What You Predict Shapes How You Memorize: Target-Parameterization and Memorization Dynamics in Flow Matching

Mohammad Al-Jaff [1] **Amir Mehrpanah** [1] **Gustav Eje Henter** [1] **Hossein Azizpour** [1 2] **Danica Kragic** [1]

Abstract

Flow matching models trained on finite data can eventually memorize parts or all of their training set: the finite-data optimum of the regression objective reproduces the empirical training distribution rather than the unknown data distribution. In practice, models often avoid this memorizing solution because of finite capacity, finite training time, architectural bias, and other implicit regularization. Separately, work on target parameterization has shown that predicting clean data $\boldsymbol{x}$, noise $\boldsymbol{\varepsilon}$, or velocity $\boldsymbol{v}$ can change empirical behavior and sample quality; we ask whether it also changes when and how flow matching models memorize. We study this in controlled regimes where memorization is observable, sweeping training-set size and model capacity while holding architecture, optimizer, training budget, sampler, and evaluation protocol fixed, and measuring checkpoint-level FID, nearest-neighbor memorization, and memorization at best FID. Across dataset sizes and model capacities, we find a consistent ordering: the three parameterizations memorize at different rates and to different degrees, with the ordering persisting even at matched sample quality, and the prediction target that is best for sample quality is not necessarily the one that memorizes least. Beyond its known role as a lever on generation quality, target parameterization also shapes when and how memorization emerges during training, a connection that, to our knowledge, has not been previously established.

1. Introduction

The promise of a generative model is to produce what it has never seen, having been shown only a finite collection of examples from a world it must learn to imitate. Flow matching and diffusion models pursue this by learning a time-dependent transformation from noise to data (Lipman et al., 2022; Tong et al., 2024; Albergo and Vanden-Eijnden, 2023), but the data distribution is only ever observed through a finite training set, and that finitude constrains everything that follows.

When generated samples are near-copies of training examples rather than genuinely new draws from the data distribution, this behavior is referred to as *memorization*. Memorization is not the same as poor sample quality. A model can generate poor samples without memorizing, and it can generate realistic samples while still reproducing parts of its training set. Sample quality alone therefore does not reveal whether a model generalizes.

On a finite training set, the globally optimal solution to the denoising regression objective itself reproduces the training distribution rather than the unknown data distribution, the Bayes-optimal solution is a memorizing one (Gu et al., 2025). Memorization is therefore not a pathology; it is the "best" a model can do given only the data it has. In practice, models trained on sufficiently large datasets generalize instead, precisely because finite capacity and finite training time prevent them from reaching this memorizing optimum (Bonnaire et al., 2026; Bertrand et al., 2026; Yoon et al., 2023). Generalization, in this view, arises from controlled regularization: the model deviates from the exact empirical solution, and that deviation is what produces novel samples rather than training-set replicas. Recent work has mapped this phenomenon in detail, showing that a generalization phase precedes a memorization phase during training, and that the gap between the two depends on dataset size, model capacity, and training duration (Yoon et al., 2023; Zhang et al., 2024; Bonnaire et al., 2026; George et al., 2025a). These studies vary data scale, capacity, and training time. They do not isolate the role of the training target itself.

Separately, recent and concurrent work has established that the choice of prediction target (clean data $\boldsymbol{x}$, noise $\boldsymbol{\varepsilon}$, or velocity $\boldsymbol{v}$) is not merely a change of coordinates under finite capacity. Li and He (2025) demonstrated that in the FM framework $\boldsymbol{x}$-prediction substantially outperforms $\boldsymbol{\varepsilon}$- and

[1]KTH Royal Institute of Technology, Stockholm, Sweden [2]Science for Life Laboratory, Stockholm, Sweden. Correspondence to: Mohammad Al-Jaff <mohamaj@kth.se>.

Published as a paper at the 1st FoGen workshop, ICML 2026, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

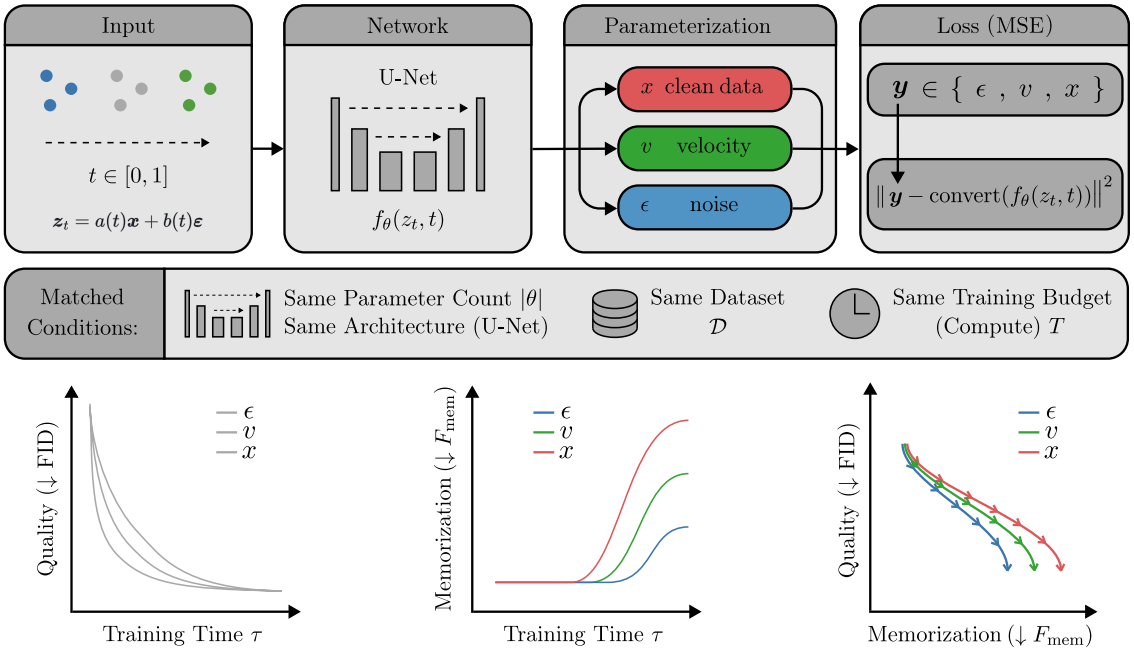

Figure 1. Under matched architecture, dataset, and training budget, the choice of prediction target ($\boldsymbol{x}$, $\boldsymbol{v}$, or $\boldsymbol{\varepsilon}$) changes not only sample quality but also when and how memorization emerges during training. **Top:** Experimental setup, a U-Net predicts in one of three target spaces; the output is converted to the configured loss space before computing the MSE. **Middle:** Controlled conditions held fixed across parameterizations. **Bottom:** The quantities we track over training. Parameterizations that reach comparable quality can differ substantially in memorization, and the target that is best for quality is not necessarily the one that memorizes least.

$\boldsymbol{v}$-prediction in terms of sample quality when the observed dimension far exceeds the intrinsic data dimension. They attribute this to the geometric structure of the targets. Natural data is approximately supported on a low-dimensional manifold embedded in a much larger ambient space (Peyré, 2009; Fefferman et al., 2016). Clean-data targets live near this manifold; noise and velocity targets span the full ambient space. Concurrent work by Jin and Wang (2026) formalizes a continuous target family and derives the quality-optimal target as a function of intrinsic and ambient dimension. Gagneux et al. (2026) complicate this picture: architecture locality and dataset size interact with target choice, and velocity prediction remains competitive with local architectures and sufficient data, even in high-dimensional settings. Together, these results establish parameterization as a lever on sample quality. None to our knowledge measure memorization.

This creates a gap between two active literatures. The memorization literature shows that generalization arises from controlled imperfection: the model does not exactly reach the empirical optimum, and this deviation is what produces novel samples. The parameterization literature shows that some targets make learning easier than others under finite capacity. But if an easier target lets the network approach the memorizing empirical solution faster, then the same property that makes a target quality-optimal may also erode the controlled imperfection that prevents memorization. Conversely, a harder target, one that fills the ambient space and is slower to learn, may preserve that imperfection longer. The quality literature and the memorization literature have not been connected, and it is not obvious which way the answer goes.

We ask whether target parameterization changes the memorization dynamics of flow matching. Specifically: under matched architecture, dataset size, optimizer, and training protocol, do formally equivalent parameterizations produce different trajectories toward memorization? This question is informative regardless of outcome. If memorization dynamics differ across parameterizations, then the training target is a previously unrecognized axis of control; one that is currently chosen based on quality alone. If they do not differ, then the practical quality differences observed across parameterizations do not route through memorization, and must be explained by other mechanisms.

We study this question using diagonal parameterization configurations: $\boldsymbol{x}/\boldsymbol{x}$, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ and $\boldsymbol{v}/\boldsymbol{v}$, where prediction and loss are computed in the same space, isolating the effect of the target before disentangling prediction space from loss space. We additionally include $\boldsymbol{x}/\boldsymbol{v}$, motivated by the findings of Li and He (2025). All comparisons are controlled: architecture, optimizer, training budget, and evaluation protocol are held fixed across parameterizations. We evaluate each con-

figuration using checkpoint-level FID, nearest-neighbour memorization fraction F_{mem}, quality-controlled memorization $F_{\mathrm{mem}}@\mathrm{FID}_{\mathrm{best}}$, and the generation and memorization timescales τ_{gen}, τ_{mem}, and their difference $\Delta\tau_{\mathrm{gen}\to\mathrm{mem}}$, across sweeps over dataset size and model capacity.

We identify target parameterization as a previously, as far as we know, unrecognized axis of memorization control in flow matching, alongside known axes such as dataset size, model capacity, training duration, data duplication, and conditioning (Yoon et al., 2023; Gu et al., 2025; Bonnaire et al., 2026; Carlini et al., 2023; Somepalli et al., 2023a;b; Wen et al., 2024). Our contributions are:

1. **Parameterization shapes memorization dynamics.** Under matched training conditions, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ memorizes latest and least, $\boldsymbol{x}/\boldsymbol{x}$ earliest and most, and $\boldsymbol{v}/\boldsymbol{v}$ lies between. This ordering holds in memorization amount, onset timing, and the generation-to-memorization training window and persists across dataset sizes, model capacities, and metric thresholds
2. **Quality and memorization can disagree.** We find that FID alone can miss memorization-relevant differences between target parameterizations. The target that achieves the best FID is not necessarily the one that memorizes least; evaluating generation quality alone can miss a memorization-relevant axis in model behavior. In our experiments, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ delays memorization compared to $\boldsymbol{x}/\boldsymbol{x}$ and $\boldsymbol{v}/\boldsymbol{v}$ while remaining competitive in FID.

In practice, the choice of parameterization has typically been guided by convention, often inheriting the default of the framework in which the model was introduced, by recent arguments favouring specific targets for sample quality (Li and He, 2025; Jin and Wang, 2026), or by empirical tuning. Memorization has not been part of this design consideration. Our results suggest it should be: parameterization is not only a lever on sample quality but also an axis that governs memorization dynamics: and the two criteria can disagree. Recent work argues that $\boldsymbol{x}$-prediction is preferable for sample quality in the flow matching setting (Li and He, 2025), yet we find that $\boldsymbol{x}/\boldsymbol{x}$ reaches memorization onset earliest and memorizes the most relative to $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ and $\boldsymbol{v}/\boldsymbol{v}$. Conversely, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ is the most memorization-resistant parameterization, while incurring comparatively little sample quality loss, and in some settings achieving better FID than the alternatives.

2. Related Work

Memorization as the finite-data optimum. On finite data, the ideal regression target of diffusion and flow matching is already a memorizing object. In diffusion, the empirical score is the score of a Gaussian mixture centered at the training samples; following it in the reverse process reproduces those samples (Kamb and Ganguli, 2025; Baptista et al., 2025). In flow matching, the closed-form empirical velocity has the same pathology: the induced ODE flows back to training points (Bertrand et al., 2026).

Why trained models can generalize before memorizing. Recent work explains practical generalization as a consequence of not fitting the empirical optimum exactly. Finite capacity, architecture, and finite training time all act as forms of regularization. Empirically, diffusion models exhibit a generation–memorization separation: useful samples appear before the model starts visibly reproducing individual training examples (Yoon et al., 2023; Bonnaire et al., 2026). In the two-timescale view of Bonnaire et al. (2026), the generation time τ_{gen} is nearly independent of dataset size, while the memorization time τ_{mem} grows roughly linearly with it. This creates a training window where models generate well without memorizing. We use this view as our reference point, but ask whether the window also depends on the target being predicted.

Geometry, architecture, and capacity. The same finite-data objective can lead to different learned solutions depending on the model class. The manifold hypothesis suggests that natural data occupy a low-dimensional subset of the ambient space (Peyré, 2009; Fefferman et al., 2016), and several works use this geometry to explain why generative models can avoid direct memorization. For convolutional models, locality and equivariance prevent learning the global empirical score and instead induce locally consistent patch mosaics (Kamb and Ganguli, 2025). Random-feature analyses give a complementary tractable picture, where memorization and generalization are controlled by the number of features, samples, and noise draws per datapoint (George et al., 2025a), with manifold structure further changing sample complexity (George et al., 2025b). In our experiments, we hold the architecture fixed and vary the target representation, so that changes in memorization can be attributed to the training target rather than to the model class.

Prediction targets under finite capacity. The common targets in diffusion and flow models, $\boldsymbol{x}$, noise $\boldsymbol{\varepsilon}$, and velocity $\boldsymbol{v}$, are algebraically related, but they are not equivalent learning problems once the network class, loss weighting, optimizer, and training budget are fixed (Salimans and Ho, 2022; Li and He, 2025). Recent work sharpens this distinction. Clean-data prediction can be easier in high dimension because $\boldsymbol{x}$ is aligned with the data manifold, while $\boldsymbol{\varepsilon}$ and $\boldsymbol{v}$ contain ambient noise directions (Li and He, 2025). Continuous target families further formalize how the quality-optimal target depends on intrinsic and ambient dimension (Jin and Wang, 2026). Complementary work shows that weighting,

parameterization, data size, and architectural locality interact, so no single target is universally best (Gagneux et al., 2026). Together, these results establish target parameterization as a lever on quality. We ask whether it is also a lever on memorization.

Concurrent work and our distinction. The closest flow-matching papers study the closed-form empirical velocity, target stochasticity, weighting, and parameterization (Bertrand et al., 2026; Gagneux et al., 2026; Jin and Wang, 2026). They explain why the empirical velocity memorizes, why target stochasticity is not the source of generalization, and why target choice affects quality. Separately, empirical work on copying, extraction, and mitigation in large diffusion models (Somepalli et al., 2023a;b; Carlini et al., 2023; Wen et al., 2024) motivates why memorization matters but does not isolate target parameterization as the controlled variable. None of these works measure how the choice of prediction target changes memorization dynamics; F_{mem} trajectories, quality-controlled memorization $F_{\mathrm{mem}}@\mathrm{FID}_{\mathrm{best}}$, or target-dependent generation and memorization timescales under otherwise matched conditions.

3. Background

3.1. Flow Matching

Let p_{data} denote the unknown data distribution, accessible only through a finite training set $\mathcal{D}_{\text{train}} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N} \subset \mathbb{R}^D$ of independent samples from p_{data}. Flow matching (FM) aims to construct/learn a time-dependent vector field $u_\theta : \mathbb{R}^D \times [0,1] \to \mathbb{R}^D$ that defines the ODE

$$\frac{d\boldsymbol{z}}{dt} = u_\theta(\boldsymbol{z}_t,\, t), \qquad \boldsymbol{z}_0 \sim p_0, \tag{1}$$

where $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a simple source distribution. The idea is that integrating from $t = 0$ to $t = 1$ transports samples from p_0 to an approximation of p_{data}. The ideal vector field u_t generates the marginal path p_t, but the corresponding FM loss

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,\, \boldsymbol{z}_t \sim p_t} \big\| u_\theta(\boldsymbol{z}_t, t) - u_t(\boldsymbol{z}_t) \big\|^2 \tag{2}$$

is intractable in practice, as u_t requires marginalising over all data, and assuming we have access to the underlying data distribution/density.

The above intractability is resolved by noticing that the loss objective containing the unconditional vector field shares the same minimiser as a conditional and tractable objective defined on individual data points (Lipman et al., 2022; Tong et al., 2024). Define the affine interpolant

$$\boldsymbol{z}_t = a(t)\, \boldsymbol{x} + b(t)\, \boldsymbol{\varepsilon}, \qquad \boldsymbol{x} \sim p_{\text{data}}, \quad \boldsymbol{\varepsilon} \sim p_0, \tag{3}$$

with differentiable schedules $a(t), b(t)$ satisfying $a(0) = 0,\ b(0) = 1,\ a(1) = 1,\ b(1) \approx 0$. The pathwise velocity is then in closed form: $\dot{\boldsymbol{z}}_t = \dot{a}(t)\, \boldsymbol{x} + \dot{b}(t)\, \boldsymbol{\varepsilon}$. The corresponding marginal vector field is obtained by averaging this pathwise velocity over all pairs $(\boldsymbol{x}, \boldsymbol{\varepsilon})$ that could produce the same state $\boldsymbol{z}$,

$$u_t(\boldsymbol{z}) = \mathbb{E}\big[\dot{a}(t)\, \boldsymbol{x} + \dot{b}(t)\, \boldsymbol{\varepsilon} \mid \boldsymbol{z}_t = \boldsymbol{z}\big]. \tag{4}$$

The conditional FM (CFM) objective,

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,\, \boldsymbol{x},\, \boldsymbol{\varepsilon}} \big\| u_\theta(\boldsymbol{z}_t, t) - \dot{a}(t)\, \boldsymbol{x} - \dot{b}(t)\, \boldsymbol{\varepsilon} \big\|^2, \tag{5}$$

is therefore fully tractable. Although $\mathcal{L}_{\text{CFM}}$ is not equal to $\mathcal{L}_{\text{FM}}$, the two objectives differ only by a θ-independent term. Hence they induce the same gradient with respect to θ and have the same minimiser (Lipman et al., 2022; Tong et al., 2024).

In this work we follow Lipman et al. (2022) and adopt the linear interpolant, $a(t) = t$ and $b(t) = 1 - t$, under which the path-wise velocity simplifies to the constant $\dot{\boldsymbol{z}}_t = \boldsymbol{x} - \boldsymbol{\varepsilon}$, and the CFM objective becomes

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,\, \boldsymbol{x},\, \boldsymbol{\varepsilon}} \big\| u_\theta(\boldsymbol{z}_t, t) - (\boldsymbol{x} - \boldsymbol{\varepsilon}) \big\|^2. \tag{6}$$

Table 1. Common regression targets as special cases of $\boldsymbol{y}_t^q = c_t^q\, \boldsymbol{x} + d_t^q\, \boldsymbol{\varepsilon}$.

Target q	$\boldsymbol{y}_t^q$	c_t^q	d_t^q
x	$\boldsymbol{x}$	1	0
ϵ	$\boldsymbol{\varepsilon}$	0	1
v_{FM}	$\boldsymbol{x} - \boldsymbol{\varepsilon}$	1	-1

3.2. Network output and loss space parameterization

When we speak of parameterization in this paper, we mean two distinct things. The first is the *prediction parameterization* π: what the output of the network f_θ represents. The second is the *loss parameterization* ℓ: the space in which the regression residual is computed. We denote a full parameterization configuration by $y = (\pi, \ell)$, inspired by Li and He (2025); Sun et al. (2025). We write f_θ for the network output, reserving u_θ for the velocity field recovered from it at inference.

Along the linear interpolant (3), any regression target can be written as an affine combination of $\boldsymbol{x}$ and $\boldsymbol{\varepsilon}$,

$$\boldsymbol{y}_t^q = c_t^q\, \boldsymbol{x} + d_t^q\, \boldsymbol{\varepsilon}, \tag{7}$$

where the coefficients (c_t^q, d_t^q) identify the representation q. The three canonical cases are listed in Table 1. In the linear FM setting these coefficients are time-independent constants, and the matched configurations x/x, ϵ/ϵ, v/v (written π/ℓ) reduce to directly having as regression targets $\boldsymbol{x}$, $\boldsymbol{\varepsilon}$, and $\boldsymbol{x} - \boldsymbol{\varepsilon}$ with no schedule dependence.

For a configuration $y = (\pi, \ell)$, the network outputs in prediction space π but the loss is evaluated in loss space ℓ. In the mismatched case $\pi \neq \ell$, since $(\boldsymbol{z}_t, \boldsymbol{y}_t^\pi)$ jointly determine $(\boldsymbol{x}, \boldsymbol{\varepsilon})$, a network prediction $\hat{\boldsymbol{y}}_t^\pi$ can always be linearly mapped to any other representation,

$$T_{\pi\to\ell}(\hat{\boldsymbol{y}}_t^\pi;\, \boldsymbol{z}_t, t) = A_t^{\pi\to\ell}\, \boldsymbol{z}_t + B_t^{\pi\to\ell}\, \hat{\boldsymbol{y}}_t^\pi, \tag{8}$$

where

$$A_t^{\pi\to\ell} = \frac{c_t^\ell d_t^\pi - d_t^\ell c_t^\pi}{\Delta_t^\pi}, \quad B_t^{\pi\to\ell} = \frac{d_t^\ell a_t - c_t^\ell b_t}{\Delta_t^\pi}, \tag{9}$$

and where $\Delta_t^\pi = a_t d_t^\pi - b_t c_t^\pi$.

The general training objective is then

$$\mathcal{L}^y(\theta) = \mathbb{E}_{t,\,\boldsymbol{x},\,\boldsymbol{\varepsilon}} \Big\| T_{\pi\to\ell}\big(f_\theta(\boldsymbol{z}_t, t);\, \boldsymbol{z}_t, t\big) - \boldsymbol{y}_t^\ell \Big\|^2, \tag{10}$$

which reduces to $\mathbb{E}_{t,\boldsymbol{x},\boldsymbol{\varepsilon}} \| f_\theta(\boldsymbol{z}_t, t) - c_t^\ell \boldsymbol{x} - d_t^\ell \boldsymbol{\varepsilon} \|^2$ in the matched case $\pi = \ell$. Our main experiments focus on the matched configurations $\mathcal{Y} = \{x/x,\, \epsilon/\epsilon,\, v/v\}$; we additionally include $\boldsymbol{x}/\boldsymbol{v}$ in select experiments as this setting has been shown to achieve very good quality performance relative to other parameterizations in Li and He (2025).

At inference, regardless of the configuration $y = (\pi, \ell)$ used during training, the network output is converted to the velocity field via $T_{\pi\to v}$, and an ODE solver integrates $\dot{\boldsymbol{z}}_t = u_\theta(\boldsymbol{z}_t, t)$ from $t = 0$ to $t = 1$ to generate samples.

3.3. Memorization and Quality Metrics

Memorization is measured via the nearest-neighbor ratio (Somepalli et al., 2023a; Bonnaire et al., 2026). For a generated sample $\hat{\boldsymbol{x}}$, let $d_k(\hat{\boldsymbol{x}})$ denote its distance to the k-th nearest neighbor in $\mathcal{D}_{\text{train}}$. The ratio

$$r(\hat{\boldsymbol{x}}) = \frac{d_1(\hat{\boldsymbol{x}})}{d_2(\hat{\boldsymbol{x}})} \tag{11}$$

is small when $\hat{\boldsymbol{x}}$ has a uniquely close match in training, the signature of memorization. Distances may be computed in pixel space or in a learned feature space (e.g. CLIP, DINO (Somepalli et al., 2023a;b)). Aggregating over M generated samples gives the memorization fraction

$$F_{\text{mem}}^\rho = \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}\big[r(\hat{\boldsymbol{x}}_m) < \rho\big], \tag{12}$$

where $\rho \in (0, 1)$ is a threshold hyperparameter. Since F_{mem}^ρ at a fixed checkpoint conflates memorization with generation quality, a more informative quantity is its value at peak quality,

$$F_{\text{mem}}@\text{FID}_{\text{best}} := F_{\text{mem}}^\rho\big(\tau^*\big), \qquad \tau^* = \arg\min_\tau \text{FID}(\tau), \tag{13}$$

reduces the confound that low memorization may reflect poor generation.

Beyond *how much* a model memorizes, a complementary question is *when*. Bonnaire et al. (2026) identify two boundaries in the (N, τ) plane (τ = gradient steps): a generation time τ_{gen}, where the model begins producing coherent samples, and a memorization time τ_{mem}, where outputs begin collapsing onto training points. They find $\tau_{\text{gen}} \sim \text{const}$ (independent of N) while $\tau_{\text{mem}} \propto N$, so the safe training window $\Delta\tau := \tau_{\text{mem}} - \tau_{\text{gen}}$ grows with dataset size. The exact value of τ_{mem} depends on what counts as "entering the memorization regime." Setting the threshold at the first nonzero F_{mem} is sensitive to sampling noise and isolated nearest-neighbor events; a higher threshold such as $F_{\text{mem}} \geq 0.01$ gives a more robust signal that the model has entered a visible memorization regime rather than producing occasional near-copies. We report results at a fixed threshold and verify that the parameterization ordering is stable across threshold choices (Appendix A.3). These boundaries are established for diffusion models; whether they carry over to flow matching, and whether parameterization $y = (\pi, \ell)$ shifts them, is the central empirical question of this work.

4. Experimental Setup

To investigate the effect of parameterization empirically, we monitor the established metrics of memorization and generalization under controlled conditions, where dataset preprocessing, flow path, sampler, evaluation protocol, and checkpoint metrics are fixed. We focus our study of the effect of target parameterization on the evolution of memorization along two axes: 1) training-set size, and 2) model capacity.

Implementation Details. All experiments use unconditional flow-matching models trained on CelebA 32×32 grayscale images, following Bonnaire et al. (2026). Images are center-cropped, resized to 32×32, converted to one channel, and normalized to $[-1, 1]$. For each seed, we draw a deterministic training subset from CelebA train split and a disjoint deterministic reference subset from the validation split. We use the convex interpolation $\boldsymbol{z}_t = t\boldsymbol{x} + (1-t)\boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $t \sim U[10^{-3}, 0.999]$. We compare the parameterizations x/x, ϵ/ϵ, and v/v, and include x/v as an ablation study. All models use a U-Net (Rombach et al., 2022) following the architecture of Bonnaire et al. (2026), and inference always converts the network output to a velocity field sampled with a 50-step Heun solver.

Axis 1: The Evolution of Memorization *wrt.* Dataset-size Under Different Parameterizations. To investigate the effect of target parameterization on memorization while the *dataset size* changes, we vary training-set size while holding

architecture, optimizer, and training budget fixed. We train on $N_{\text{train}} \in \{256, 512, \ldots, 16384\}$ using three seeds, with Adam at learning rate 10^{-4}, weight decay zero, batch size capped at 256, and a fixed 10^6-step training budget.

Axis 2: The Evolution of Memorization *wrt.* Model Capacity Under Different Parameterizations. To investigate the effect of target parameterization on memorization while the *model capacity* changes, We vary model capacity at $N_{\text{train}} \in \{256, 2048, 16384\}$ using three seeds, monitoring how parameterization affects memorization while model size varies. We use three U-Net capacities of approximately 3.9M, 15.6M, and 25.8M parameters, obtained by varying base channels, channel multipliers, and number of residual blocks.

Checkpoint evaluation. At each checkpoint, we generate 10,000 samples from a fixed Gaussian noise bank and evaluate them against 10,000 held-out CelebA validation images; the fixed noise bank makes trajectories comparable within each run. Nearest-neighbor memorization is computed in raw image space via Euclidean distance on flattened normalized images, and we report $F_{\text{mem}} = F_{\text{mem}}^{0.5}$ unless stated otherwise. For sample quality we report $\text{FID}(\tau)$ at every checkpoint; for quality-controlled memorization we report $F_{\text{mem}}@\text{FID}_{\text{best}}$, defined at the checkpoint with lowest FID per run, which prevents lower memorization from being explained by poor sample quality alone.

5. Results

Target parameterization shifts memorization across dataset size. We first check whether our flow-matching setup reproduces the expected dataset-size effect: at fixed model capacity, memorization should decrease as the training set grows. Figure 2 confirms this. Across all three matched parameterizations, increasing N_{train} pushes F_{mem} downward and delays the onset of memorization.

The key observation is that this curve is target-dependent. At the same N_{train}, $\boldsymbol{x}/\boldsymbol{x}$ memorizes earliest and most, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ memorizes latest and least, and $\boldsymbol{v}/\boldsymbol{v}$ usually falls between them. Target parameterization therefore does not only change sample quality; it shifts how the model moves from generation toward memorization under otherwise matched conditions. The full grid of per-(N_{train}, parameterization) trajectories, including all seven dataset sizes, is shown in Appendix A.2

Lower memorization is not worse generation. A model can have low F_{mem} simply because it produces poor samples, so we evaluate memorization at the best-FID checkpoint for each run. Figure 3 reports the best FID reached during training and F_{mem} at that checkpoint, asking how much each model memorizes when it produces its best samples. The memorization ordering remains visible at the best-FID checkpoint, especially before memorization saturates near zero at the largest dataset sizes: $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ has the lowest $F_{\text{mem}}@\text{FID}_{\text{best}}$, $\boldsymbol{x}/\boldsymbol{x}$ the highest, and $\boldsymbol{v}/\boldsymbol{v}$ lies between them. The FID panel tells a different story: $\boldsymbol{v}/\boldsymbol{v}$ and $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ reach comparable or better FID than $\boldsymbol{x}/\boldsymbol{x}$, while $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ remains the most memorization-resistant. The target that produces the best samples is therefore not necessarily the one that memorizes least in our controlled setting.

Target parameterization shifts the generation–memorization window. We summarize each training trajectory by the generation time τ_{gen}, the memorization time τ_{mem}, and their gap $\Delta\tau = \tau_{\text{mem}} - \tau_{\text{gen}}$. Figure 4 shows that τ_{gen} does not grow monotonically with N_{train}, while τ_{mem} depends strongly on both dataset size and parameterization.

For $\boldsymbol{x}/\boldsymbol{x}$, memorization appears early across most dataset sizes. For $\boldsymbol{v}/\boldsymbol{v}$, τ_{mem} shifts to later training at larger N_{train}. For $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, this shift occurs at smaller N_{train}, and many runs do not cross the memorization threshold within the training budget. These censored runs are plotted at the final checkpoint, so their true τ_{mem} may be larger than shown.

The gap $\Delta\tau$ inherits this ordering. At small N_{train}, the gap is negative for some parameterizations, meaning the model crosses the memorization threshold before reaching its best-FID region. As N_{train} grows, $\Delta\tau$ becomes positive and widens, but it does so earliest for $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$. The window in which a model generates well before visible memorization therefore depends on the prediction target, not only on dataset size. Note that since censored $\tau_m em$ values are lower bounds, true gaps may be larger. Appendix A.3 repeats this analysis under the stricter $F_{\text{mem}} > 0$ thresholds of Bonnaire et al. (2026) where it is observed that the ordering persists despite noisier estimates.

Ablation studies. The parameterization ordering is stable across model capacity and metric threshold. In capacity sweeps spanning 3.9M–25.8M parameters (Appendix A.4, Figure 9), larger models memorize more at fixed N_{train}, but the target-dependent separation is preserved. The ordering is likewise robust to the nearest-neighbour ratio threshold ρ (Appendix A.3, Figure 7). The off-diagonal $\boldsymbol{x}/\boldsymbol{v}$ configuration, which shares prediction space with $\boldsymbol{x}/\boldsymbol{x}$ but loss space with $\boldsymbol{v}/\boldsymbol{v}$, behaves closer to $\boldsymbol{x}/\boldsymbol{x}$ in memorization dynamics (Figure 5), suggesting that prediction space is the primary driver.

6. Discussion

What the results show. Our experiments show that target parameterization changes the route from generation to mem-

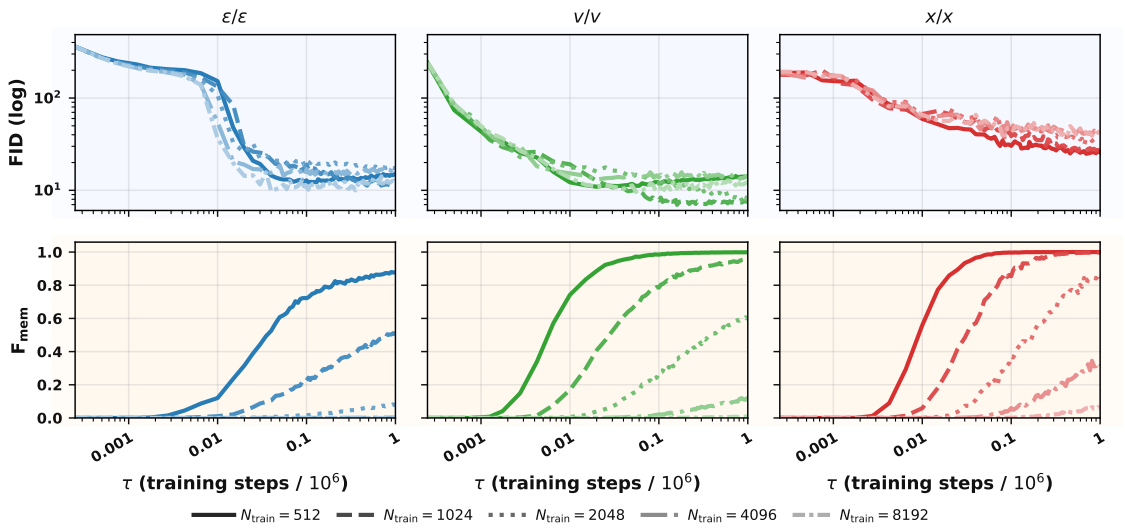

Figure 2. **FID and F_{mem} training trajectories across parameterizations and dataset sizes.** Top row: FID (log scale) vs. τ (training steps / 10^6). Bottom row: F_{mem} vs. τ. Columns correspond to $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, $\boldsymbol{v}/\boldsymbol{v}$, and $\boldsymbol{x}/\boldsymbol{x}$. Line style encodes N_{train} (solid to dash-dot, darkest to lightest).

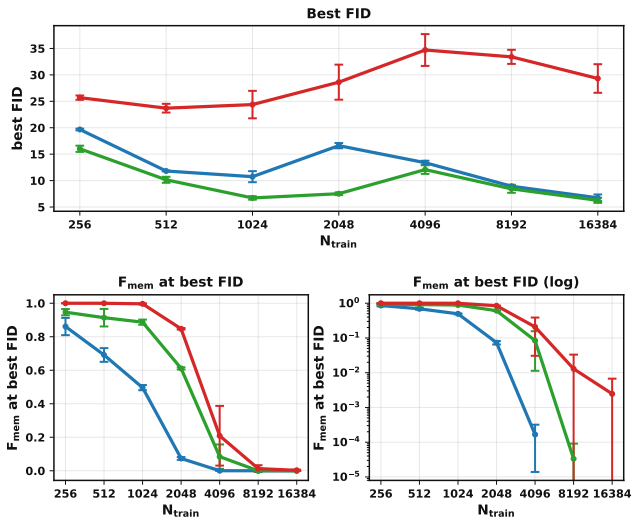

Figure 3. **Quality-controlled memorization across dataset size. Top:** best FID achieved during training. **Bottom left:** F_{mem} at the best-FID checkpoint (linear scale). **Bottom right:** same as bottom left, on a logarithmic scale to emphasize differences at large N_{train}. Curves show $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, $\boldsymbol{v}/\boldsymbol{v}$, and $\boldsymbol{x}/\boldsymbol{x}$ as a function of N_{train}. Error bars show ± 1 standard error over three seeds.

orization, not just sample quality. Under matched training conditions, the three matched parameterizations follow different memorization trajectories, differ in $F_{\mathrm{mem}}@\mathrm{FID}_{\mathrm{best}}$, and enter the memorization regime at different times. This suggests that target choice changes how a finite model moves from useful generation toward memorization.

The main empirical pattern is consistent across the settings we test: $\boldsymbol{x}/\boldsymbol{x}$ memorizes earlier and more strongly, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ memorizes later and less, and $\boldsymbol{v}/\boldsymbol{v}$ usually lies between them. The quality ordering is not the same as the memorization ordering: $\boldsymbol{v}/\boldsymbol{v}$ and $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ can reach comparable or better FID than $\boldsymbol{x}/\boldsymbol{x}$, while $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ remains the least memorizing target. This is the central point: target parameterization changes the quality–memorization tradeoff.

What the results do not yet disentangle. The targets $\boldsymbol{x}$, $\boldsymbol{\varepsilon}$, and $\boldsymbol{v}$ are linearly related along the flow path, but this algebraic relation does not make the finite learning problems identical. A finite network may learn one target faster, with different errors, or with a different tendency to fit sample-specific structure. At the same time, changing parameterization can also change the effective loss weighting over time. Our main experiments focus on matched prediction–loss settings, so they show that parameterization matters, but they do not fully separate the role of the network output space from the role of the loss space. The $\boldsymbol{x}/\boldsymbol{v}$ experiments are a first probe of this distinction; the full prediction–loss grid remains future work.

There are also limits to any single quality-controlled summary. Evaluating F_{mem} at the best-FID checkpoint reduces the most direct confound, namely that a bad generator may avoid nearest-neighbor memorization simply by producing poor samples. But this does not replace the full checkpoint trajectories, and it does not rule out subtler interactions between sample quality and memorization.

Interpreting the generation–memorization window. The timescale analysis should be read as a coarse summary of training dynamics. A positive $\Delta\tau$ means that a model reaches good sample quality before crossing the memoriza-

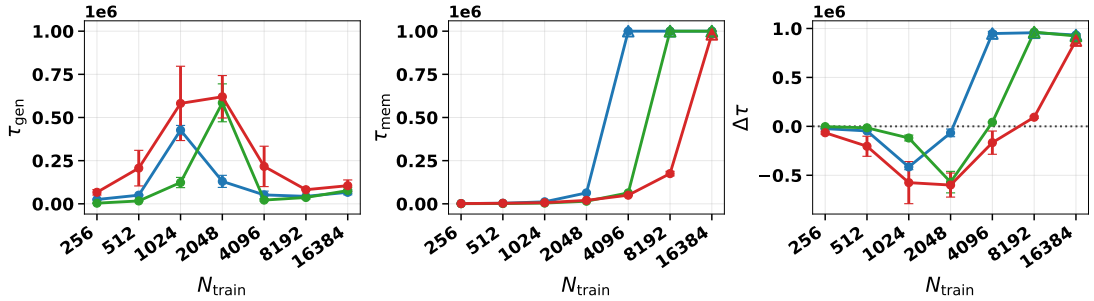

Figure 4. **Generation and memorization timescales across dataset size and parameterization. Left:** generation time τ_{gen}, the first checkpoint with FID $\leq 1.10\times$ run-best FID. Note: This is a post-hoc trajectory summary, not an online stopping rule. **Middle:** memorization time τ_{mem}, the first checkpoint with $F_{\text{mem}} \geq 0.01$. **Right:** the gap $\Delta\tau = \tau_{\text{mem}} - \tau_{\text{gen}}$; the dotted line marks $\Delta\tau = 0$. All three quantities are reported in training steps and shown as a function of N_{train}. Curves show $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, $\boldsymbol{v}/\boldsymbol{v}$, and $\boldsymbol{x}/\boldsymbol{x}$. Censored runs (no checkpoint crossing the threshold within the training budget) use the final checkpoint. Variation bars show ± 1 standard 'error' over three seeds.

tion threshold. A negative $\Delta\tau$ means that visible memorization appears before the best-FID checkpoint. This is not a contradiction: in small-data regimes, near-copies of training examples can still be data-like samples, so the model may improve FID partly by reproducing training examples. In that regime, generation and memorization are not cleanly separated phases. The important observation is that this separation changes with both dataset size and target parameterization.

A possible mechanism: target geometry as effective capacity. One possible explanation is target geometry. Clean-data targets lie closer to the data manifold, while noise and velocity targets contain ambient directions (Li and He, 2025; Peyré, 2009; Fefferman et al., 2016). If a target is easier or more aligned with data structure, a fixed architecture may have more usable capacity for fitting individual training examples. This can improve sample quality, but it can also make it easier to approach the finite-data memorizing solution. Under this interpretation, $\boldsymbol{x}/\boldsymbol{x}$ memorizes early because clean-data prediction is capacity-efficient, while $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ memorizes later because predicting full-ambient noise consumes capacity that would otherwise be available for fitting sample-specific structure. We do not prove this mechanism here. A natural next step is to test it with random-feature learning-curve analyses of target-parameterized flow matching (George et al., 2025a).

7. Limitations and Future Work

Our study is deliberately controlled: low-resolution grayscale CelebA, U-Net architecture close to the setting of Bonnaire et al. (2026), unconditional generation, and a fixed training budget. This makes the comparison clean, but limits scope. The present experiments do not yet cover high-resolution generation, larger or more varied datasets, modern larger-scale architectures, or conditional models. Extending the experiments along these axes is an immediate next step.

We mainly study diagonal configurations ($\boldsymbol{x}/\boldsymbol{x}$, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, $\boldsymbol{v}/\boldsymbol{v}$). This shows that target choice affects memorization, but does not fully separate the role of what the network predicts from where the loss is applied. Testing the full prediction–loss grid is left for future work.

Our memorization metric is based on nearest-neighbor ratios in image space; useful signal for detecting training-set replication, but does not exhaust all forms of memorization. Future work should combine nearest-neighbor tests with duplication sensitive feature-space metrics (Somepalli et al., 2023a;b).

The present work is also mainly empirical. A natural theoretical next step is to extend random-feature learning-curve analyses of denoising score matching (George et al., 2025a) to target-parameterized flow matching, similar to (Bonnaire et al., 2026) for the diffusion setting. Such models already provide a tractable way to study generalization and memorization under finite data, finite capacity, and finite training time. This would turn the empirical hypothesis of target-dependent memorization dynamics into a more explicit spectral picture.

8. Conclusion

Under matched training conditions, the choice of prediction target changes when and how much flow-matching models memorize. The ordering $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon} < \boldsymbol{v}/\boldsymbol{v} < \boldsymbol{x}/\boldsymbol{x}$ in memorization onset and degree holds under quality-controlled evaluation and across model capacities. Target parameterization is therefore not only a quality lever but also an axis of memorization control, and the two criteria can disagree. Since parameterization is a design choice made before training and is currently guided almost entirely by sample quality, our results suggest memorization should be part of that decision.

References

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

Ricardo Baptista, Agnimitra Dasgupta, Nikola B. Kovachki, Assad Oberai, and Andrew M. Stuart. Memorization and regularization in generative diffusion models, 2025. URL https://arxiv.org/abs/2501.15785.

Quentin Bertrand, Anne Gagneux, Mathurin Massias, and Rémi Emonet. On the closed-form of flow matching: Generalization does not arise from target stochasticity. *Advances in neural information processing systems*, 38: 8522–8549, 2026.

Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. *Advances in Neural Information Processing Systems*, 38:141266–141286, 2026.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium*, pages 5253–5270, 2023.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Anne Gagneux, Ségolène Martin, Rémi Gribonval, and Mathurin Massias. Training flow matching: The role of weighting and parameterization. *arXiv preprint arXiv:2603.06454*, 2026.

Anand Jerry George, Rodrigo Veiga, and Nicolas Macris. Denoising score matching with random features: Insights on diffusion models from precise learning curves, 2025a. URL https://arxiv.org/abs/2502.00336.

Anand Jerry George, Rodrigo Veiga, and Nicolas Macris. Analysis of diffusion models for manifold data. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pages 1–6. IEEE, 2025b.

Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=D3DBqvSDbj. Accepted by TMLR.

Qing Jin and Chaoyang Wang. Revisiting diffusion model predictions through dimensionality, 2026. URL https://arxiv.org/abs/2601.21419.

Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. In *Proceedings of the Forty-Second International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2412.20292.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2025. URL https://arxiv.org/abs/2511.13720.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2022. URL https://arxiv.org/abs/2210.02747.

Gabriel Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2):249–260, 2009.

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

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

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? Investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pages 47783–47803, 2023b.

Qiao Sun, Zhicheng Jiang, Hanhong Zhao, and Kaiming He. Is noise conditioning necessary for denoising generative models?, 2025. URL https://arxiv.org/abs/2502.13129.

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2302.00482.

Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=84n3UwkH7b.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K. Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML Workshop on Structured Probabilistic Inference and Generative Modeling*, 2023. URL https://openreview.net/forum?id=shciCbSk9h.

Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 60558–60590. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhang24cn.html.

A. Appendix

A.1. Training Trajectories I

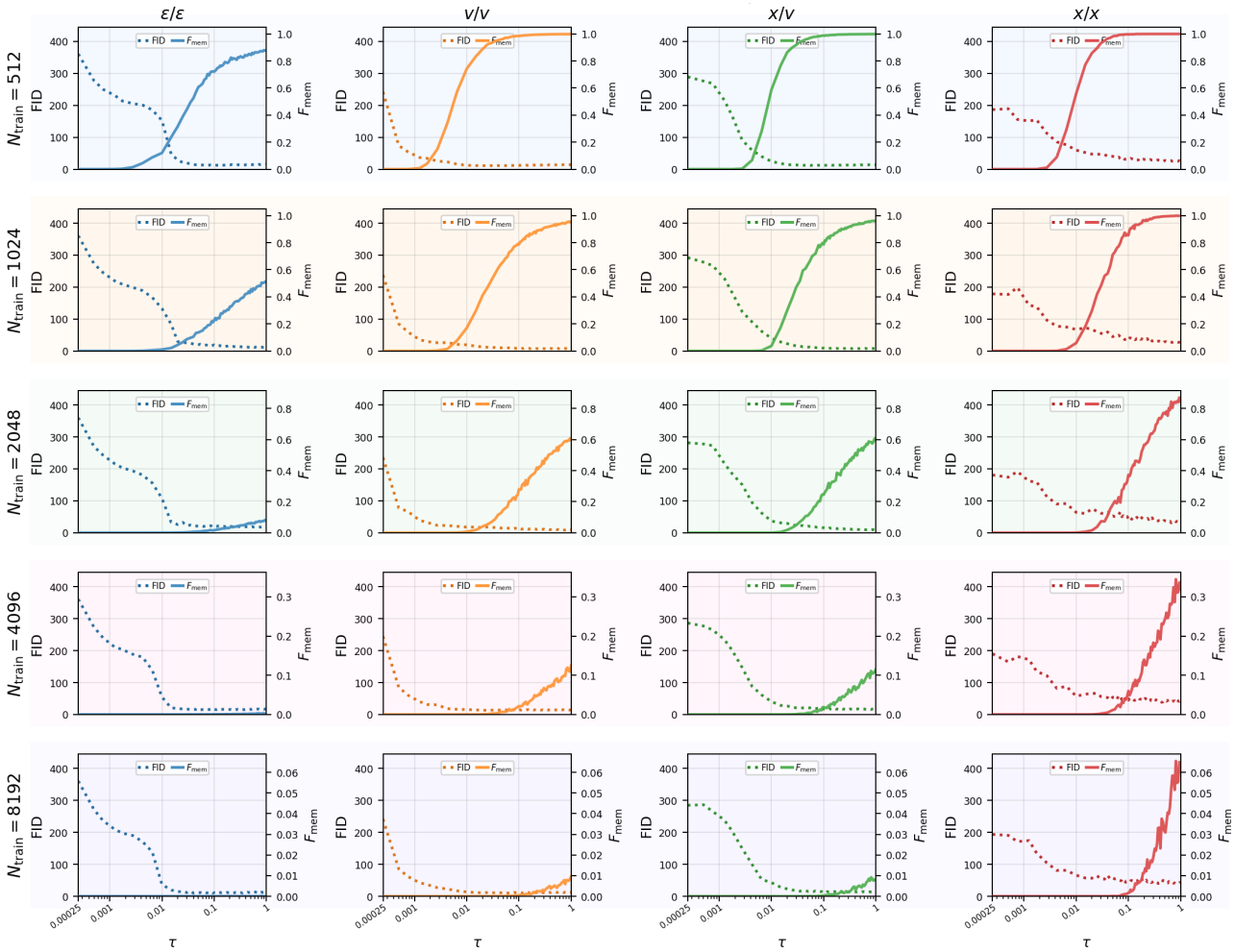

Figure 5. Joint FID and F_{mem} trajectories including the off-diagonal parameterization $\boldsymbol{x}/\boldsymbol{v}$. Layout follows Figure 6: FID (dotted, left axis) and F_{mem} (solid, right axis) over normalised training time τ. Columns correspond to $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, $\boldsymbol{v}/\boldsymbol{v}$, $\boldsymbol{x}/\boldsymbol{v}$, and $\boldsymbol{x}/\boldsymbol{x}$; rows correspond to $N_{\text{train}} \in \{512, 1024, 2048, 4096, 8192\}$. The F_{mem} axis range shrinks with increasing N_{train}. Seed-averaged over three seeds. The $\boldsymbol{x}/\boldsymbol{v}$ configuration behaves similarly to $\boldsymbol{x}/\boldsymbol{x}$ in memorization onset and magnitude, consistent with both sharing $\boldsymbol{x}$-prediction as their network output: the network predicts clean data in both cases, differing only in the space where the loss residual is computed. This suggests that the prediction space, rather than the loss space, is the primary driver of memorization dynamics. The ordering $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon} < \boldsymbol{v}/\boldsymbol{v} \lesssim \boldsymbol{x}/\boldsymbol{v} \approx \boldsymbol{x}/\boldsymbol{x}$ in memorization onset is visible across dataset sizes, though the separation between $\boldsymbol{x}/\boldsymbol{v}$ and $\boldsymbol{x}/\boldsymbol{x}$ is small relative to seed variance.

A.2. Training Trajectories II

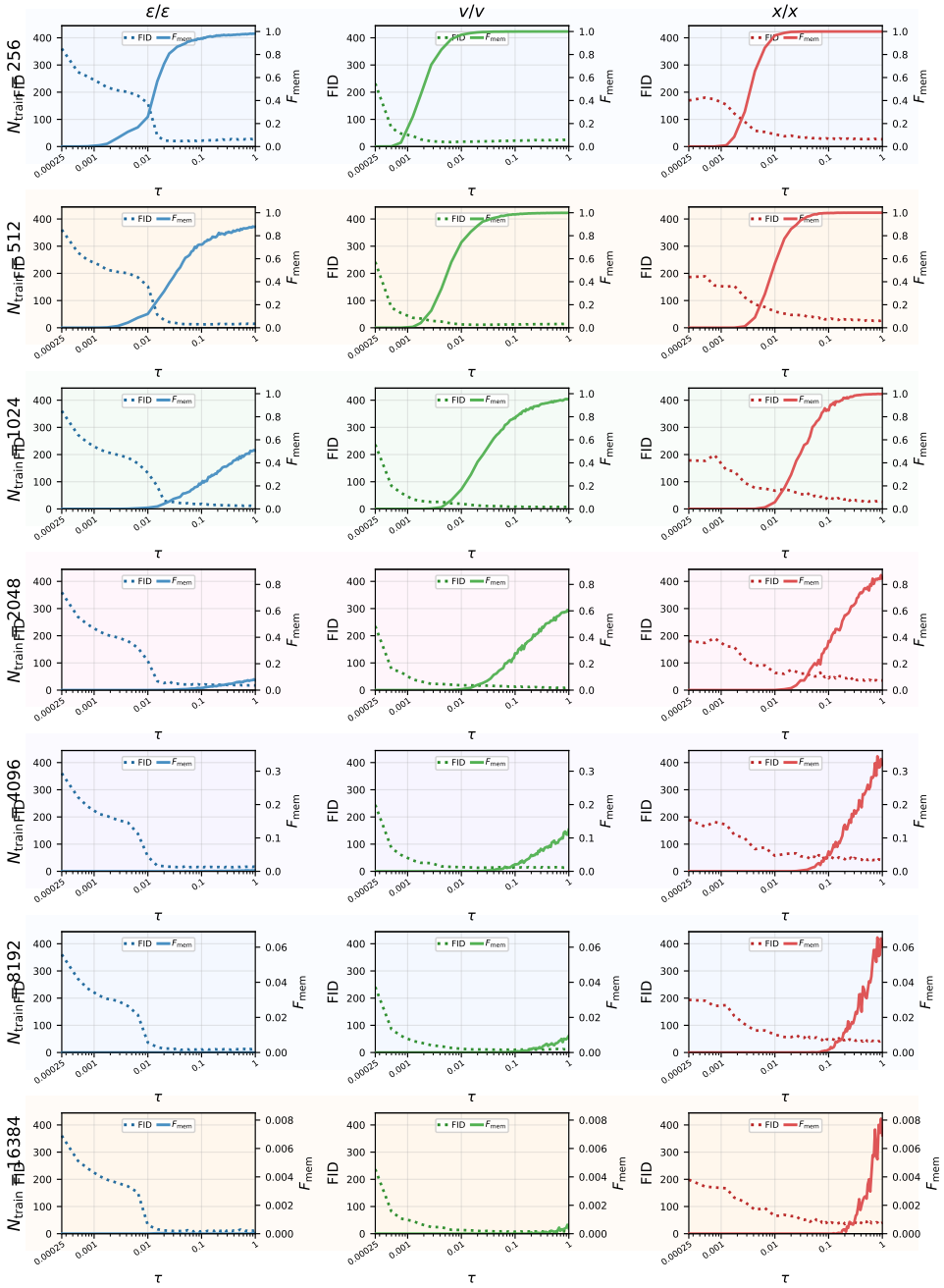

Figure 6. Joint FID and F_{mem} trajectories for each (parameterization, N_{train}) combination. FID (dotted, left axis) and F_{mem} (solid, right axis) over normalised training time τ. Columns correspond to $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, $\boldsymbol{v}/\boldsymbol{v}$, and $\boldsymbol{x}/\boldsymbol{x}$; rows correspond to $N_{\text{train}} \in \{256, 512, 1024, 2048, 4096, 8192, 16384\}$. The F_{mem} axis range shrinks with increasing N_{train}, reflecting the overall decrease in memorization at larger dataset sizes. Seed-averaged over three seeds. At small N_{train} (top rows), all three parameterizations reach high F_{mem} quickly, but $\boldsymbol{x}/\boldsymbol{x}$ rises earliest and steepest while $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ rises latest. As N_{train} grows, the separation becomes more pronounced: at $N_{\text{train}} = 4096$, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ reaches only $F_{\text{mem}} \approx 0.1$ by the end of training while $\boldsymbol{x}/\boldsymbol{x}$ reaches ≈ 0.3; at $N_{\text{train}} = 8192$ and 16384, all parameterizations show negligible memorization, but $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ remains the lowest. The FID trajectories show that all three parameterizations reach comparable best-FID values at each N_{train}, confirming that the memorization differences are not explained by differences in sample quality. Notably, for $\boldsymbol{x}/\boldsymbol{x}$ at small N_{train}, the F_{mem} rise begins before or near the FID minimum, whereas for $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ the FID minimum is reached well before visible memorization onset.

A.3. Threshold and Timescale Sensitivity

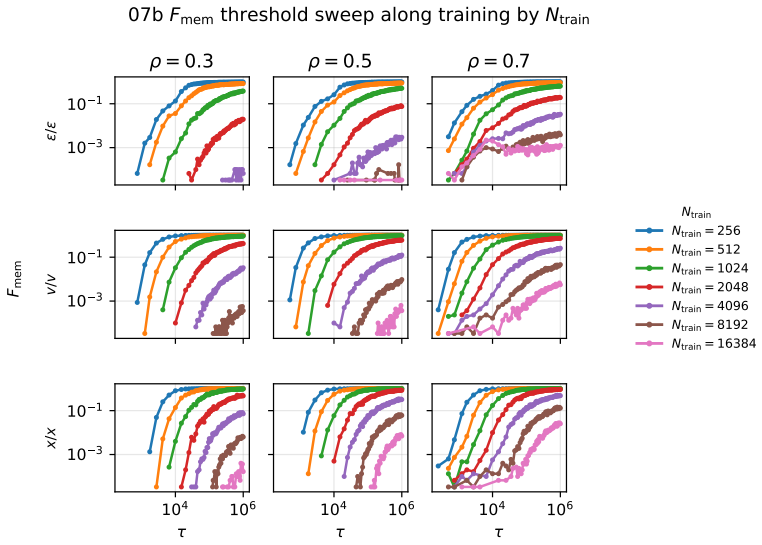

Figure 7. Sensitivity of F_{mem} trajectories to the nearest-neighbour ratio threshold ρ. Rows: parameterization ($\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$, $\boldsymbol{v}/\boldsymbol{v}$, $\boldsymbol{x}/\boldsymbol{x}$). Columns: threshold $\rho \in \{0.3,\ 0.5,\ 0.7\}$. Line colour encodes dataset size from $N_{\text{train}} = 256$ (darkest) to $N_{\text{train}} = 16384$ (lightest). All axes are log-scaled. Lowering ρ to 0.3 reduces overall F_{mem} levels (requiring stricter nearest-neighbour matches to count as memorized), while raising ρ to 0.7 increases them, but the relative ordering across parameterizations and dataset sizes is stable across all three thresholds. In particular, at each ρ, $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ consistently shows later onset and lower saturation levels than $\boldsymbol{v}/\boldsymbol{v}$ and $\boldsymbol{x}/\boldsymbol{x}$ at matched N_{train}. The dataset-size ordering (smaller N_{train} memorizes more and earlier) is likewise preserved across thresholds. This confirms that the reported parameterization ordering is not an artefact of the specific threshold $\rho = 0.5$ used in the main text.

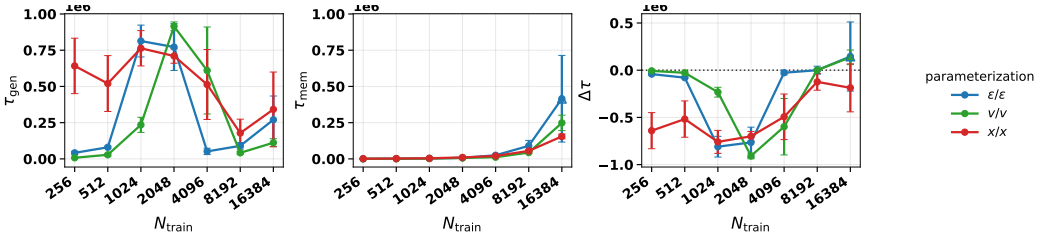

Figure 8. Generation and memorization timescales using the original definitions of Bonnaire et al. (2026). **Left:** generation time τ_{gen}, defined as the first checkpoint attaining the run-minimum FID (i.e., the best-FID checkpoint itself). **Middle:** memorization time τ_{mem}, defined as the first checkpoint with $F_{\text{mem}} > 0$ (any nonzero memorization). **Right:** the gap $\Delta\tau = \tau_{\text{mem}} - \tau_{\text{gen}}$; the dotted line marks $\Delta\tau = 0$. Curves show $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ (blue), $\boldsymbol{v}/\boldsymbol{v}$ (orange), $\boldsymbol{x}/\boldsymbol{v}$ (green), and $\boldsymbol{x}/\boldsymbol{x}$ (red). Error bars show ± 1 standard error over three seeds. Censored runs (no checkpoint crossing the memorization threshold within the training budget) use the final checkpoint. Compared to the relaxed definitions used in the main text, the stricter $F_{\text{mem}} > 0$ threshold makes τ_{mem} noisier—a single near-copy in 10,000 generated samples suffices to trigger onset, producing larger variation bars and less smooth curves, particularly at intermediate N_{train}. Despite this noise, the broad ordering persists: $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ tends toward the largest τ_{mem} and most positive $\Delta\tau$, while $\boldsymbol{x}/\boldsymbol{x}$ shows the earliest memorization onset. The $\Delta\tau$ panel shows that most configurations have negative gaps at small N_{train} (memorization precedes best quality), with the gap turning positive as N_{train} grows—and this transition occurs at smaller N_{train} for $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ than for $\boldsymbol{x}/\boldsymbol{x}$, consistent with the main-text results under the relaxed threshold.

A.4. Capacity Sweep

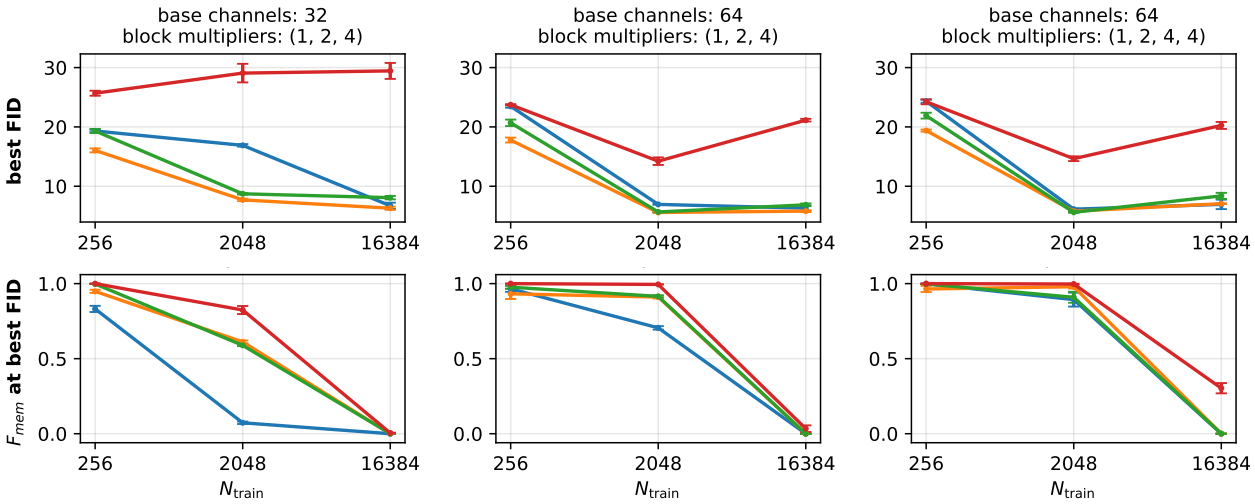

Figure 9. Capacity sweep at $N_{\text{train}} \in \{256,\ 2048,\ 16384\}$ across three U-Net configurations of increasing capacity (columns, left to right): base channels 32 with multipliers $(1, 2, 4)$ (≈3.9M parameters), base channels 64 with multipliers $(1, 2, 4)$ (≈15.6M parameters), and base channels 64 with multipliers $(1, 2, 4, 4)$ (≈25.8M parameters). **Top:** best FID achieved during training. **Bottom:** F_{mem} at the best-FID checkpoint. Colours: $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ (blue), $\boldsymbol{v}/\boldsymbol{v}$ (orange), $\boldsymbol{x}/\boldsymbol{v}$ (green), $\boldsymbol{x}/\boldsymbol{x}$ (red). Error bars: ± 1 standard error over three seeds. At $N_{\text{train}} = 256$, all parameterizations memorize heavily ($F_{\text{mem}} \approx 0.8$–$1.0$) regardless of capacity, though $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ remains slightly lower than the others.

At $N_{\text{train}} = 2048$, the parameterization ordering becomes clearly visible: $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ shows the lowest F_{mem}@FID$_{\text{best}}$ across all three capacities, while $\boldsymbol{x}/\boldsymbol{x}$ shows the highest. Increasing capacity from 3.9M to 25.8M raises memorization for all parameterizations at this dataset size, but the relative ordering is preserved. At $N_{\text{train}} = 16384$, memorization drops to near zero for all parameterizations at the two larger capacities, while $\boldsymbol{x}/\boldsymbol{x}$ retains residual memorization at the smallest capacity. The FID panels show that $\boldsymbol{\varepsilon}/\boldsymbol{\varepsilon}$ and $\boldsymbol{v}/\boldsymbol{v}$ achieve comparable or better FID than $\boldsymbol{x}/\boldsymbol{x}$ across most settings, with $\boldsymbol{x}/\boldsymbol{x}$ notably underperforming at larger N_{train}—the parameterization that memorizes most also tends to produce worse samples in the data-rich regime.