC is optimized using the same reconstruction objective as the RAE decoder,

$$\mathcal{L}_{\text{SRC}} = \mathcal{L}_1(x, \hat{x}) + \lambda_{\text{p}}\mathcal{L}_{\text{LPIPS}}(x, \hat{x}) + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}(x, \hat{x}). \tag{40}$$

After SRC training, both C_{enc} and C_{dec} are frozen for all flow training and inference experiments.

B.4 Normalizing Flow Training

We train the normalizing flow on compact semantic representations produced by the frozen SRC encoder. The flow follows the Transformer Autoregressive Flow architecture used in recent latent NF models. It contains $K = 6$ autoregressive affine blocks and adopts the deep-shallow design: the first $K - 1$ blocks are shallow, while the final block closest to the Gaussian prior is deep. For the i-th token in block k, a causal Transformer predicts the affine parameters (μ_i^k, α_i^k) from preceding tokens and the optional class label.

During training, the compact semantic representation is constructed with constant Gaussian noise,

$$z_c = \text{norm}_2\left(C_{\text{enc}}\left(\text{norm}(E(x) + \epsilon_{\text{flow}})\right)\right), \qquad \epsilon_{\text{flow}} \sim \mathcal{N}(0, \sigma_{\text{flow}}^2 I). \tag{41}$$

Unlike SRC training, where the noise level is sampled per example, flow training uses a fixed standard deviation $\sigma_{\text{flow}} = 0.4$ for all samples. This matches the fixed unconditional bijection learned by the normalizing flow and avoids training on a mixture of differently perturbed semantic distributions.

The flow is trained by minimizing the exact negative log-likelihood derived in Section A. The key training hyperparameters for SRC and flow training are summarized in Table 10.

B.5 Classifier-Free Guidance

For class-conditional generation, we adopt classifier-free guidance for autoregressive flows following the score-based formulation used in STARFlow. During training, the class label is independently dropped with probability p_{drop} for each sample, allowing the model to learn both conditional and unconditional predictions. At inference time, conditional and unconditional predictions are combined using a guidance weight ω.

Table 10: **Training hyperparameters.** We train SRC-Flow in two stages. Stage 1 trains only the SRC with the frozen RAE encoder and decoder, while Stage 2 freezes the SRC and trains the normalizing flow on compact semantic representations.

Setting	Stage 1: SRC training	Stage 2: Flow training
Trainable modules	$C_{\text{enc}}, C_{\text{dec}}$	f_θ
Frozen modules	E, D	E, D, C_{enc}, C_{dec}
Compact feature shape	$32 \times \frac{H}{16} \times \frac{W}{16}$	$32 \times \frac{H}{16} \times \frac{W}{16}$
Epochs	16	320
Batch size	256	256
Optimizer	AdamW	AdamW
Learning rate	2×10^{-4}	1×10^{-4}
Final learning rate	2×10^{-5}	1×10^{-6}
AdamW β	$(0.9, 0.95)$	$(0.9, 0.999)$
Weight decay	0.0	0.0
LR schedule	Cosine, 1 epoch warmup	Cosine, decay from epoch 160
Noise strategy	Per-example RAE noise	Constant flow noise
Noise level	$\sigma_{\text{src}} \sim \mathcal{U}(0, 0.8)$	$\sigma_{\text{flow}} = 0.4$
EMA decay	0.9978	0.9999
Class drop probability	–	0.1

Table 11: **Model scale ablation.** We vary only the normalizing flow size while keeping the RAE, SRC, compact dimension $d = 32$, and training schedule fixed. The default setting is highlighted in gray.

Model	#Params	Epochs	gFID↓ w/o cfg	IS↑ w/o cfg	gFID↓ w/ cfg	IS↑ w/ cfg
S	37M	320	42.8	62.4	10.6	168.5
B	141M	320	26.7	81.3	6.42	207.8
L	475M	320	15.9	105.6	3.18	257.4
XL	695M	320	11.7	117.9	2.28	275.6
XXL	1.4B	320	8.40	127.4	1.65	289.7

The main results use the guidance weight that gives the best gFID under the validation protocol. We provide an ablation over ω in Section C.2. In general, increasing ω improves class fidelity and perceptual quality up to a moderate range, after which gFID saturates or degrades slightly while Inception Score may continue increasing.

C Additional Ablations

C.1 Model Scale

We study the effect of normalizing flow scale while keeping the representation space fixed. Specifically, the frozen RAE encoder, frozen RAE decoder, SRC architecture, compact dimension $d = 32$, and noise level $\sigma_{\text{flow}} = 0.4$ are shared across all variants. Thus, changing the model scale only changes the capacity of the normalizing flow f_θ, while reconstruction quality remains determined by the fixed SRC, RAE encoder and decoder.

Table 11 reports generation quality for different flow sizes after training for 320 epochs. Larger flows consistently improve both unguided and guided generation quality, indicating that compact semantic representations remain capacity-scalable for normalizing flows. The gains are substantial from S to L, and then gradually saturate from XL to XXL. We use the XXL model as the default setting.

C.2 Guidance Weight

We study the effect of classifier-free guidance weight ω on generation quality. Figure 9 shows gFID and Inception Score (IS) under different classifier-free guidance weights.

When $\omega = 0$, the model reduces to unguided sampling and yields substantially worse generation quality. As ω increases from 0 to around 1.0, gFID decreases rapidly while IS improves consistently,

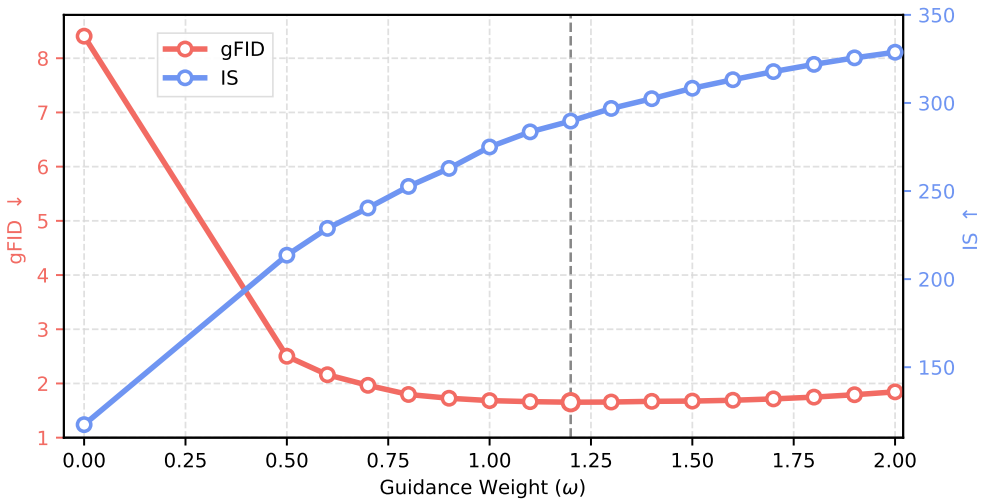


Figure 9: Effect of classifier-free guidance weight ω on generation quality. Increasing ω substantially improves both gFID and IS at first. The best gFID is achieved at $\omega = 1.2$, while IS continues to increase.

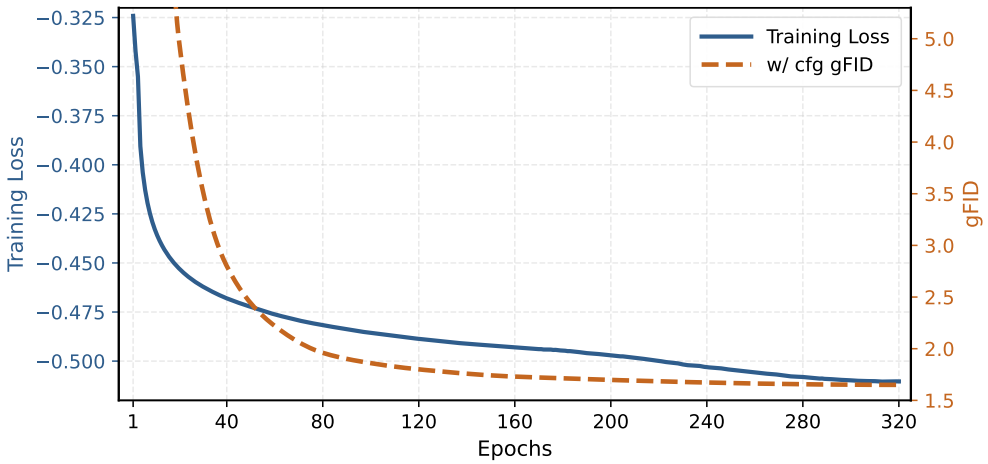


Figure 10: Training dynamics of SRC-Flow over 320 epochs. The training loss decreases steadily, while gFID under classifier-free guidance improves consistently and reaches 1.65 at epoch 320.

showing that classifier-free guidance greatly improves both fidelity and class consistency. The best gFID is achieved at $\omega = 1.2$, where SRC-Flow reaches 1.65 gFID with an IS of 289.7. Beyond this point, gFID saturates and slightly degrades, while IS continues to increase. This indicates the usual trade-off of classifier-free guidance: stronger guidance improves class confidence and sample sharpness, but overly large guidance starts to hurt distributional fidelity. Therefore, we use the guidance weight that gives the best validation gFID in the main experiments.

These results are consistent with prior observations in class-conditional generative models, where moderate guidance provides the best balance between fidelity and diversity. In our setting, $\omega \in [1.1, 1.3]$ forms a stable high-performance range.

C.3 Training Dynamics

We further analyze the training dynamics of SRC-Flow over the full 320-epoch schedule. Figure 10 shows the training loss together with the validation gFID measured at different training epochs.

The training loss decreases rapidly in the early stage and then gradually stabilizes, indicating that optimization in the compact semantic representation space is stable throughout training. Meanwhile, gFID improves consistently as training proceeds: it drops quickly in the first stage, and then continues to improve more gradually at later epochs, eventually reaching 1.65 at epoch 320. This trend suggests that most coarse distribution alignment is learned early, while later training mainly refines sample quality and semantic fidelity.

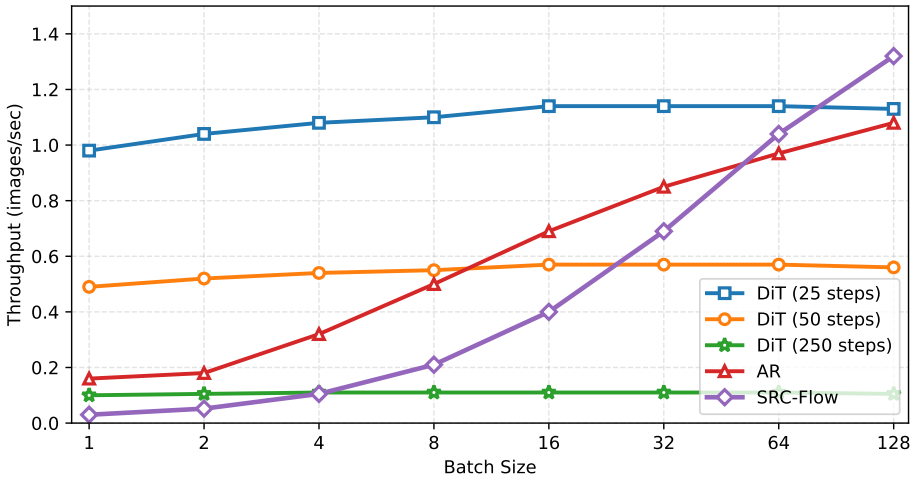


Figure 11: Inference throughput as a function of batch size. SRC-Flow follows the autoregressive-flow scaling trend: it is slower at small batch sizes but benefits substantially from larger-batch sampling.

Overall, the smooth decrease in both training loss and gFID indicates that SRC-Flow scales well with longer training and does not exhibit strong optimization instability under the 320-epoch setting.

C.4 Inference Speed

We analyze the inference throughput of SRC-Flow as a function of batch size. Since SRC-Flow adopts an autoregressive flow architecture, inverse sampling is sequential over tokens and is less efficient at very small batch sizes. However, the computation can be better amortized as batch size increases, leading to improved throughput at larger batches.

Figure 11 compares the throughput of SRC-Flow with representative diffusion and autoregressive baselines. SRC-Flow is slower than few-step diffusion models at small batch sizes, but its throughput improves substantially with larger batches. This behavior is consistent with autoregressive flow sampling: sequential token generation limits low-batch efficiency, while compact semantic representation modeling and batch-level parallelism improve large-batch throughput.

C.5 Compact Dimension

Figure 12 plots the compact-dimension sweep of Table 5, with and without classifier-free guidance, together with reconstruction quality.

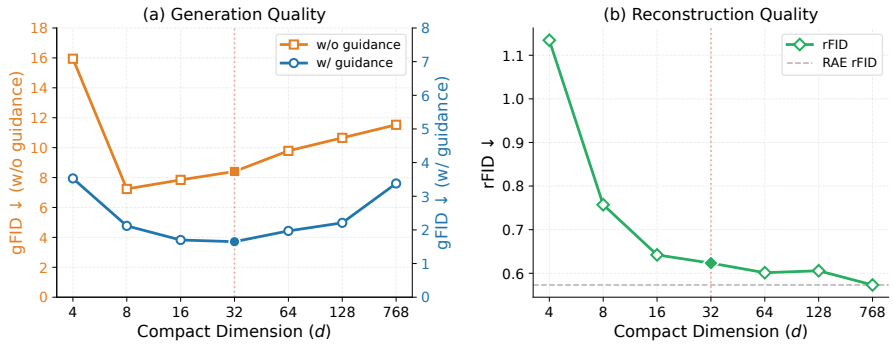


Figure 12: Effect of compact dimension d. Generation is best at $d = 32$, while reconstruction improves with larger d.

D Additional Controlled Experiments

This section collects additional controlled experiments on (i) the training budget and evaluation variance of the comparison with SimFlow, (ii) whether the gain stems from compression or from

retraining a tokenizer, (iii) the exact compact-space likelihood, (iv) direct semantic preservation and coverage, (v) transfer to other pretrained encoders and to COCO, and (vi) training and sampling cost. Unless stated otherwise, all results are on ImageNet 256×256 and use the same NF backbone and evaluation protocol as the main experiments.

D.1 Matched Training Budget

SRC-Flow is trained for 320 epochs, whereas SimFlow reports results at 160 epochs. To separate the effect of the representation from that of the longer schedule, we compare SRC-Flow at 80/160/320 epochs with SimFlow at its reported 160 epochs and with SimFlow continued to 320 epochs using its official configuration (trained by us; Table 12).

Table 12: Matched training budget on ImageNet 256×256. SimFlow-320 is trained by us with the official configuration.

Method	Epochs	gFID w/o cfg↓	IS w/o cfg↑	gFID w/ cfg↓	IS w/ cfg↑	Prec./Rec. w/ cfg
SimFlow	160	10.13	124.7	1.91	284.4	0.82/0.60
SimFlow	320	9.37	125.6	1.90	285.2	0.82/0.60
SRC-Flow	80	11.12	117.8	1.96	276.3	0.81/0.59
SRC-Flow	160	**9.28**	**125.1**	**1.73**	**288.9**	0.82/**0.61**
SRC-Flow	320	**8.40**	**127.4**	**1.65**	**289.7**	0.82/**0.61**

At a matched 160 NF-training epochs, SRC-Flow improves guided gFID by 9.4% (1.91→1.73) and unguided gFID by 8.4% (10.13→9.28). Extending SimFlow to 320 epochs improves its unguided gFID to 9.37 but leaves guided gFID essentially unchanged at 1.90. The principal guided-generation gain is therefore already present under the matched NF-training budget and cannot be attributed to the longer schedule. SRC additionally requires a 16-epoch compressor stage, whose cost is reported in Appendix D.9. The rFID of SimFlow-320 is 1.08.

D.2 Evaluation-Seed Variance

A single 50k draw does not quantify Monte Carlo evaluation variance, which matters for close comparisons. We fix each trained checkpoint and repeat the complete class-balanced ImageNet 256×256 evaluation with 10 independent generation seeds, each producing 50 images per class (50,000 images). Checkpoint and EMA weights, class order, decoders, reference statistics, evaluation code, and method-specific guidance settings (SimFlow CFG 1.1, SRC-Flow CFG 1.2) are fixed; only the Gaussian-prior sampling seed changes. Table 13 lists every evaluation and Table 14 summarizes them.

Table 13: Guided gFID/IS of fixed checkpoints over 10 evaluation seeds (ImageNet 256×256).

Seed	SimFlow-160	SRC-Flow-160	SimFlow-320	SRC-Flow-320
1	1.9297 / 282.18	1.7296 / 290.04	1.9231 / 283.99	1.6493 / 290.09
2	1.9144 / 280.87	1.7306 / 288.40	1.9103 / 282.84	1.6412 / 293.19
3	1.9431 / 283.72	1.7238 / 289.99	1.9344 / 285.35	1.6528 / 289.33
4	1.8990 / 282.80	1.7439 / 288.13	1.8974 / 284.55	1.6527 / 289.82
5	1.9026 / 283.16	1.7333 / 288.61	1.9004 / 284.86	1.6636 / 288.06
6	1.9015 / 283.00	1.7403 / 289.42	1.8995 / 284.72	1.6488 / 290.57
7	1.9043 / 282.91	1.7418 / 289.79	1.9018 / 284.64	1.6583 / 290.63
8	1.9079 / 283.40	1.7325 / 289.43	1.9048 / 285.08	1.6562 / 292.20
9	1.9035 / 282.66	1.7376 / 289.19	1.9012 / 284.42	1.6493 / 290.03
10	1.9154 / 282.50	1.7264 / 287.09	1.9111 / 284.28	1.6451 / 293.03

Table 14: Summary over 10 evaluation seeds. SD is the sample standard deviation across evaluation sets.

Checkpoint	gFID mean ± SD	gFID min–max	IS mean ± SD	IS min–max
SimFlow-160	1.9121 ± 0.0142	1.8990–1.9431	282.72 ± 0.79	280.87–283.72
SRC-Flow-160	1.7340 ± 0.0067	1.7238–1.7439	289.01 ± 0.94	287.09–290.04
SimFlow-320	1.9084 ± 0.0119	1.8974–1.9344	284.47 ± 0.69	282.84–285.35
SRC-Flow-320	1.6517 ± 0.0065	1.6412–1.6636	290.70 ± 1.65	288.06–293.19

The ranges are fully separated: at 160 epochs the worst SRC-Flow evaluation (1.7439) is better than the best SimFlow evaluation (1.8990), and at 320 epochs the corresponding values are 1.6636 and 1.8974. Unpaired Welch 95% intervals for the SimFlow−SRC-Flow gFID difference are

0.1782 $[0.1674, 0.1889]$ at 160 epochs and 0.2567 $[0.2475, 0.2659]$ at 320 epochs; for the SRC-Flow−SimFlow IS difference they are 6.29 $[5.47, 7.11]$ and 6.22 $[4.99, 7.45]$. The 95% t-intervals for mean gFID are $[1.9020, 1.9223]$ (SimFlow) and $[1.7292, 1.7388]$ (SRC-Flow) at 160 epochs, and $[1.8999, 1.9169]$ and $[1.6471, 1.6564]$ at 320 epochs. SimFlow's mean guided gFID changes by only 0.0037 from 160 to 320 epochs (overlapping intervals), whereas SRC-Flow improves by 0.0823. These intervals quantify evaluation-sampling variance of fixed checkpoints; they do not measure training-seed variance from retraining the models, and we do not conflate the two. Multi-seed samples are not available for most published baselines, so their variance cannot be estimated retrospectively.

D.3 Compression versus Tokenizer Retraining

SRC is itself a trained, decoder-adapted component, so part of the gain could come from retraining a tokenizer rather than from compression. A fair latent-tokenizer control must also account for decoder robustness: under the NF protocols considered here, Gaussian latent noise smooths the empirical support and stabilizes maximum-likelihood training, so the decoder must reconstruct perturbed and slightly off-manifold latents produced by the flow. A reconstruction-only VAE is therefore not directly comparable unless it is trained for the same noise robustness. We train two noise-robust VA-VAE (Yao et al., 2025) controls with the same NF backbone, 320-epoch schedule, and evaluation protocol as SRC-Flow (Table 15).

Table 15: Tokenizer controls with the same NF backbone and 320-epoch schedule (w/ guidance). †Reported by STARFlow.

Representation	Decoder noise	rFID↓	gFID↓	IS↑
STARFlow noise-robust VAE†	0.5	2.73	2.40	-
Noise-robust VA-VAE	0.5	1.04	1.97	282.3
Noise-robust VA-VAE	0.4	0.78	1.89	285.3
Full RAE, width 1152	RAE protocol	**0.57**	3.58	247.6
SRC ($d = 32$)	0.4	0.62	**1.65**	**289.7**

The full RAE reconstructs best but generates worst, and the noise-robust VA-VAE is strong but remains below SRC. Reconstruction quality, decoder retraining, or noise robustness alone therefore cannot explain the gain. The compressor ablation in Table 6 provides a second controlled chain: full RAE (3.54 at width 2048, Table 1), non-learned PCA 3.31, Linear 2.86, Conv 2.14, and Transformer SRC 1.65. PCA already improves on the full RAE without learning a new tokenizer, which isolates a compression benefit, and the progression from Linear to Conv to Transformer shows that preserving cross-token semantic structure contributes the remaining improvement.

D.4 Compact-Space Negative Log-Likelihood

We evaluate class-conditional NLL on ImageNet validation latents under the same Gaussian-noise distribution used for flow training, with the ground-truth class label, no guidance, and one forward evaluation per latent. Values include the Gaussian normalization constant and are normalized by the number of modeled scalars $D = Nd$ (nats/dim). Each row satisfies NLL = prior quadratic + $\frac{1}{2}\log(2\pi) + (-\log|\det J|/D)$ (Table 16).

Table 16: Exact compact-space NLL and its decomposition (nats/dim, ImageNet validation).

Representation	d	Flow width	NLL↓	Prior quadratic	$-\log\lvert\det J\rvert/D$
SRC	4	1152	0.380	0.605	−1.144
SRC	8	1152	0.345	0.612	−1.186
SRC	16	1152	0.350	0.578	−1.147
SRC	32	1152	0.409	0.529	−1.039
SRC	64	1152	0.455	0.512	−0.976
SRC	128	1152	0.565	0.505	−0.859
Full RAE	768	1152	0.850	0.504	−0.573
Full RAE	768	2048	0.838	0.503	−0.584

Widening the full-RAE flow from 1152 to 2048 improves NLL only from 0.850 to 0.838, mirroring the small guided-gFID change from 3.58 to 3.54. The prior quadratic is close to its Gaussian expectation at moderate and high d, so the differences are primarily reflected in the Jacobian term. The lowest normalized NLL occurs at $d \in \{8, 16\}$, whereas generation is best at $d = 32$: extremely compact

variables are easier to fit but discard semantic information. Because different d define different random variables, these values diagnose flow tractability; they are not pixel-space bits/dim and do not replace gFID.

D.5 Direct Semantic Preservation

PCA explained variance and rFID are indirect evidence that the information needed for generation is preserved. Table 5 in the main text reports the classifier accuracy of reconstructions across d; here we provide the complete semantic-preservation results. We evaluate ImageNet validation reconstructions with a separately trained, frozen MLP classifier (Top-1/Top-5), DINOv2-L with registers (Darcet et al., 2024), and OpenCLIP (Cherti et al., 2023). DINO and CLIP-I similarities are cosine similarities computed per image between the original and its reconstruction and then averaged (Table 17). Because the RAE encoder is itself DINOv2-based, we treat DINO similarity as a complementary measure and rely primarily on the independent classifier and on CLIP, which is trained with a different objective and data.

Table 17: Semantic preservation of reconstructions on ImageNet validation.

Representation	rFID↓	Top-1/Top-5↑	DINO sim.↑	CLIP-I sim.↑
Full RAE	0.57	82.6/96.6	0.9389	0.9178
SRC-128	0.60	78.9/95.1	0.9347	0.9172
SRC-64	0.60	74.7/93.8	0.9341	0.9164
SRC-32	0.62	69.1/89.8	0.9325	0.9152
SRC-16	0.64	47.3/70.4	0.9222	0.9113
SRC-8	0.75	20.7/43.2	0.9009	0.9017
SRC-4	1.13	7.3/19.6	0.8495	0.8767

At $d = 32$, global DINO and CLIP semantics remain close to the full RAE and Top-5 accuracy is 89.8%. We do not claim that every fine-grained discriminative cue is preserved: Top-1 continues to improve with larger d. These larger representations, however, do not improve generation; moving from $d = 32$ to $d = 64$ improves rFID from 0.62 to 0.60 and Top-1 from 69.1 to 74.7 but worsens 160-epoch guided gFID from 1.73 to 2.01. This quantifies the intended trade-off between semantic sufficiency and tractability: $d = 32$ is selected for the best generation balance rather than to maximize reconstruction or classification.

D.6 Coverage and Class-Conditional Diversity

For each ImageNet class we generate 50 samples and compute the mean pairwise cosine distance between normalized embeddings from an independent DINOv2-L-with-registers evaluator (Darcet et al., 2024) (DINO dispersion). Retention is reported relative to real ImageNet validation images (Table 18).

Table 18: Coverage and class-conditional semantic diversity (ImageNet 256×256, w/ guidance).

Method	gFID↓	Precision↑	Recall↑	DINO dispersion↑	Retention vs. real
SRC-4	3.53	0.80	0.52	0.222	83.5%
SRC-8	2.12	0.81	0.57	0.237	89.1%
SRC-16	1.69	0.82	0.60	0.245	92.1%
SRC-32	1.65	0.82	0.61	0.247	92.9%
SRC-64	1.97	0.81	0.61	0.246	92.5%
SRC-128	2.21	0.80	0.61	0.246	92.5%
Full RAE	3.58	0.79	0.56	0.234	88.0%
SimFlow-320	1.90	0.82	0.60	0.246	92.5%
Real ImageNet-val	-	-	-	0.266	100%

Over-compression clearly reduces coverage: from $d = 4$ to $d = 32$, Recall increases from 0.52 to 0.61 and diversity retention from 83.5% to 92.9%, after which both saturate while gFID degrades. Relative to SimFlow, SRC-32 improves gFID while matching Precision and matching or slightly exceeding Recall and dispersion. We find no evidence that SRC obtains its quality improvement by sacrificing coverage relative to the closest NF baseline, although neither method reaches real-data diversity or the Recall of the strongest diffusion and autoregressive models.

D.7 Transfer across Pretrained Encoders

We train SRC and the same NF backbone on the released DINOv2-, SigLIP2-, and MAE-based RAEs, using a matched 160-epoch schedule, the same evaluation protocol, and $\sigma_{\text{flow}} = 0.4$ without encoder-specific retuning (Table 19).

Table 19: SRC-Flow with different RAE encoders (160 epochs, ImageNet 256×256, w/ guidance).

Encoder	rFID at $d = 16/32/64$	gFID at $d = 16/32/64$	IS at $d = 16/32/64$
DINOv2	0.64 / 0.62 / 0.60	1.82 / **1.73** / 2.01	286.5 / **288.9** / 280.8
SigLIP2	0.97 / 0.91 / 0.87	2.12 / **2.05** / 2.32	279.4 / **281.8** / 270.3
MAE	0.58 / 0.43 / 0.36	3.72 / **3.36** / 3.37	243.3 / **256.2** / 255.7

All three encoders exhibit a compact optimum or saturation near $d = 32$, and $d = 64$ improves rFID without improving generation. Reusing $\sigma_{\text{flow}} = 0.4$ without retuning remains effective, although we do not claim that either $d = 32$ or $\sigma_{\text{flow}} = 0.4$ is universally optimal.

D.8 Zero-Shot Transfer to COCO

We evaluate the frozen ImageNet-trained RAE and SRC on all 5,000 COCO val2017 images (Lin et al., 2014) without finetuning (Table 20). DINO and CLIP-I similarity are per-image cosine similarities between the original and the reconstruction. To measure how much caption-relevant semantics survive compression, we use the (typically five) human captions of each COCO image. For image x with reconstruction $\hat{x}$ and captions $\{c_k\}$, we compute $S_{\text{ori}}(x) = \frac{1}{|\{c_k\}|} \sum_k \text{CLIP}(x, c_k)$ and $S_{\text{rec}}(x) = \frac{1}{|\{c_k\}|} \sum_k \text{CLIP}(\hat{x}, c_k)$, and report the caption-alignment retention

$$\text{Retention} = \frac{\mathbb{E}_x \left[S_{\text{rec}}(x) \right]}{\mathbb{E}_x \left[S_{\text{ori}}(x) \right]}, \tag{42}$$

i.e., the ratio of dataset means rather than the mean of per-image ratios, which would be sensitive to images whose original image–caption similarity is small. Normalizing by the original alignment accounts for the fact that the natural match between an image and its captions varies across images and caption styles, so retention directly measures how much of the caption-related semantics of the original image the reconstruction preserves.

Table 20: Zero-shot reconstruction and semantic preservation on COCO val2017.

Representation	COCO rFID↓	DINO sim.↑	CLIP-I sim.↑	Caption-align. retention↑
Full RAE	6.005	0.9034	0.9088	99.23%
SRC-128	6.202	0.8983	0.9087	99.12%
SRC-64	6.241	0.8961	0.9076	98.99%
SRC-32	6.388	0.8904	0.9061	98.97%
SRC-16	6.705	0.8782	0.9003	98.93%
SRC-8	7.257	0.8488	0.8871	98.68%
SRC-4	8.826	0.7736	0.8487	97.54%

The COCO sweep mirrors ImageNet: reconstruction and semantic preservation improve rapidly up to $d = 32$ and then show diminishing returns. This supports zero-shot representation transfer beyond ImageNet. It is not a text-to-image evaluation: the current model has no text-conditioning interface, and fine-grained generative control over attributes, counts, spatial relations, or compositions is not established here.

D.9 Training Compute

All costs are measured on $8\times$A100 GPUs (Table 21).

Table 21: Stage-wise training cost.

Component	Epochs	GPU-hours	8-GPU wall-clock
RAE decoder	16	288	36.0 h
SRC	16	84	10.5 h
SRC-Flow NF	160 / 320	1,008 / 2,016	126 / 252 h
SimFlow joint VAE+NF	160 / 320	1,260 / 2,520	157.5 / 315 h

Starting from the released RAE checkpoint, SRC+NF costs 1,092 GPU-hours at 160 epochs and 2,100 at 320; the SRC stage is 4.0% of the 320-epoch cost. Counting RAE decoder training, the totals are 1,380 and 2,388 GPU-hours versus 1,260 and 2,520 for SimFlow. Under full-pipeline accounting SRC-Flow is therefore 9.5% more expensive at 160 epochs and 5.2% less expensive at 320 epochs; under released-checkpoint accounting it is cheaper at both budgets.

D.10 Sampling Throughput

Table 22 lists representative throughput values underlying Figure 11, measured as aggregate images/s on $8\times$A100 under the same timing protocol and global batch sizes.

Table 22: Sampling throughput (aggregate images/s, $8\times$A100) versus global batch size.

Batch	DiT-25 steps	DiT-50 steps	DiT-250 steps	AR	SRC-Flow
1	0.980	0.490	0.100	0.160	0.030
16	1.140	0.570	0.110	0.690	0.400
64	1.140	0.570	0.110	0.970	1.040
128	1.130	0.560	0.105	1.080	1.320

SRC-Flow remains autoregressive and therefore has substantially worse single-sample latency; deterministic sampling does not imply fast low-batch generation. Its practical advantage is batch-level parallelism: throughput rises from 0.03 images/s at batch 1 to 1.32 at batch 128, where it becomes competitive with or exceeds the evaluated autoregressive and iterative baselines.

E Additional Visualizations

E.1 Reconstruction across Compact Dimensions

Figure 13 visualizes reconstruction results under different compact representation dimensions. Very small dimensions lose visible details, while larger dimensions gradually approach the original RAE reconstruction. The $d = 32$ SRC provides a favorable visual trade-off, preserving most image details while substantially reducing the modeled channel dimension.

E.2 Reconstruction under Different Noise Levels

Figures 14 and 15 compare reconstructions under different input noise levels for the $d = 32$ SRC and the original RAE representation, respectively. Although the quantitative rFID values vary with noise strength, the visual differences remain subtle across the tested perturbation range. This suggests that both representations are visually robust to moderate perturbations, while the metric differences reported in the main paper capture more fine-grained distributional changes.

E.3 Additional Class-Conditional Samples

Figures 16–21 provide additional class-conditional samples generated by SRC-Flow. Each figure corresponds to one broad semantic category and contains four ImageNet subclasses from that category. For each broad category, we show samples at both 256×256 and 512×512 resolutions using the same guidance setting as in the main experiments. These samples further demonstrate the generation quality and semantic diversity of SRC-Flow across different object groups.

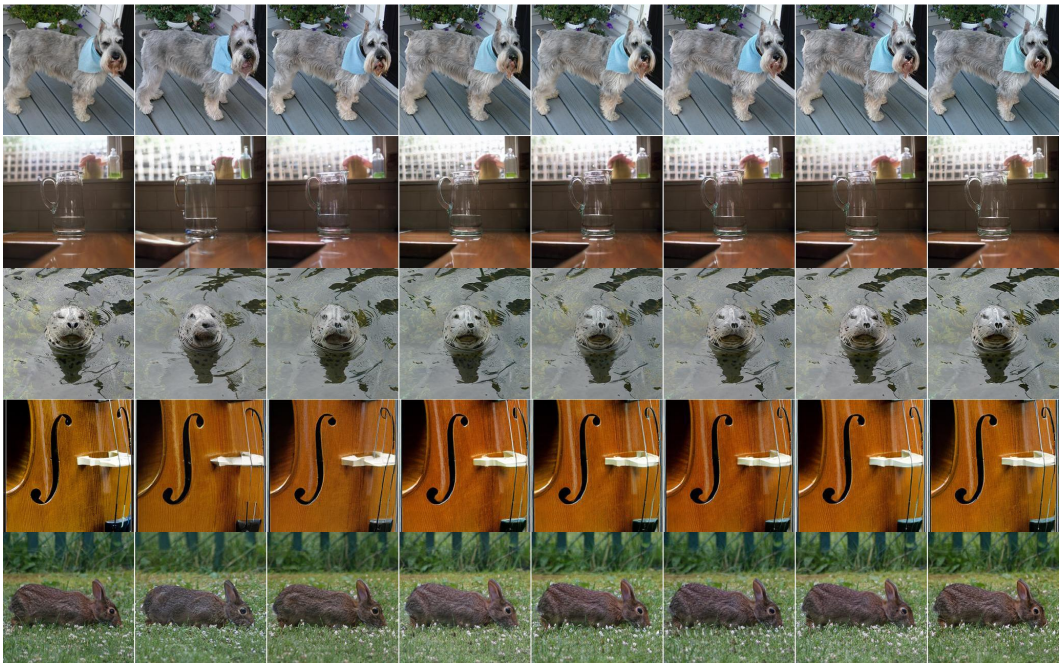


Figure 13: Reconstruction visualization across compact representation dimensions. The $d = 32$ SRC preserves visual details close to the original RAE while substantially reducing the modeled channel dimension.

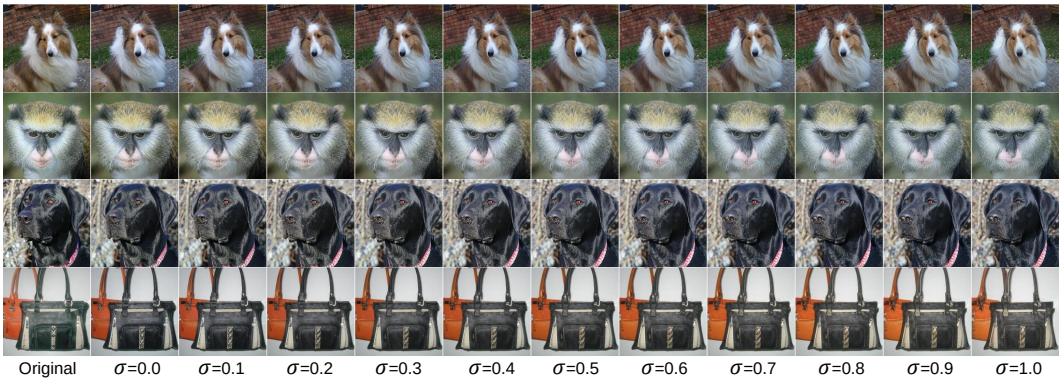


Figure 14: Reconstruction of the $d = 32$ SRC under different noise levels. Visual differences remain subtle across the tested perturbation strengths, although quantitative reconstruction metrics vary with noise.

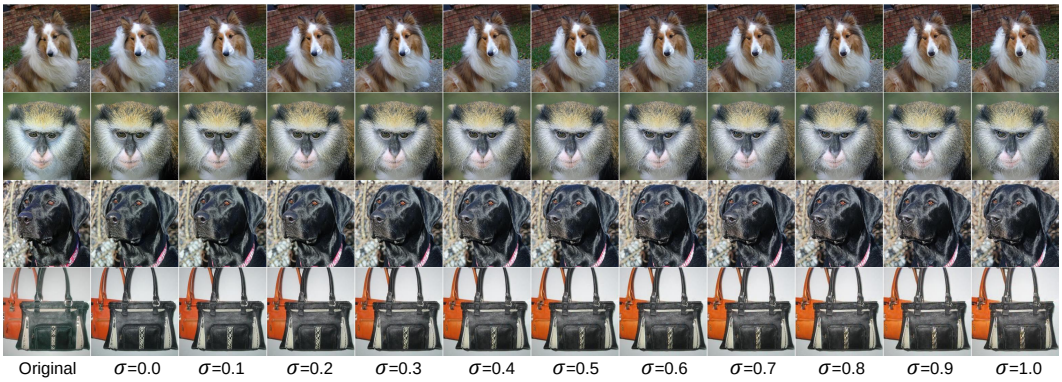


Figure 15: Reconstruction of the original RAE representation under different noise levels. Similar to SRC, the visual effect of noise remains relatively small, despite measurable changes in reconstruction metrics.



Figure 16: Additional class-conditional samples from the dog category.



Figure 17: Additional class-conditional samples from the bird category.



Figure 18: Additional class-conditional samples from the car category.



Figure 19: Additional class-conditional samples from the cat category.



Figure 20: Additional class-conditional samples from the fish category.



Figure 21: Additional class-conditional samples from the plant category.