ROMS-IMLE: A Minimalist Approach to Competitive One-Step Generative Modelling

Chirag Vashist
APEX Lab
Simon Fraser University
chirag_vashist@sfu.ca

Ke Li
APEX Lab
Simon Fraser University, Amii and CIFAR
keli@sfu.ca

Abstract

Generative models have undergone many generations of evolution, from VAEs/GANs to diffusion/flow matching. Along the way, the underlying techniques have become more complicated and various beliefs about what drives strong empirical performance have taken hold. Due to the success of diffusion models and flow matching, one of the more common beliefs is the importance of iterative denoising. We ask whether iterative denoising is truly necessary, and take a minimalist approach to designing a competitive generative model. We start with the bare-bones essentials, namely just a training objective and a model. We purposefully make both simple. For the training objective, we choose Implicit Maximum Likelihood Estimation (IMLE), and eschew more complicated alternatives such as variational inference, adversarial training and numerical integration. For the model, we eschew transformers and instead choose a moderately sized convolutional network. Then we judiciously added elements that are truly essential, which surprisingly do *not* include iterative denoising. The result is a one-step parameter-efficient generative model that produces high quality samples at fast speed: it achieves an FID of 2.56 on ImageNet 256 and simultaneously attains good precision and recall.

1 Introduction

Image generation has advanced through a succession of paradigms over the past decade. Variational autoencoders (VAEs) [29] and generative adversarial networks (GANs) [19] were among the first models to generate realistic images in a single forward pass, yet each approach carries a characteristic weakness. GANs are unstable to train and prone to mode collapse, which limits the diversity of their samples. VAEs train stably but tend to produce blurry, low-fidelity samples. Stochastic interpolant models (SI for short), including diffusion and flow-matching models [23, 47, 35, 4], resolved many of these issues and now define the state of the art in image synthesis. Inspired by this success, subsequent work has grown steadily more complicated, introducing elaborate architectures [41, 38], sophisticated sampling schemes [28, 46], and intricate training objectives [48, 17, 56, 16].

We ask whether all of the complex machinery introduced in recent years is truly necessary. Rather than add to it, we take a minimalist approach: we start from the simplest generative model we can and add only the components from modern approaches that prove essential. We choose to build on Implicit Maximum Likelihood Estimation (IMLE) [34], which offers a stable, mode-covering training objective that is easy to understand. IMLE fits this minimalist aim because it optimizes a likelihood-based objective without the variational bounds of VAEs, the adversarial training of GANs, or the iterative numerical integration used by stochastic interpolant models.

The advantage of modern SI models over earlier single-step models is widely attributed to how they decompose generation [53, 13]: instead of mapping the prior to the data distribution in one large transformation, they break it into a sequence of smaller, simpler transformations, which is thought to

Preprint.

Figure 1: Random samples from our class-conditional model trained on ImageNet

be the key to producing high-quality samples. Each of these transformations is typically carried out by a separate step in the sampling process, and as a result sampling takes many steps to iterate over all the transformations. Iterative sampling is, however, computationally expensive, since it involves multiple forward passes through the model. If we can achieve high-quality generation without the cost of iterative sampling, we can make image generation both cheaper and simpler.

To that end, we start with a single-step model and add only a minimal set of modifications that are necessary to achieve competitive performance. To devise the modifications, we find the mechanisms behind SI models that are truly important, abstract them at a high level to enable application to the single-step setting and add them to our single-step model. Rather than viewing SI models as performing iterative sampling, we take a testing-centric view. We hypothesize that two ingredients that power their success are per-stage supervision and spectral specialization and add them to our single-step model. In addition, we identify challenges specific to the one-step setting, develop simple fixes and add them to our design. In line with our minimalist design philosophy, we use a ConvNeXt-based architecture [36] rather than the transformer-based architectures that now dominate. Together, these design choices yield **Ro**bust **M**ulti-**S**tage IMLE (ROMS-IMLE), a single-step generative model that combines spectrally-aware per-stage supervision with a robust training loss. Across three diverse datasets, our model attains competitive FID and strong precision and recall. In pixel space, ROMS-IMLE attains the highest precision and recall compared to baselines on CIFAR-10 and CelebA-HQ. For latent generation on ImageNet 256, it reaches an FID of 2.56 in a single forward pass, with 54% fewer parameters and 250$\times$ fewer function evaluations than DiT-XL/2 and SiT-XL/2. Through our results, we demonstrate that a simple single-step generative model can indeed retain the sample quality and diversity commonly exhibited by popular SI models.

2 Related Works

Power law in images Early works [49, 1] in computer vision observed that the average power spectrum for natural images is approximately of the form $1/f^{\alpha}$, where f is the spatial frequency and $\alpha \sim 2$. This implies that lower spatial frequencies contain significantly more magnitude and encode the dominant structural components of images, whereas higher frequencies contribute relatively less to the overall image content. This fact has been used for many applications including image compression. In particular, JPEG [52] truncates DCT [3] coefficients of image blocks, keeping only the low-frequency components.

One-step generative models GANs remain strong one-step image generators [10, 45, 54, 40] but are difficult to train [6, 5] and prone to mode collapse [9, 15]. A separate line of research tries to make iterative stochastic interpolant models fast by reducing the number of sampling steps, through

distillation [44], consistency training [48], or redesigned objectives such as shortcut models [16], MeanFlow [17, 18], and inductive moment matching [56]. In recent years, IMLE [34] has received increasing attention for its mode-covering objective and strong performance in few-shot image synthesis [51, 2]. IMLE has also been adopted beyond image generation, in areas such as visuomotor policy learning [42], trajectory planning [33], and shape completion [7].

3 Method

3.1 Background: IMLE

The goal of generative modeling is to transform a simple, easy-to-sample prior distribution into the complex data distribution we wish to model. One way to approach this problem is to learn the transformation directly, using a neural generator that maps samples from the prior to data samples in a single pass. In general, this approach is difficult because the likelihood of the data under the generator is intractable (refer to Appendix A for more details). Implicit Maximum Likelihood Estimation (IMLE) [34, 2] provides a workaround by implicitly maximizing the likelihood without requiring to know it explicitly.

Formally, IMLE learns a direct mapping f_θ from a prior $\pi(\mathbf{x})$ to the data distribution $p_{\text{data}}(\mathbf{x})$ by minimizing the following objective. At each training step, m latent codes $\mathbf{z}_1, \ldots, \mathbf{z}_m$ are sampled from the prior and each real image $\mathbf{x}_i$ is matched to its nearest generated sample in the set $\{f_\theta(\mathbf{z}_1), \ldots, f_\theta(\mathbf{z}_m)\}$. The parameters are then updated to bring each generated sample closer to its matched real image. Here, $d(\cdot, \cdot)$ is a distance function. The objective can be written as follows:

$$\theta_{\text{IMLE}} = \arg\min_\theta \mathbb{E}_{\mathbf{z}_1,\ldots,\mathbf{z}_m \sim \mathcal{N}(0,I)} \left[\sum_{i=1}^{n} \min_{j \in [m]} d\Big(\mathbf{x}_i, f_\theta(\mathbf{z}_j) \Big) \right] \tag{1}$$

Our goal in this paper is to test whether IMLE, a single-step generator, can match the sample quality of modern generative models that generate an image over many sequential steps. We hypothesize that the sample quality of these many-step models stems not from iterative sampling itself but from their parameter estimation strategy that can be adapted to IMLE. To this end, we first present an alternative view of stochastic interpolant models, and use it to identify the training properties behind their success.

3.2 Testing-Centric View of Diffusion and Flow Matching

Stochastic interpolant (SI) methods, such as diffusion and flow matching, are commonly understood to perform iterative denoising. At training time, SI methods train a model, commonly known as a denoising network, to convert a noisy image to a clean image. At test time, the denoising network is repeatedly applied to convert pure noise to a generated sample. For the purposes of this analysis, we use diffusion as a representative case; the argument applies equally to other SI methods.

A key distinguishing feature of SI methods is the iterative application of the denoising network, and this is what SI methods are believed to owe their success to. This creates a mismatch between training and test; the denoising network is applied once for each sample during training, whereas it is applied iteratively to generate a sample during test.

We consider an alternative view of this mismatch. Rather than viewing the model as one instance of the denoising network, we unroll the different iterations of the sampling procedure and view the model as a composition of the different iterations. Under this view, at test time, the model is just applied once to convert pure noise to a generated sample, and at training time, only a part of the model is evaluated at a time. This is in some sense the flip side of the conventional view; under the conventional view, the test time procedure is what deviates from the norms, whereas under the alternative view, the training time procedure is what deviates from the norms. Hence, we will term the conventional view the *training-centric view* and the alternative view the *testing-centric view*.

We now construct concretely the model under the testing-centric view. Consider a deterministic sampler in an SI method, such as those used in DDIM or flow matching. Starting from Gaussian noise, the sampler applies a succession of updates parameterized by the denoising network evaluated at different time steps. More precisely, if we denote the update at iteration t as a function g_θ^t, the

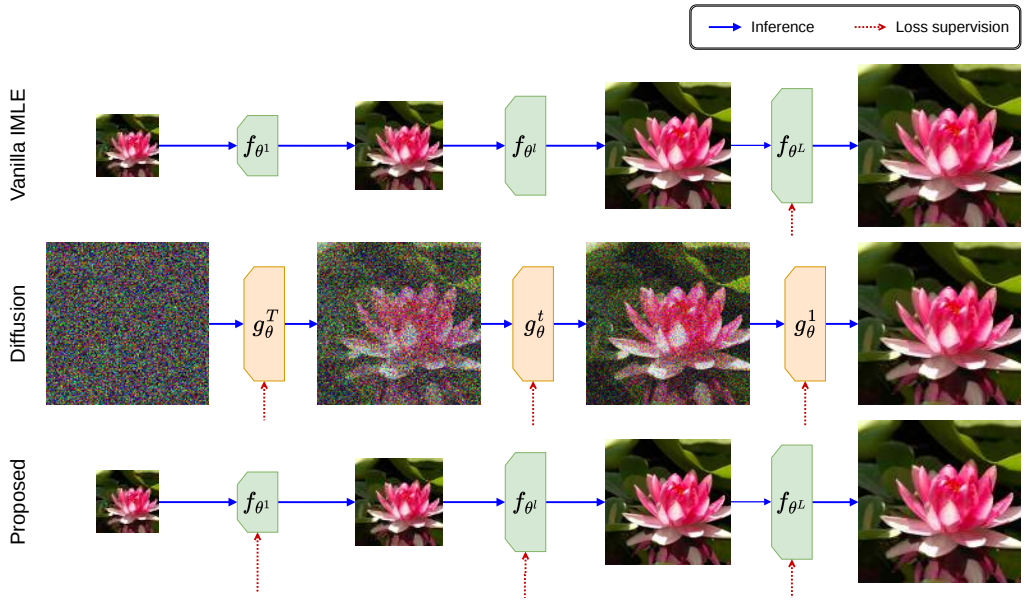


Figure 2: **Per-stage supervision:** Each row is a generator drawn as a composition of stages; blue arrows are the inference path and red dotted arrows show where the loss is applied. Vanilla IMLE (top) supervises only its final stage, whereas diffusion (middle) supervises every denoising stage. Our proposed method (bottom) retains IMLE's architecture and one-step inference while extending direct supervision to every stage.

sampler starts by sampling $\mathbf{x}_T \sim p_T(\mathbf{x})$ and produces a sequence $\mathbf{x}_T \to \mathbf{x}_{T-1} \to \cdots \to \mathbf{x}_0$, where at each iteration $\mathbf{x}_{t-1}$ is obtained from $g_\theta^t(\mathbf{x}_t)$.

Under the testing-centric view, the model is the composition of all the updates done by the sampler, namely $g_\theta^T, \ldots, g_\theta^1$.

$$\mathbf{x}_0 = \underbrace{g_\theta^1 \circ g_\theta^2 \circ \cdots \circ g_\theta^T}_{h_\theta}(\mathbf{x}_T), \tag{2}$$

This allows us to view the reverse process as a *single* compositional model h_θ, that maps a sample from the prior $\mathbf{x}_T$ to a data sample $\mathbf{x}_0$. Under this view, each $\mathbf{x}_{t-1} = g_\theta^t(\mathbf{x}_t)$ is an intermediate layer of h_θ.

Now we consider the training objective of SI methods. In general, they supervise the prediction of the denoising network at each time step, with equal weight applied to all time steps. More precisely, the training objective takes the following form:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,T],\, \mathbf{x}_0\sim p_{\text{data}},\, \mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)} \left[\ell_t(g^t(\mathbf{x}_0, \mathbf{x}_t), g_\theta^t(\mathbf{x}_t))\right] \tag{3}$$

where g^t is the supervision target and q_t characterizes the interpolation between the empirical data distribution p_{data} and the prior p_T. If we restrict the support of t to integer time steps, we can rewrite the objective as:

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_0\sim p_{\text{data}},\, \mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)} \left[\ell_t(g^t(\mathbf{x}_0, \mathbf{x}_t), g_\theta^t(\mathbf{x}_t))\right] \tag{4}$$

Hence, what the training-centric view treats as an iteration over time steps is now viewed as an iteration over depth in the compositional model.

3.3 Spectrally-Aware Per-Stage Supervision

Recall that our goal was to add desirable properties of SI models to our minimalist IMLE framework. In the previous section, we presented the SI sampler as a single compositional model h_θ. We can

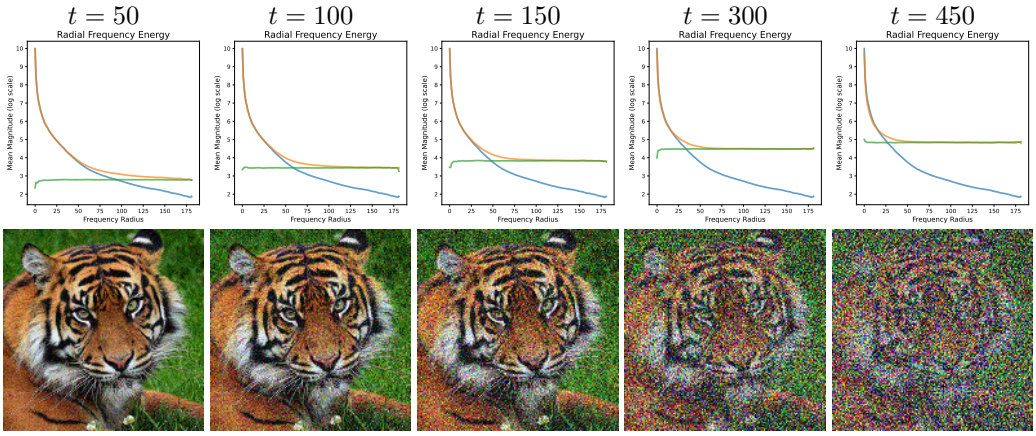


Figure 3: The noising process in diffusion methods behaves like a progressive low-pass filter. Top row: log-magnitude vs. frequency plots at different diffusion time steps, illustrating how high-frequency components of the original image are increasingly suppressed. Blue denotes the original unnoised images, green represents the added noise, and orange indicates the resulting noised image. Bottom row: corresponding noised images at each time step.

now contrast this compositional model with the single-step generation setup of vanilla IMLE. Two training-time properties stand out, which we argue are the reason for high-quality generation in SI models.

Per-stage supervision Let us examine how the compositional model h_θ is trained by reconsidering the loss in Equation 4. We note that each stage g_θ^t in the composition is directly supervised. Under the testing-centric view, the diffusion training objective therefore performs *intermediate per-stage supervision*: every intermediate layer g_θ^t of the compositional model h_θ receives a direct training signal (Figure 2).

Spectral specialization This intermediate supervision is also spectrally structured. It has been observed that diffusion models produce samples in a coarse-to-fine manner: coarser-scale structure appears in earlier denoising steps, whereas later denoising steps add more and more fine-grained details. Recent work [14] shows that adding noise to an image in the forward process has an effect similar to applying a low-pass filter, with high noise levels progressively removing higher frequencies (illustrated in Figure 3). Hence, within the composition h_θ, the earlier layers are responsible for coarse, low-level structure while later layers progressively add higher frequency details. In short, generation happens in a *spectrally-aware* fashion.

3.4 Adapting Spectrally-Aware Per-Stage Supervision to IMLE

We now bring this training signal to the IMLE framework. The unrolled view introduced above applies naturally to IMLE: its generator can also be seen as a prior-to-data network composed of stages, only executed in a single forward pass rather than through iterative sampling. What it lacks is per-stage supervision, which we add through a change to the training objective; spectral specialization then follows as a corollary.

Compositional scaffold We choose to build the IMLE generator f_θ as a stack of upsampling blocks. We adopt this design deliberately; similar to Section 3.2, it lets us write f_θ as a composition

$$f_\theta = f_{\theta^L} \circ f_{\theta^{L-1}} \circ \cdots \circ f_{\theta^1},$$

in which each stage f_{θ^l} operates at a progressively higher resolution, with per-stage parameters $\theta = (\theta^1, \ldots, \theta^L)$. This recasts IMLE in a form similar to the diffusion construction, with each f_{θ^l} now analogous to g_θ^t, only indexed by resolution rather than noise level.

Property 1: Per-stage supervision The architecture of f_θ in existing IMLE methods [51, 2] is compositional, but the training is not. The loss in Equation 1 is computed only at the final output $f_\theta(\mathbf{z})$, which directly supervises only stage L. Stages $1, \ldots, L-1$ receive a training signal only through gradients that flow back from this final loss. We address this by supervising every stage directly, at the resolution at which it operates.

Concretely, two things are needed: a way to read out a prediction at each stage, and a way to obtain a target at the matching resolution. We add a per-stage output head (see Figure 2), so that stage f_{θ^l} outputs an image $\tilde{\mathbf{x}}^l$ at its own resolution. We obtain the target by downsampling the real image $\mathbf{x}_i$ to the same resolution with a kernel K_l. The training objective then sums a per-stage loss across all L stages, comparing each $\tilde{\mathbf{x}}^l$ against $K_l(\mathbf{x}_i)$.

Property 2: Spectral specialization It is well known that the downsampling kernel K_l is a low-pass filter: $K_l(\mathbf{x}_i)$ retains only the frequencies representable at stage l's resolution and discards the rest. Stage l is therefore supervised to produce only the low-frequency content of $\mathbf{x}_i$ at that resolution, leaving finer detail to later stages. The composition f_θ as a whole generates coarsely first and progressively refines, like the spectral specialization property of diffusion models.

Algorithm 1 ROMS-IMLE training procedure

Require: Dataset $\{\mathbf{x}_i\}_{i=1}^n$, generator $f_\theta = f_{\theta^L} \circ \cdots \circ f_{\theta^1}$ with parameters $\theta = (\theta^1, \ldots, \theta^L)$, downsampling kernel $K_l(\cdot)$, robust loss function $d(\cdot, \cdot)$

Initialize θ to a random vector
for $t = 1$ **to** T **do**
  Draw i.i.d. latent codes $\mathbf{z}_1, \ldots, \mathbf{z}_m \sim \mathcal{N}(0, I)$
  Generate $\tilde{\mathbf{x}}_j \leftarrow f_\theta(\mathbf{z}_j)$ for all $j \in [m]$
  $\sigma(i) \leftarrow \arg\min_{j \in [m]} \|\mathbf{x}_i - \tilde{\mathbf{x}}_j\|_2^2 \quad \forall i \in [n]$ ▷ Nearest-neighbour matching
  for $r = 1$ **to** R_{steps} **do**
    Pick a random batch $S \subseteq [n]$
    Compute per-stage outputs $\tilde{\mathbf{x}}^l_{\sigma(i)} \leftarrow (f_{\theta^l} \circ \cdots \circ f_{\theta^1})(\mathbf{z}_{\sigma(i)})$ for all $i \in S, l \in [L]$
    $\mathcal{L} \leftarrow 0$
    for $l = 1$ **to** L **do**
      $\mathcal{L} \leftarrow \mathcal{L} + \frac{1}{|S|} \sum_{i \in S} d\Big(K_l(\mathbf{x}_i),\ \tilde{\mathbf{x}}^l_{\sigma(i)}\Big)$ ▷ Per-stage supervision
    end for
    $\theta \leftarrow \theta - \frac{\eta}{L} \nabla_\theta \mathcal{L}$ ▷ Gradient step
  end for
end for
return θ

We test this new training procedure on Oxford Flowers [39], an image dataset of roughly 1000 images at 256×256 resolution. We use this dataset for the ablation because it is small enough to train and iterate over quickly, yet visually diverse enough that mode collapse is easy to spot in the generated samples. Table 1 reports the results and Figure 4 shows random samples from each configuration: our proposed training paradigm with multi-stage supervision (B) substantially outperforms vanilla IMLE (A).

Table 1: Ablation study on Oxford Flowers [39] (256×256). Row B adds multi-stage supervision to the baseline. Rows C-E each replace the loss function in Row B with a different robust loss.

	Configuration	FID ↓	Pr. ↑	Re. ↑
A	Vanilla IMLE	69.35	0.37	0.40
B	+ Multi-stage supervision	35.07	0.75	0.70
C	(B) + Cauchy loss	32.70	0.78	0.71
D	(B) + Pseudo-Huber loss	17.26	0.89	0.74
E	(B) + Geman-McClure loss **(ours)**	**15.46**	**0.91**	**0.73**

3.5 Robust Loss for Mismatched Pairs

SI models interpolate between the noise distribution and the empirical data distribution along a predetermined trajectory, and so the flow between them does not need to be recomputed during training. On the other hand, in a single-step model, the model distribution changes during training, so the matching between model samples and ground truth data points needs to be recomputed. Due to randomness in the sampling, there may be no samples drawn from a mode of the model distribution, and as a result, a ground truth data point that can otherwise be explained by that mode would not

(a) **Conf-A**: Vanilla IMLE (b) **Conf-B**: Multi-stage supervision (c) **Conf-E**: Multi-stage supervision + Geman-McClure loss

Figure 4: **Qualitative ablation**: Random samples from IMLE models trained on Oxford Flowers (256×256) for selected configurations in Table 1. The multi-stage supervision in Conf-B substantially improves sample quality over the vanilla IMLE in Conf-A. The robust loss in Conf-E further improves sample quality, yielding sharper and more realistic images.

Figure 5: **Mismatch in nearest-neighbour pairing.** Ground-truth image (top) and its nearest neighbours among generated samples in the current round (middle) and the previous round (bottom). For most images, both are close to the ground truth. For the highlighted image, the current-round nearest neighbour does not resemble the ground truth.

be well matched. Note that this is different from mode collapse: the poor matching results from randomness in the sampling, not a deficiency in the model distribution.

Figure 5 shows this phenomenon, where the ground truth image (top row) alongside its nearest-neighbour matches in the current round (middle) and the previous round (bottom). For most images, both matches found in consecutive rounds of sampling are close to the ground truth, indicating that the generator has learned these targets well. For the highlighted image, however, the current-round match does not look like the ground truth, while the previous-round match does.

The standard squared ℓ_2 loss lets these mismatches dominate training, since its gradient with respect to the generated sample is proportional to the residual. A mismatched pair has a large residual and therefore contributes a correspondingly large gradient. The problem is, in essence, one of outliers: a small number of large-residual pairs distort the gradient computed from a much larger pool of well-matched pairs.

The classical remedy in robust statistics is to replace the squared loss with a function that grows sub-quadratically in the residual. We adopt this approach: we replace $d(\cdot, \cdot)$ with a robust loss function. When a mismatch occurs and produces a large residual, the robust loss reduces its influence on the gradient, allowing the remaining well-matched pairs to dominate the update. We apply the robust loss at each decoder stage and ablate three candidates [8]: Cauchy, Pseudo-Huber, and Geman-McClure. As shown in Table 1, all three improve performance over the non-robust loss, with Geman-McClure performing best. Figure 4 shows that Geman-McClure with multi-stage supervision also yields the best qualitative performance, producing sharper and more realistic images than the other configurations.

3.6 Round-Trip Rejection for Latent-Space Generation

For high-resolution datasets, a common paradigm is to generate in the latent space [50, 43] of a frozen, pretrained autoencoder and decode the generated latents into images. In the literature on SI

Figure 6: Random samples from our unconditional model trained on CelebA-HQ

models and GANs, inference-time tradeoffs of quality against diversity are popular and are enabled by techniques such as classifier-free guidance (CFG) [22] and the truncation trick [10]. To enable the same tradeoff, we design a simple technique, which we call round-trip rejection.

Latent space generation introduces a distribution mismatch: the autoencoder faithfully reconstructs only the images it was trained on, but our generator is not constrained to stay within that distribution and occasionally produces images the autoencoder does not represent well, which decode into visibly degraded samples. We add an inference-time step that identifies and discards these samples. We detect them using a *round-trip cost*: the distance between a generated image and the image obtained by encoding and then decoding it again. This round trip projects an image onto the distribution the autoencoder can represent, so a well-represented image changes little (low cost) while a poorly-represented one changes substantially (high cost). We reject generated samples whose round-trip cost is high. We include further implementation details in Appendix B.

Putting it together. Combining per-stage supervision with the robust Geman-McClure loss yields our final training recipe, which we refer to as Robust Multi-Stage Implicit Maximum Likelihood Estimation (ROMS-IMLE). The architecture and inference path are unchanged from vanilla IMLE; the changes are entirely in the training objective. Algorithm 1 states the full procedure.

4 Experiments

Datasets We demonstrate the performance of our method on three datasets: unconditional generation on CIFAR-10 (50K images, 32×32 resolution) [31] and CelebA-HQ (30K images, 256×256 resolution) [26], and class-conditional generation on ImageNet 256 [12].

Evaluation metrics We use the FID-50k metric [21] to assess the perceptual quality of the generated images, as is standard in prior work. Since FID is not always a perfect indicator of perceptual quality [25], we additionally evaluate the fidelity and coverage by computing precision and recall using the metric defined by Kynkäänniemi et al. [32].

Implementation details Our network architecture comprises a fully-connected mapping network inspired by [27] and a synthesis network constructed using ConvNeXt [36] modules. We use bicubic interpolation for downsampling images/latents. We use the AdamW optimizer [37] with β_1, β_2 set to 0.9. Our method works in both pixel and latent space: directly in pixels for CIFAR-10 and CelebA-HQ, and in the EQ-VAE [30] latent space for ImageNet 256.

Table 2: Results for unconditional generation on CIFAR-10

Method	NFE	FID ↓	IS ↑	Pr. ↑	Re. ↑
DDIM [46]	100	11.18	8.62	0.62	0.57
CT [48]	1	8.79	8.61	<u>0.70</u>	0.42
Mean Flows [17]	1	**2.87**	10.10	0.69	0.59
IMM [56]	1	3.16	10.10	0.66	<u>0.60</u>
StyleFormer [40]	1	<u>2.88</u>	**10.26**	0.64	0.54
Ours (Pixel)	1	4.01	<u>10.16</u>	**0.91**	**0.80**

Table 3: Results for unconditional generation on CelebA-HQ

Method	NFE	FID ↓	Pr. ↑	Re. ↑
Shortcut [16]	1	24.65	<u>0.75</u>	0.18
Shortcut [16]	128	12.88	0.61	<u>0.42</u>
StyleSwin [54]	1	**5.32**	0.70	0.39
Ours (Pixel)	1	<u>6.70</u>	**0.96**	**0.61**

Table 4: Results for class-conditional image generation on ImageNet 256

Model	Params (M)	NFE	FID ↓	Pr. ↑	Re. ↑
BigGAN[10]	112	1	6.69	0.47	0.14
StyleGAN-XL[45]	166	1	2.08	0.29	0.40
DiT-L/2 [41]	458	250	23.30	-	-
DiT-XL/2	675	250	9.60	-	-
DiT-XL/2 (cfg $= 1.5$)	675	250	2.64	0.54	0.60
SiT-L/2 [38]	458	250	18.80	-	-
SiT-XL/2	675	250	8.30	-	-
SiT-XL/2 (cfg $= 1.5$)	675	250	1.74	0.48	0.59
MeanFlow[17]	130	1	13.04	0.29	0.31
iMeanFlow[18]	1000	1	1.48	0.48	0.57
IMM (1 step, cfg $= 1.5$)[56]	676	1	8.13	0.32	0.62
IMM (2 steps, cfg $= 1.5$)	676	2	3.64	0.38	0.68
Ours (Latent-based)	310	1	4.16	0.77	0.49
Ours (Latent-based) + Round-trip rejection	310	1	2.56	0.79	0.50

4.1 Results and Analysis

CIFAR-10 Table 2 shows that our model attains the best precision and recall in the table by wide margins: precision of 0.91 against 0.70 for the next best method, and recall of 0.80 against 0.60. Notably, these margins hold even against the 100-step DDIM baseline (0.619 / 0.567), so a single forward pass of our model covers the data distribution better than a hundred evaluations of a diffusion sampler. At one NFE (number of function evaluations), our FID is competitive with the strongest one-step baselines, and our Inception Score (10.16) is within their range.

CelebA-HQ Table 3 shows that our model has the best precision and recall, while being the second-best on FID. The 128-step Shortcut model [16] is outperformed on every metric despite it using more evaluation steps. Compared to StyleSwin, which achieves the best FID, our model produces more diverse samples (recall 0.61 vs. 0.39) at a small cost in fidelity.

ImageNet 256 Table 4 shows that our latent-space model reaches an FID of 4.16 at one NFE with 310M parameters. With round-trip rejection, our FID improves to 2.56. Our FID is competitive with strong one-NFE and multi-step baselines such as iMeanFlow, StyleGAN-XL, DiT-XL/2, and SiT-XL/2, while using a single forward pass and no classifier-free guidance. Our model also achieves the best precision (0.79) among all the methods.

5 Discussion and Conclusion

Limitations Although ROMS-IMLE substantially outperforms vanilla IMLE, it processes images across multiple resolutions during training, which marginally increases training time.

Figure 7: **Latent space interpolation for CelebA-HQ**. We observe that the output images change smoothly in a meaningful manner.

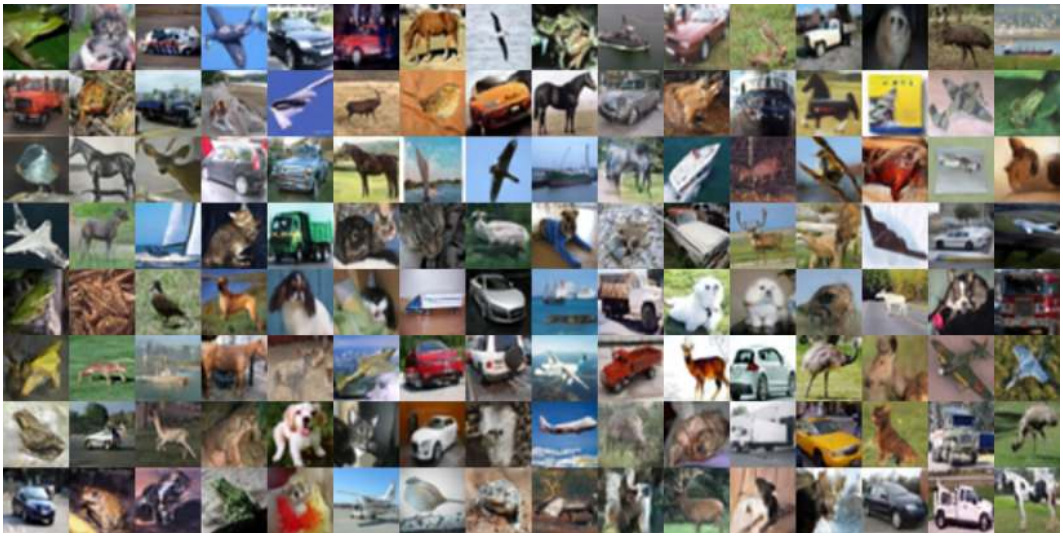


Figure 8: Random samples from our unconditional model trained on CIFAR-10

Societal Impact We are aware that generative models can facilitate creating fake images ("deepfakes") that can be exploited for disinformation or fraudulent activities, but note that watermarking techniques, e.g., [20], have been developed that can mitigate such harms.

Conclusion We introduced ROMS-IMLE, a single-step IMLE-based generative model. The method modifies IMLE training in two ways: per-stage supervision at each upsampling resolution, and a robust loss that handles mismatched nearest-neighbour pairs. Across CIFAR-10, CelebA-HQ, and ImageNet 256, these changes make ROMS-IMLE competitive in FID while attaining strong precision and recall among one-step methods. More broadly, our results show that a single-step generator like IMLE can match much of the quality and diversity of iterative models when it is trained with a spectrally-aware, per-stage signal.

Acknowledgement This research was enabled in part by NSERC, the Canada Foundation for Innovation, the Canada CIFAR AI Chairs program, the BC DRI Group and the Digital Research Alliance of Canada.

References

[1] Antonio Torralba and Aude Oliva . Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3):391, may 2003. doi: 10.1088/0954-898X/14/3/302. URL https://dx.doi.org/10.1088/0954-898X/14/3/302.

[2] Mehran Aghabozorgi, Shichong Peng, and Ke Li. Adaptive IMLE for few-shot pretraining-free generative modelling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 248–264. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/aghabozorgi23a.html.

[3] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 2006.

[4] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023. URL https://arxiv.org/abs/2209.15571.

[5] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks, 2017. URL https://arxiv.org/abs/1701.04862.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. URL https://arxiv.org/abs/1701.07875.

[7] Himanshu Arora, Saurabh Mishra, Shichong Peng, Ke Li, and Ali Mahdavi-Amiri. Multimodal shape completion via imle, 2021. URL https://arxiv.org/abs/2106.16237.

[8] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4331–4339, 2019.

[9] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate, 2019. URL https://arxiv.org/abs/1910.11626.

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL https://arxiv.org/abs/2104.14294.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[13] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

[14] Sander Dieleman. Diffusion is spectral autoregression, 2024. URL https://sander.ai/2024/09/02/spectral-autoregression.html.

[15] Ricard Durall, Avraam Chatzimichailidis, Peter Labus, and Janis Keuper. Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues, 2020. URL https://arxiv.org/abs/2012.09673.

[16] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models, 2024. URL https://arxiv.org/abs/2410.12557.

[17] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling, 2025. URL https://arxiv.org/abs/2505.13447.

[18] Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025.

[19] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

[20] Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. *arXiv preprint arXiv:2410.07369*, 2024.

[21] 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 *NIPS*, 2017.

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[24] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017. URL https://arxiv.org/abs/1703.06868.

[25] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation, 2024. URL https://arxiv.org/abs/2401.09603.

[26] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. URL https://arxiv.org/abs/1710.10196.

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. URL https://arxiv.org/abs/1812.04948.

[28] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. URL https://arxiv.org/abs/2206.00364.

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

[30] Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling, 2025. URL https://arxiv.org/abs/2502.09509.

[31] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[32] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[33] Grayson Lee, Minh Bui, Shuzi Zhou, Yankai Li, Mo Chen, and Ke Li. Implicit maximum likelihood estimation for real-time generative model predictive control, 2026. URL https://arxiv.org/abs/2603.13733.

[34] Ke Li and Jitendra Malik. Implicit maximum likelihood estimation, 2018. URL https://arxiv.org/abs/1809.09087.

[35] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. URL https://arxiv.org/abs/2201.03545.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

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

[39] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[40] Jeeseung Park and Younggeun Kim. Styleformer: Transformer based generative adversarial networks with style vector, 2022. URL https://arxiv.org/abs/2106.07023.

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[42] Krishan Rana, Robert Lee, David Pershouse, and Niko Suenderhauf. Imle policy: Fast and sample efficient visuomotor policy learning via implicit maximum likelihood estimation, 2025. URL https://arxiv.org/abs/2502.12371.

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[44] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. URL https://arxiv.org/abs/2202.00512.

[45] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets, 2022. URL https://arxiv.org/abs/2202.00273.

[46] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

[47] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

[48] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. URL https://arxiv.org/abs/2303.01469.

[49] D. J. Tolhurst, Y. Tadmor, and Tang Chao. Amplitude spectra of natural images. *Ophthalmic and Physiological Optics*, 12(2):229–232, April 1992. ISSN 1475-1313. doi: 10.1111/j.1475-1313.1992.tb00296.x. URL http://dx.doi.org/10.1111/j.1475-1313.1992.tb00296.x.

[50] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space, 2021. URL https://arxiv.org/abs/2106.05931.

[51] Chirag Vashist, Shichong Peng, and Ke Li. Rejection sampling imle: Designing priors for better few-shot image synthesis, 2024. URL https://arxiv.org/abs/2409.17439.

[52] Gregory K. Wallace. The jpeg still picture compression standard. *Commun. ACM*, 34(4):30–44, April 1991. ISSN 0001-0782. doi: 10.1145/103085.103089. URL https://doi.org/10.1145/103085.103089.

[53] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans, 2022. URL https://arxiv.org/abs/2112.07804.

[54] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation, 2022. URL https://arxiv.org/abs/2112.10762.

[55] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

[56] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Forty-second International Conference on Machine Learning*, 2025.

A Note on Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is widely used in generative modeling because it provides a principled method for fitting models to data by maximizing the likelihood that observed samples originate from the model's distribution. Consequently, samples generated from models optimized with MLE (and its variants) are, by design, expected to closely resemble the data points. Directly optimizing the likelihood is often computationally intractable, since the data is generally high-dimensional and the model's mapping from the prior to the data distribution is typically not surjective. Different generative models take different approaches to get around this issue.

Diffusion models, for example, circumvent this by maximizing a variational lower bound known as the Evidence Lower Bound (ELBO). Specifically, diffusion models factorize the generation process into multiple small, iterative transitions from a known prior distribution $p(\mathbf{x}_T)$. This can be written as follows:

$$\begin{aligned} p(\mathbf{x}_0 \mid \mathbf{x}_T) &= \int \cdots \int p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)\, p_\theta(\mathbf{x}_1 \mid \mathbf{x}_2) \cdots p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)\, \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_{T-1} \\ &= \int \cdots \int \left[\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \right] \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_{T-1} \end{aligned} \tag{5}$$

In contrast, Implicit Maximum Likelihood Estimation (IMLE) models the transition from the prior $\mathbf{x}_T$ to the data distribution $\mathbf{x}_0$ in a single step, implicitly optimizing the MLE without explicitly factorizing into intermediate distributions.

B Architecture and training details

We provide the detailed network architecture in Figure 9.
Our generator maps a latent code ($\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}), \mathbf{z} \in \mathbb{R}^{1024}$) to a style vector $\mathbf{w}$ via an MLP (mapping layer), which modulates each layer of a ConvNeXt-style decoder using Adaptive Instance Normalization (AdaIN)[24]. Starting from a learned constant feature map at low resolution, the decoder progressively upsamples the representation through a stack of residual ConvNeXt[36] blocks with per-resolution width control.

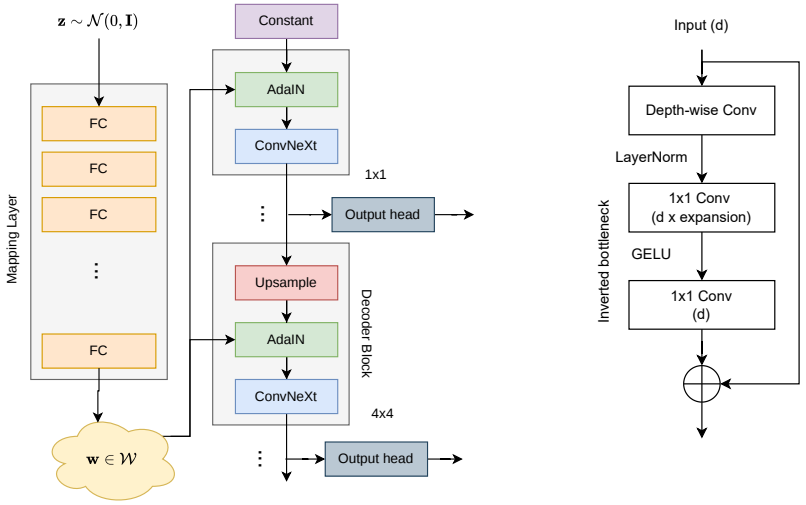
Training Details The training was done on NVIDIA L40S GPUs. The training time depends primarily on the size of the dataset. Default PyTorch implementation of bicubic interpolation was used for all resizing operations, as we saw that achieved the best results. We used a cosine scheduler with warmup for 100 iterations.

Pixel-based models We used a weighted combination of LPIPS [55] loss, DINO [11] loss and pixel loss for optimization, with weighting factors of 1.0, 1.0 and 0.1 respectively. Since DINO expects 224×224 inputs, all images were resized before being passed into the DINO network. LPIPS requires images to be of at least 32×32 resolution; hence images below this resolution were resized to 32×32 before being passed into the LPIPS network. The 32×32 variant (used for CIFAR-10) contains 110 million trainable parameters, while the 256×256 variant (used for CelebA-HQ) has 138 million. We do not use any data augmentation.

Latent-based models For ImageNet, we train our generator in the latent space of an EQ-VAE [30], which is kept frozen throughout training. The model contains 310 million trainable parameters. The per-stage training loss is the Geman-McClure loss applied to elementwise residuals computed directly on the latent representations. Per-stage targets are obtained by encoding the real image with the frozen EQ-VAE encoder and downsampling the resulting latent map with K_l to each stage's resolution.

We use rejection sampling to remove low-quality samples, with a different rejection criterion: at evaluation time, we compute for each generated sample the round-trip reconstruction cost of encoding and decoding it with the EQ-VAE. The distance between the generated sample and its round-trip reconstruction is measured in LPIPS [55] space. Low-quality samples incur a high round-trip cost, and we reject samples whose cost exceeds a fixed threshold, which removes roughly 5% of generated samples. We find that removing these samples improves FID. Reported ImageNet metrics are computed on samples that pass this filter.

We provide additional details about the training setup that are relevant for reproduction and for clarifying the experimental protocol.



(a) Network Architecture (b) Standard ConvNeXt block

Figure 9: (a) Network architecture, which comprises a mapping network, upsampling layers and ConvNeXt blocks. (b) Inner workings of a standard ConvNeXt block.

Budget for nearest-neighbour search. For each training round, we draw $m = 5n$ candidate latents, where n is the number of real images, giving 5 generated samples per real image on average. This setting is held fixed across all three datasets (CIFAR-10, CelebA-HQ, ImageNet 256).

Distance metric for nearest-neighbour selection. The distance $d(\cdot, \cdot)$ used in the $\arg\min_j$ step of Algorithm 1 differs between pixel-space and latent-space models. For our pixel-space models (CIFAR-10, CelebA-HQ), we use LPIPS [55] as the selection distance. For our latent-space model on ImageNet 256, we use the squared ℓ_2 distance in the EQ-VAE [30] latent space.

Robust loss usage. As shown in Algorithm 1, the robust loss is applied only during the optimization step (the per-stage training loss in Algorithm 1), not during nearest-neighbour selection. It wraps each component of our training loss: pixel residuals, LPIPS, and DINO for the pixel-based models, and latent residuals for the latent-based model. Selection always uses the unwrapped distances above.

Nearest-neighbour search at scale. We use FAISS-GPU to accelerate exact nearest-neighbour search across the m candidate samples.

Precision and recall. We report precision and recall using the standard implementation of Kynkäänniemi et al. [32], with the InceptionV3 feature extractor and $k = 3$ for the k-nearest-neighbour radius. This matches the protocol most commonly used in prior one-step generation literature.