UNFOLDING GENERATIVE FLOWS WITH KOOPMAN OPERATORS: FAST AND INTERPRETABLE SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous Normalizing Flows (CNFs) offer elegant generative modeling but remain bottlenecked by slow sampling: producing a single sample requires solving a nonlinear ODE with hundreds of function evaluations. Recent approaches such as Rectified Flow and OT-CFM accelerate sampling by straightening trajectories, yet the learned dynamics remain nonlinear black boxes, limiting both efficiency and interpretability. We propose a fundamentally different perspective: globally linearizing flow dynamics via Koopman theory. By lifting Conditional Flow Matching (CFM) into a higher-dimensional Koopman space, we represent its evolution with a single linear operator. This yields two key benefits. First, sampling becomes one-step and parallelizable, computed analytically via the matrix exponential. Second, the Koopman operator provides a spectral blueprint of generation, enabling novel interpretability through its eigenvalues and modes. We derive a practical, simulation-free training objective that enforces infinitesimal consistency with the teacher's dynamics and show that this alignment preserves fidelity along the full generative path, distinguishing our method from boundary-only distillation. Empirically, our approach achieves competitive sample quality with dramatic speedups, while uniquely enabling spectral analysis of generative flows.

1 INTRODUCTION

While classic generative models like VAEs Kingma & Welling (2014) and GANs Goodfellow et al. (2014) offer fast, interpretable sampling, they have been surpassed in sample fidelity by dynamical system-based approaches like Diffusion Models Ho et al. (2020); Song et al. (2020) and Continuous Normalizing Flows (CNFs) Chen et al. (2018). This leap in quality, however, comes at the cost of slow, iterative sampling and limited interpretability.

For both model families, sampling is an iterative and slow process. Diffusion models learn to iteratively denoise data and therefore require multiple evaluations to generate samples, while sampling CNFs requires solving an ODE. In the case of CNFs, recent work has focused on accelerating sampling, with approaches such as Rectified Flow (Liu et al., 2023a) and Optimal Transport Conditional Flow Matching (Tong et al., 2024; Pooladian et al., 2023) that learn straighter generative paths. These methods successfully reduce the computational cost of generation while maintaining similar fidelity; however, they do not address the sampling process's lack of interpretability. This flaw limits our ability to understand *how* the model generates data, trust its outputs, and meaningfully control the generation process.

In this work, we address the challenges of slow sampling and limited interpretability in generative models grounded in dynamical systems. We build on Koopman operator theory, a classical framework for linearizing complex dynamical systems (Koopman, 1931; Mezić, 2005; Brunton et al., 2022). Originally developed in the 1930s, this theory has seen a resurgence in recent years thanks to machine learning methods that learn finite-dimensional approximations of the operator from data (Brunton et al., 2022; Bevanda et al., 2021). Neural network–based approaches such as Koopman autoencoders (Lusch et al., 2018; Otto & Rowley, 2019; Azencot et al., 2020) have successfully learned linear embeddings for complex systems in fields like fluid dynamics (Rowley et al., 2009) and molecular dynamics (Klus et al., 2018). We apply this approach to the dynamics of a pre-trained CNF, learning a latent space in which the dynamics evolve linearly under a corresponding learned linear operator (Lusch et al., 2018; Azencot et al., 2020). This transformation provides two key advantages:

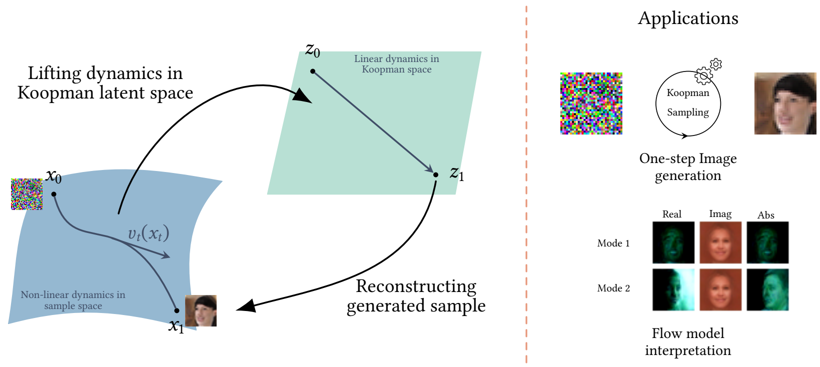


Figure 1: Simple illustration of our approach: We propose to apply Koopman theory to the dynamics of generative modeling from continuous normalizing flow models. We learn a Koopman latent space and its linear dynamics from the non-linear original CNF model. This approach presents two direct applications: one-step sampling and flow model interpretability.

1. **Generative process decomposition:** The learned Koopman operator acts as an interpretable blueprint of the generative process. We show that either the learned canonical frame of the Koopman latent space, or the eigendecomposition of the Koopman operator reveal semantic components of the dynamics. This allows for an unprecedented analysis of how CNF models constructs data from noise.

2. **One-Step Analytical Sampling:** A direct consequence of this linearization is that the solution to the generative ODE becomes analytical, given by a matrix exponential. This allows us to map noise to a data sample in a single, parallelizable step, eliminating the iterative sampling cost entirely.

Our core contribution is a practical, simulation-free training objective that learns this Koopman representation. We theoretically prove that naïve supervision strategies yield suboptimal objectives and impractical training processes. Crucially, we derive an efficient supervision strategy that constrains the learned linear dynamics to stay consistent with the teacher model's vector field along the *entire* generative path. We show that this can be enforced while remaining simulation-free, inheriting the properties of the underlying Continuous Flow Matching model. This distinguishes our approach from standard distillation methods, that only match the start and end points of the trajectory, while incurring only a moderate additional computational cost. Specifically, our contributions are:

- We introduce a novel framework **for learning** a global Koopman linearization of the non-autonomous dynamics in Conditional Flow Matching models.

- We derive a practical, simulation-free training objective that enforces consistency along the full generative trajectory, yielding a **full** linearization rather than mere boundary-focused distillation.

- We demonstrate empirically that our method achieves competitive one-step sampling performance while uniquely enabling spectral analysis, disentangled generative control, and **improved robustness in** downstream tasks.

2 Related Work

Our work connects four main areas: flow-based generative models, methods for accelerated sampling, Koopman operator theory for dynamical systems, and interpretability in generative modeling. We defer a formal introduction of Koopman operator theory to Section 3.2. For an overview of the field, we urge the interested reader to refer to the excellent introduction by Brunton et al. (2022).

2.1 Flow-Based Generative Models

Flow-based models learn an invertible mapping between a data distribution and a simple base distribution, offering tractable likelihoods (Dinh et al., 2014; 2017; Kingma & Dhariwal, 2018). Continuous Normalizing Flows (CNFs) parameterize this map as the solution to an ODE (Chen et al., 2018). Although powerful, training early CNFs was often unstable and computationally intensive. Conditional Flow Matching (CFM) represents a major step forward, providing a stable and efficient simulation-free training objective by regressing a neural network to a conditional vector field (Lipman et al., 2023; Tong et al., 2023; Liu et al., 2023b). However, while these models have achieved high accuracy for generative modeling, their sampling process remains inherently slow, opening the way for distilled models for faster sampling.

2.2 Accelerated and One-Step Sampling

The slow and iterative sampling of CNFs has motivated extensive research into acceleration. One popular direction, which includes Rectified Flow (Liu et al., 2023a) and OT-CFM (Pooladian et al., 2023), regularizes the learned ODE to have straighter trajectories, thus requiring fewer discretization steps. Another direction uses knowledge distillation to train a separate student model capable of single-step generation. This includes Consistency Models (Song et al., 2023) and other distillation techniques (Salimans & Ho, 2022; Luo et al., 2023; Liu et al., 2025). Although these methods achieve remarkable speed, they typically produce a compressed, black-box sampler that does not offer the interpretability or analytical control that our Koopman framework provides.

We also note that concurrently with our work, Berman et al. (2025) propose a Koopman-based generative model that learns a discrete-time Koopman operator, mapping noisy samples at $t = 0$ directly to target data at $t = 1$. While their approach is primarily positioned as an enhancement to diffusion models (though not exclusive to them), *our work focuses on conditional flow matching*, framing the problem as supervised learning of vector fields over time. In contrast to their discrete formulation, *we explicitly model the full continuous-time dynamics* by learning the Koopman generator, granting access to the entire latent flow from $t = 0$ to $t = 1$.

2.3 Interpreting and Explaining Generative Models

While methods exist for interpreting the latent spaces of classic models, such as VAEs and GANs, extending these powerful editing techniques to modern, iterative models like diffusion and flows has proven challenging due to their complex dynamics. Existing approaches for these models are often more complicated than the earlier methods Kwon et al. (2022); Yang et al. (2023); Meng et al. (2022); Kulikov et al. (2024), in addition to lacking the conceptual clarity of the latter. In contrast, our work offers a direct path to interpretability by learning a global linearization of the generative dynamics, which naturally yields a simple and editable latent space. A more detailed review of interpretability methods is provided in Appendix F.

3 Mathematical Background

3.1 Conditional Flow Matching

A Continuous Normalizing Flow (CNF) maps a prior distribution p_0 to a data distribution p_1 by solving the ODE

$$\frac{dx_t}{dt} = v_t(x_t) \tag{1}$$

, where v_t is a time-dependent vector field Chen et al. (2018). A naive regression loss to learn v_t is intractable, as both the true field v_t and the marginal path distribution p_t are unknown Lipman et al. (2023). Conditional Flow Matching (CFM) provides a tractable, simulation-free objective by regressing a neural network v_θ onto a *conditional* velocity field $u_t(x_t|x_1)$.

While effective, sampling from a trained CFM model requires numerically integrating its ODE via $x_1 = x_0 + \int_0^1 v_\theta(s, x_s)ds$, a slow process with potentially many function evaluations Chen et al. (2018). However, if the dynamics were linear, i.e., of the form $\frac{dx_t}{dt} = Ax_t$, sampling would become a single, analytical step: $x_t = e^{At}x_0$ that can be solved via matrix exponentiation. This vast efficiency gap motivates our core objective: to find a global linearization of the learned CFM dynamics.

3.2 KOOPMAN THEORY FOR AUTONOMOUS SYSTEMS

Koopman theory provides a powerful framework for globally linearizing nonlinear dynamical systems (Koopman, 1931; Mezić, 2005; Brunton et al., 2022). The central idea is to shift perspective from the finite-dimensional state space, where dynamics are nonlinear, to the infinite-dimensional space of functions - referred to as “observables” - where the dynamics become linear.

Formally, consider an autonomous dynamical system $\frac{dx_t}{dt} = v(x_t)$. This system induces a *flow map* F_t that advances an initial state x to its value at time t, namely $x_t = F_t(x)$, along the trajectories defined by v. Let $g : \mathbb{R}^d \to \mathbb{R}$ be an *observable function* on the state space. Given an initial state x, we define the *Koopman operator* $\mathcal{K}_t$ on the space of observables, denoted $\mathcal{G}(\mathbb{R}^d)$, which evolves observables along the trajectories generated by the vector field v:

$$\mathcal{K}_t g(x) := (g \circ F_t)(x) = g(F_t(x)) = g(x_t). \tag{2}$$

Koopman theory builds on the fact that this operator is trivially linear (regardless of the non-linearity of F_t) due to the linearity of the composition of functions: $K_t(g_1 + g_2)(x) = (g_1 + g_2) \circ F_t(x) = g_1 \circ F_t(x) + g_2 \circ F_t(x) = \mathcal{K}_t g_1(x) + \mathcal{K}_t g_2(x)$, for all observables g_1, g_2.

Taking the Lie derivative, we can the define the **Koopman generator**, $\mathcal{L}$, such that $\mathcal{L}g := \lim_{t\to 0} \frac{\mathcal{K}_t g - g}{t}$, and one can show that Brunton et al. (2022)

$$\mathcal{L}g = \frac{dg}{dt} = \nabla_x g(x) \cdot v(x), \tag{3}$$

which is also trivially linear in g, leading to a linear equation on the space of observables. The operator and generator are related by the matrix exponential, $\mathcal{K}_t = \exp(t\mathcal{L})$. Finding $\mathcal{L}$ is the objective of Koopman theory.

In summary, the potentially complex and non-linear ODE 1 on the finite-dimensional state space $\mathbb{R}^d$ can be expressed as a linear equation in another space, $\mathcal{G}(\mathbb{R}^d)$, which consists of scalar-valued functions defined on the state space. The practical challenge in Koopman theory is to find *invertible mappings* $f : \mathbb{R}^d \to \mathcal{G}(\mathbb{R}^d)$ that allow solving the linear equation in the observable space and then recovering the solution in the original state space. However, computing such a mapping is often intractable in practice due to the *infinite dimensionality* of $\mathcal{G}(\mathbb{R}^d)$.

A particular case arises when there exists an m-dimensional linear subspace of $\mathcal{G}(\mathbb{R}^d)$, $F = \text{span}\{g_i\}_{i=1}^m$, invariant under the linear operator $\mathcal{L}$. The action of the generator on F can then be represented by a single finite-dimensional matrix $L \in \mathbb{R}^{m\times m}$. The dynamics on this space of observables can then be written as:

$$\frac{d\mathbf{g}_t}{dt}(x) = L\mathbf{g}_t(x), \tag{4}$$

where $\mathbf{g}_t(x) = [g_1(x_t),\ \ldots,\ g_m(x_t)]^\intercal \in \mathbb{R}^m$ are the *Koopman coordinates*, i.e., the values of the observables $\{g_i\}_{i=1}^m$ evaluated at the state x_t, where x_t is the evolution of the initial state x to time t along the trajectories generated by the dynamics.

Thus, the general goal when applying Koopman theory to dynamical systems is to (1) identify a sufficiently expressive set of observables $\{g_i\}_{i=1}^m$ and (2) determine the Koopman generator matrix L on this space of observables. With this in hand, we can build an invertible Koopman representation $g : \mathbb{R}^d \to \mathbb{R}^m$ that maps a state x to its Koopman coordinates $\mathbf{g}(x)$. This enables us, given an initial state $x_0 \in \mathbb{R}^d$, to solve the ODE associated with a *nonlinear dynamical system* in a space where it evolves linearly, using the matrix exponential $\mathbf{g}_1 = e^L g(x_0) \in \mathbb{R}^m$. We can then recover the solution of the ODE in the original state space by applying the inverse map $x_1 = g^{-1}\big(e^L g(x_0)\big) \in \mathbb{R}^d$.

4 METHODOLOGY AND THEORETICAL RESULTS

Our objective is to learn a Koopman representation for a pre-trained CFM model, specified by its vector field v_t. This involves learning an encoder g_ϕ for the Koopman representation that linearizes the dynamics, a generator matrix L, and a decoder g_ψ^{-1} that maps back to the state space. Here ϕ and ψ^{-1} are the learnable parameters of the corresponding neural networks. Several additional challenges arise compared to previous neural Koopman-based approaches Lusch et al. (2018):

1. CFM dynamics are non-autonomous (explicitly time-dependent), whereas classic Koopman theory applies to autonomous systems.
2. The training objective for the Koopman representation must be tractable, ideally inheriting the simulation-free nature of CFM.
3. The learned observables g must be expressive enough to capture the dynamics and allow for accurately generated samples.

4.1 Adapting Koopman Theory to Non-Autonomous Dynamics

Time dependence trick. As mentioned above, Koopman theory applies to autonomous dynamics, where the velocity $v(x_t)$ does not depend on the time. We can address this time-dependence of $v_t(x_t)$ by using a standard trick in system dynamics literature (Strogatz (2000), Chap 1.): we augment the state space to include time. The state becomes $y_t = (t, x_t)$, and the dynamics are defined on this augmented space with respect to a new external time parameter τ:

$$\frac{dy}{d\tau} = \frac{d(t, x_t)}{d\tau} = [1,\ v_t(x_t)]\,. \tag{5}$$

Our observables are now functions of both space and time, $g(t, x)$. A crucial detail, however, is how we parameterize the linear dynamics on this augmented state to ensure the time variable evolves correctly (i.e., $\dot{t} = 1$).

Affine lift for time evolution. To enforce the constraint $\dot{t} \equiv 1$, we use an *affine lift*. The state is augmented with a constant bias coordinate to become $\mathbf{z}_t = [1,\ t,\ g(t,\ x)]^T$. For the dynamics $\dot{\mathbf{z}} = L\mathbf{z}$ to satisfy the physical constraints $\dot{1} = 0$ and $\dot{t} = 1$ for all states, the generator L is uniquely constrained to adopt a block structure. The precise parameterization of L is available in the appendix.

4.2 Learning Koopman Dynamics

Given a pre-trained CFM teacher network v_t, our main goal is to learn observable functions $\{g_i\}_{i=1}^m$ that span a finite-dimensional subspace *invariant under* the Koopman generator $\mathcal{L}$ *associated with* the dynamics v_t, and to learn the corresponding generator on this space. We learn the observables with an encoder g_ϕ that maps an initial state $x \in \mathbb{R}^d$ to its Koopman coordinates at time t, $\mathbf{g}_t(t,\ x) = [g_1(t,\ x_t),\ \ldots,\ g_m(t,\ x_t)]^\top \in \mathbb{R}^m$. We also learn the Koopman generator on this space as a dense matrix $L \in \mathbb{R}^{m\times m}$. To recover the solution of the ODE in the original state space and ensure the learned linear dynamics correspond to the *underlying nonlinear dynamics*, we also learn a decoder network g_ψ^{-1} that maps the Koopman coordinates $\mathbf{g}_t(x)$ back to the state x_t at time t.

We generate noise and target-data pairs $(x_0,\ x_1)$ using the pretrained CFM model, and aim to learn the following mapping:

$$x_t \simeq g^{-1}(e^{tL} g(0,\ x_0)).$$

Training loss Our training objective is as follows:

$$\mathcal{L}_{\text{train}} = \lambda_{\text{phase}}\mathcal{L}_{\text{phase}} + \lambda_{\text{target}}\mathcal{L}_{\text{target}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}}.$$

The first two terms ensure that the integrated linear dynamics map the start of a trajectory to its end in the Koopman space (phase loss):

$$\mathcal{L}_{\text{phase}} = \mathbb{E}_{(x_0,\ x_1)} \left\| e^L g_\phi(0,\ x_0) - g_\phi(1,\ x_1) \right\|^2. \tag{6}$$

and in the state space (after decoding - target loss):

$$\mathcal{L}_{\text{target}} = \mathbb{E}_{(x_0,\ x_1)} \left\| g_\psi^{-1}\left(e^L g_\phi(0,\ x_0)\right) - x_1 \right\|^2, \tag{7}$$

The third term encourages that we can retrieve the final state with the decoder:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{x_1} \left\| g_\psi^{-1}\left(g_\phi(1,\ x_1)\right) - x_1 \right\|^2. \tag{8}$$

The reconstruction loss is particularly important due to an inherent non-identifiability in the Koopman representation, as formalized in the proposition below. This term allows us to find, among the space of Koopman linearizing coordinate systems, the decodable ones.

We choose to only decode at $t = 1$ for those reasons: first, learning to reconstruct random noise may affect the capacity of the decoder to reconstruct images faithfully. Second, by not reconstructing intermediary states from observables, we give more flexibility to the encoder and generator to learn the proper Koopman representation space that manages to linearize the dynamics.

Proposition 1 (Non-identifiability up to linear transformation)**.** *The Koopman observable coordinates g are identifiable only up to an arbitrary invertible linear transformation M. If the pair (g, L) satisfies the consistency and phase objectives, so does the transformed pair $(M^{-1}g,\ M^{-1}LM)$.*

Corollary 1.1. *A reconstruction loss of the form $\|g^{-1}(g(t,\ x)) - x\|^2$, with a fixed decoder g^{-1}, breaks this invariance. It "fixes the gauge" by selecting the specific coordinate system that the chosen decoder can successfully map back to the data space.*

The proof is provided in Appendix A. This result motivates the necessity of $\mathcal{L}_{\text{recon}}$ to obtain a unique and useful representation.

Finally, the consistency loss forces the dynamics in the learned latent space to be governed by the linear generator L, by adapting Equation (3) to our problem:

$$\mathcal{L}_{\text{cons}} = \mathbb{E}_{t,\ x_t \sim p_t(x_t)} \left\| L g_\phi(t,\ x_t) - \nabla_{\boldsymbol{x}} g_\phi(x_t) v_t(x_t) \right\|^2 \tag{9}$$

4.3 Efficient Dynamics learning

One might notice that, similarly to the CNF loss, the consistency loss $\mathcal{L}_{\text{cons}}$ is intractable, as it would require sampling from the path distribution $x_t \sim p_t(x_t)$. A first solution would be to generate full trajectories $(x_t)_t$, but this would pose both discretization and scale problems for storing the pre-computed trajectories. Another solution is to hope to substitute the marginal velocity $v_t(x_t)$ with the conditional velocity $u_t(x_t|x_1)$ and sample from the tractable $p_t(x_t|x_1)$, mirroring the CFM training strategy. However, as the following proposition shows, these two objectives are not equivalent when learning the encoder g.

Proposition 2 (Marginal vs. Conditional Objectives)**.** *Let $\mathcal{L}_{marg}$ be the desired consistency loss evaluated over the marginal distribution $p_t(x_t)$, and let $\mathcal{L}_{cond}$ be the tractable alternative evaluated using conditional samples and velocities. The two objectives are related by:*

$$\mathcal{L}_{cond} = \mathcal{L}_{marg} + \Delta(g) \tag{10}$$

where $\Delta(g) = \mathbb{E}_{t,\ x_1,\ x_t} \left\| \nabla_x g(t,\ x_t)(u_t(x_t|x_1) - v_t(x_t)) \right\|^2 \geq 0.$

The proof is provided in Appendix B. Because of the positive, g-dependent term $\Delta(g)$, minimizing $\mathcal{L}_{\text{cond}}$ will not necessarily minimize $\mathcal{L}_{\text{marg}}$.

Fortunately, as we have a pre-trained CFM model, the marginal velocity field $v_t(x_t)$ is known. This allows us to formulate a practical estimator for the true marginal loss, as stated in the following proposition.

Proposition 3 (Practical Estimator for the Consistency Loss)**.** *Given that the marginal path distribution $p_t(x_t)$ is defined as $p_t(x_t) = \int p_t(x_t|x_1)q(x_1)dx_1$, the marginal consistency loss $\mathcal{L}_{cons}$ can be estimated tractably using samples from the data distribution $q(x_1)$ and the conditional path $p_t(\cdot|x_1)$ as follows:*

$$\mathcal{L}_{cons} = \mathbb{E}_{t,\ x_1 \sim q_1,\ x_t \sim p_t(\cdot|x_1)} \left\| L g_\phi(t,\ x_t) - \frac{\partial g_\phi}{\partial t} - \nabla_x g_\phi(t,\ x_t) \cdot v_t(t,\ x_t) \right\|^2 \tag{11}$$

The proof is provided in Appendix C. This result is key: it allows us to **optimize the correct marginal objective using the same efficient, simulation-free sampling strategy** as CFM training, bypassing the need to compute and store full ODE trajectories.

Moreover, this loss is a key distinction of our method. Most single-step distillation-based generative models Song et al. (2023) focus on learning a direct mapping $\mathcal{D} : x_0 \mapsto x_1$ that minimizes a boundary-condition loss, like $\|\mathcal{D}(x_0) - x_1\|^2$. However, by focusing on endpoints, the distillation completely ignores the dynamics of the generative ODE. An infinite number of vector fields can satisfy the boundary conditions. In contrast, our approach seeks to perform a true *linearization* of the **full dynamics**. The inclusion of the infinitesimal consistency loss, $\mathcal{L}_{\text{cons}}$, forced our Koopman representation to not only match the endpoints but to remain faithful to the teacher's dynamics **at every point along the trajectory**.

4.4 Decomposition of the Koopman observables and generator

Once the generator L, and the observables $\{g_i\}_{i=1}^m$ are known, it enables one-step sampling generation using the matrix exponentiation (Appendix, Algorithm 2). Moreover, it is possible to interpret the behavior of Koopman dynamics. We investigate two methods of dynamics interpretation.

Diracs of the Koopman space. A first option is to take the canonical basis of the Koopman coordinate frame. Given a single image x and its Koopman observations $z = [g_1(1,\ x), ..., g_m(1,\ x)]^T$, we define the canonical vector $e_i = [0,\ \dots, 1,\ \dots, 0]^T \in \mathbb{R}^m$. We can perturb the latent code z along the direction $e_i : z_{pert} = z_1 + \alpha e^L v_i$, as an analogy to Dirac delta perturbations.

Koopman spectral analysis. A second option is to use the spectral decomposition of the Koopman operator. If L is diagonalizable, its eigendecomposition $L = P \Lambda P^{-1}$ can serve to analyze the underlying dynamics. The eigenvectors v_i (columns of P) and the eigenvalues λ_i of L form the *Koopman mode decomposition* Mezić (2005). Any trajectory in the Koopman space is a linear combination of these modes, and the dynamics of the i-th mode coefficient $c_i(t)$ are decoupled, as they become a simple scalar multiplication; $c_i(t) = c_i(0)e^{\lambda_i t}$.

In our case, each eigenvector represents a coherent pattern or feature within the generative process, and its corresponding eigenvalue governs its behavior:

The **real part** of an eigenvalue, $\text{Re}(\lambda_i)$, determines the growth or decay rate of the mode. Modes with $\text{Re}(\lambda_i) > 0$ correspond to features that are amplified as the generation progresses from noise to data.

The **imaginary part**, $\text{Im}(\lambda_i)$, determines the mode's frequency of oscillation, corresponding to rotational or periodic patterns in the dynamics as features are formed.

We show in the experiments, Section 5.4 and in the Appendix C.3, C.4, that when consistency is enforced, both canonical observables and Koopman modes provide insightful directions on the underlying image distribution, in the case of faces. Moreover, we demonstrate in Section 5.5 that we can also un-lift the Koopman modes to the original CFM dynamics.

5 Experiments

To validate our framework, we investigate three key questions: (1) Can our one-step sampler achieve competitive generative quality? (2) Is the infinitesimal consistency loss ($\mathcal{L}_{cons}$) crucial for learning an interpretable linearization, as opposed to a simple boundary-matching distillation? (3) Does this learned structure lead to a more robust and functionally useful model? Our experiments show that while a simple distillation model can achieve a competitive FID Heusel et al. (2017) score, only the model trained with $\mathcal{L}_{cons}$ learns a disentangled, editable, and robust generative process.

5.1 Experimental Setup

Datasets and Teacher Model. We evaluate on MNIST LeCun et al. (2010), CIFAR-10 Krizhevsky et al. (2009), and a 32x32 downsampled version of the FFHQ face dataset Karras et al. (2019). Our teacher is a pre-trained Optimal Transport Conditional Flow Matching (OT-CFM) model with a U-Net architecture. For boundary-based losses ($\mathcal{L}_{\text{target}}$, $\mathcal{L}_{\text{phase}}$, $\mathcal{L}_{\text{recon}}$), we use 1 million pre-generated (x_0, x_1) pairs from the teacher network.

Koopman-CFM Architecture. Our model consists of an encoder (g_ϕ) and decoder (g_ψ^{-1}), both using a `SongUNet` architecture Karras et al. (2022), which map to and from a 1024-dimensional latent space. The dynamics are governed by a learned affine linear generator ($\tilde{L}$).

Training and Baselines. We train for 800,000 iterations using the Adam optimizer Kingma & Ba (2017). Our primary baseline is an ablation of our own model trained without the consistency loss ($\mathcal{L}_{cons} = 0$), which reduces it to a standard distillation model.

5.2 Generation Quality

Table 1: FID (↓) and sampling time (s/img, ↓) on three benchmark datasets. Our Koopman formulation achieves competitive or superior generation quality while enabling fast inference. Baselines are trained under identical preprocessing for fair comparisons. ♯ Indicates reproduction.

Method	NFE	MNIST	FFHQ	CIFAR-10	Sampling Time (ms/img)
Koopman (ours, w/ consistency)	1	7.1	10.1	17.4	0.4
Koopman (ours, w/o consistency)	1	6.4	7.5	14.1	0.4
OT-CFM	1	181	149	226	0.6
OT-CFM	3	28.1	51	59.3	1.4
OT-CFM	5	12.5	31.4	31.5	2.3
OT-CFM	25	4.4	11.6	12.3	10.7
OT-CFM (Tong et al. (2024))	100	1.9	8.5	7	43.3
Consistency Flow Matching (Yang et al. (2024))	2	♯ 7.2	♯ 15.7	5.3	0.96

We evaluate sample quality using the Fréchet Inception Distance (FID), shown in Table 1. Our full Koopman-CFM model with consistency achieves competitive performance. Interestingly, the model trained without consistency achieves a slightly superior FID on FFHQ (7.5 vs. 8.5 for the teacher). This suggests that, when only constraining the endpoints, the distillation model is free to find a combination of paths and latent space that is easier to learn. As mentioned above, however, such a model is not guaranteed to replicate the trajectories of the teacher model. We provide uncurated generated examples with the consistency trained model in the appendix Section D.

5.3 Ablation

Koopman space dimension. As shown in Figure 2, the Koopman dimension of 1026 (1024+2) is optimal for the generation quality. Notably, increasing the dimension to 1026 does not affect the quality with potential instabilities of the Koopman sampling components, such as the exponentiation.

Impact of consistency on trajectories. We measure how $\mathcal{L}_{cons}$ affects the capacity of the model to reproduce the teacher's dynamics. To test this, we encode a teacher's trajectory $\{x_t\}_{t\in[0,1]}$ in the latent space and compare this ground truth path $z_t = g_\phi(t, x_t)$ against the analytical linear trajectory from our model, $\tilde{z}_t = \exp(\tilde{L}t)\tilde{z}_0$. We show the results in Table 2, with more details in the appendix. The trajectories are significantly better when using the consistency loss.

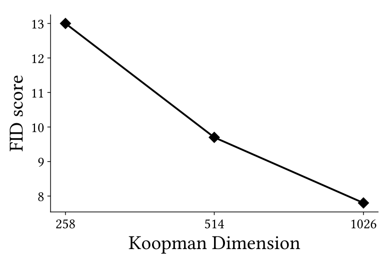


Figure 2: FID score as a function of Koopman dimension on the FFHQ dataset. The higher the dimension, the lower the FID.

Dataset	Mean MSE
FFHQ (w/ consistency)	$\mathbf{5.0 \times 10^{-6}}$
FFHQ (w/o consistency)	1.30×10^{-3}
CIFAR-10 (w/ consistency)	$\mathbf{1.0 \times 10^{-5}}$
CIFAR-10 (w/o consistency)	1.74×10^{-3}

Table 2: Mean, standard deviation of MSE between CFM trajectories and predicted Koopman trajectories. The consistency-trained model consistently outperforms the distilled model for trajectory fidelity

5.4 Interpretability via operator decomposition

We visualize on Figure 3 the effect of the perturbation along a canonical direction and an eigenvector direction. Across all basis directions and samples, we observed that the perturbation around a sample in the latent space of a model trained without consistency loss did not yield any modification of the sample. The model trained with consistency loss yields meaningful perturbations in the image space, for instance, adding sunglasses.

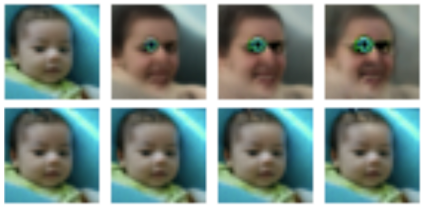


Figure 3.a Perturbation sweep along a latent canonical basis vector e_i.

Figure 3.b Perturbation sweep along a single Koopman mode for a real image.

Figure 3: Effect of varying the amplitude of the perturbation of the latent of a real image in a given direction. results for the model trained with consistency loss (top rows), versus without (bottom row).

5.5 RECOVERING KOOPMAN MODES IN PIXEL-SPACE DYNAMICS

In this section, we un-lift the Koopman modes to the CFM dynamics. We do this by solving an inverse problem: Let $x_0 \sim p_0$ be a sample noise, $\mathbf{v}_i$ a Koopman mode. We search x^i_{pert}, such that:

$$x^i_{\text{pert}} = \arg\min_x ||g_\phi(0, x_0 + x) - g_\phi(0, x_0) + \alpha \mathbf{v}_i||^2$$

We then generate samples along the line $x'_0 = x_0 + \alpha x^i_{\text{pert}}$, using the CFM teacher.

We show the results for a few modes in Figure 4. We observe a similar behaviour as when perturbing Koopman modes (until α becomes too high, and x^i_{pert} becomes too dominant). This demonstrates that Koopman modes also provide direct interpretability of the underlying CFM teacher dynamics. We pursue this discussion further in the appendix.

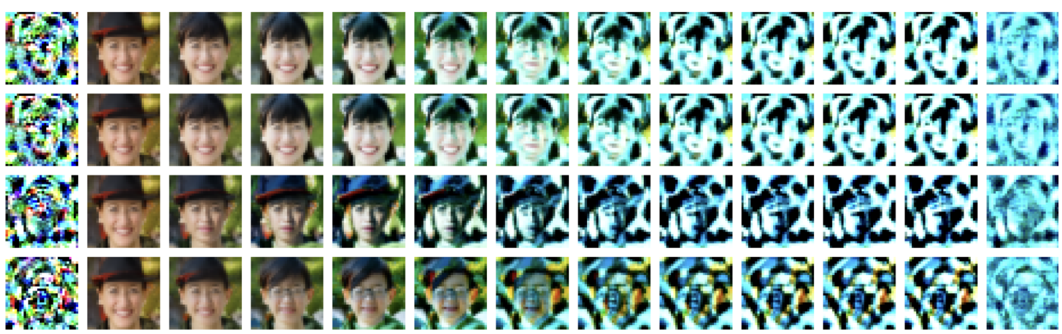


Figure 4: Recovering Koopman Modes in Pixel-Space Dynamics. (Left Column) The optimized, structured noise perturbation (x^i_{pert}) for 4 Koopman modes v_i. (Center Columns) Images generated by the *CFM model* from initial noise $x'_0 = x_0 + \alpha x^i_{\text{pert}} v_i$ with increasing α. (Right Column) The image generated by directly decoding the target Koopman mode v_i.

6 CONCLUSION AND DISCUSSION

We introduced a principled Koopman operator framework to linearize Conditional Flow Matching, achieving fast, one-step, and interpretable generative modeling on realistic image domains. Key challenges remain in scaling to high-resolution images, where the generator matrix becomes prohibitively large and its exponential can be numerically unstable. Future work should explore structured operator approximations and specialized matrix exponential algorithms to address these computational hurdles. Furthermore, we observe that the quality gap between our method and traditional CFM widens on more complex datasets, motivating a deeper theoretical investigation into the conditions under which CFM dynamics admit a finite-dimensional Koopman representation Iacob et al. (2023). Finally, the modality-agnostic nature of our framework opens exciting avenues for adapting this linearization approach to other data types, such as audio and 3D shapes.

REFERENCES

Omri Azencot, N Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning*, pp. 475–485. PMLR, 2020.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Nimrod Berman, Ilan Naiman, Moshe Eliasof, Hedi Zisling, and Omri Azencot. One-step offline distillation of diffusion-based models via koopman modeling. *arXiv preprint arXiv:2505.13358*, 2025.

Petar Bevanda, Stefan Sosnowski, and Sandra Hirche. Koopman operator dynamical models: Learning, analysis and control. *Annual Reviews in Control*, 52:197–212, 2021.

Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern koopman theory for dynamical systems. *SIAM Review*, 64(2):229–340, 2022. doi: 10.1137/21M1401243.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2180–2188, 2016.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *Workshop on Deep Learning, NIPS*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations (ICLR)*, 2017.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

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, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Proc. NeurIPS*, 2020.

Lucian C Iacob, Maarten Schoukens, and Roland Tóth. Finite dimensional koopman form of polynomial nonlinear systems. *IFAC-PapersOnLine*, 56(2):6423–6428, 2023.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the ”steerability” of generative adversarial networks. In *International Conference on Learning Representations*, 2020.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

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

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Stefan Klus, Feliks Nüske, Péter Koltai, Hao Wu, Ioannis Kevrekidis, Christof Schütte, and Frank Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28:985–1010, 2018.

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. *arXiv preprint arXiv:2412.08629*, 2024.

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *The Eleventh International Conference on Learning Representations*, 2022.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Yaron Lipman, Ioannis Gkioulekas, Tatsunori Hashimoto, William T Liu, Ben Poole, Ricky Richter-Powell, Robin Rombach, Ali Toker, and Jiaxin Wu. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Enshu Liu, Xuefei Ning, Yu Wang, and Zinan Lin. Distilled decoding 1: One-step sampling of image auto-regressive models with flow matching. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate straight lines improves generalization and efficiency. In *International Conference on Learning Representations (ICLR)*, 2023a. Often referred to as Rectified Flow.

Yujia Liu, Chuan Guo Li, Kuan-Chieh Zhou, and Anima Anandkumar. Flow matching with stochastic differential equations. *arXiv preprint arXiv:2306.02393*, 2023b.

Simian Luo, Yiqin Wu, Surui Wang, Puchao Chen, Shijie Zhao, Jun Zhu, et al. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

Aram-Alexandre Pooladian, Alexander Gushchin, Regina Barzilay, and Tommi Jaakkola. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2305.17160*, 2023.

Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1532–1540, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Prafulla Dhariwal, Mark Zhang, and Karsten Kreis. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2000.

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Liu, Kilian Rector-Brooks, Guy Wolf Fatras, Elizabeth Creager, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, pp. 9786–9796. PMLR, 2020.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency, 2024. URL `https://arxiv.org/abs/2407.02398`.

Tao Yang, Yuwang Wang, Yan Lv, and Nanning Zheng. Disdiff: Unsupervised disentanglement of diffusion probabilistic models. *arXiv preprint arXiv:2301.13721*, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

APPENDIX

The supplementary materials below provide an expanded theoretical motivation, experimental details, and additional results that support and extend the main paper. Each section corresponds to specific elements of our method and results, with backward references to the main paper for clarity.

APPENDIX OVERVIEW

- **Section A: Theoretical Results and Proofs**
  In Section A we provide additional theoretical results and proofs, including the non-identifiability of Koopman coordinates, the non-equivalence of the conditional and marginal velocity field estimators and a tractable estimator for the marginal consistency loss.
- **Section B: Detailed Experimental Setup**
  In Section B we give details on the experimental setup, covering dataset preparation, architectures, hyperparameters, and computational resources.
- **Section C: Ablations**
  In Section C we present ablations, shedding light on the impact of loss terms on FID, the effect of Koopman dimension on FID, the role of consistency in trajectory fidelity, and the interpretability of modes with and without consistency.
- **Section D: Uncurated Samples and Sampling Speeds**
  In Section D we provide uncurated samples and wall-clock timings to further illustrate the speed–fidelity–interpretability tradeoff of our Koopman sampler.
- **Section E: Recovering Koopman Modes in Pixel-Space Dynamics: extended results**
- **Section F: Extended survey on interpretability of generative models**

Together, these sections provide a deeper understanding of our Koopman-CFM framework and support its efficiency, stability, and interpretability as claimed in the main paper.

A THEORETICAL RESULTS AND PROOFS

In this section we expand on the theoretical foundations introduced in Section 4 of the main paper. We provide detailed proofs of Theorem 1 and Propositions 1–3, which establish the non-identifiability of Koopman coordinates up to linear transformations and justify the inclusion of the reconstruction loss, as well as the derivation of a tractable marginal consistency objective. These results complement the main text by giving formal guarantees for the claims underlying our Koopman-CFM framework.

A.1 PRELIMINARIES ON CFM

We remind here the main components of Conditional Flow Matching Tong et al. (2023), before deriving the proofs of our propositions. A Continuous Normalizing Flow (CNF) models the transformation from a prior distribution p_0 to a data distribution $p_1 = q_1$ via a probability path p_t. This path is induced by a time-dependent vector field v_t through the ODE:

$$\frac{dx_t}{dt} = v_t(x_t) \tag{12}$$

where $x_t \in \mathbb{R}^d$ is a sample at time t. A naive objective to learn v_t would be a regression loss:

$$\mathcal{L}_{\text{naive}} = \mathbb{E}_{t\sim U(0,1),x_t\sim p_t} \left\|v_\theta(t, x_t) - v_t(x_t)\right\|^2 \tag{13}$$

This objective is intractable because both the true vector field v_t and the marginal path distribution p_t are unknown. Conditional Flow Matching (CFM) circumvents this by defining a tractable conditional probability path $p_t(x_t|x_1)$ and its corresponding conditional vector field $u_t(x_t|x_1)$. The marginal velocity field v_t can be expressed as an expectation over these conditional fields:

$$v_t(x_t) = \mathbb{E}_{x_1\sim q(x_1|x_t)}[u_t(x_t|x_1)] = \int \frac{p_t(x_t|x_1)q(x_1)}{p_t(x_t)} u_t(x_t|x_1)dx_1 \tag{14}$$

Remarkably, CFM shows that minimizing a simulation-free objective based on the conditional velocity field is equivalent to minimizing the intractable marginal objective. The CFM loss is:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t\sim U(0,1),x_1\sim q_1,x_t\sim p_t(\cdot|x_1)} \left\|v_\theta(t, x_t) - u_t(x_t|x_1)\right\|^2 \tag{15}$$

While this makes training efficient, sampling requires solving the integral:

$$x_1 = x_0 + \int_0^1 v_\theta(s, x_s)ds \tag{16}$$

A.2 Proof of Theorem 1

Proof. Let the augmented state observable be $E(t, x) = [t, g(t, x)]^T$. We show that the objectives are invariant under the transformation $E \mapsto E_T = T^{-1}E$ and $L \mapsto L_T = T^{-1}LT$ for any invertible block-diagonal matrix $T = \text{diag}(1, M)$.

We use two facts. First, the chain rule implies that the Jacobian transforms as:

$$\mathrm{D}(E_T)[1, v_t] = \mathrm{D}(T^{-1}E)[1, v_t] = T^{-1}\mathrm{D}E[1, v_t]. \tag{J}$$

Second, the matrix exponential (and thus the flow) is conjugate under T:

$$\exp(\Delta t L_T) = T^{-1} \exp(\Delta t L)T. \tag{C}$$

Infinitesimal Consistency. The residual is $R_{\text{cons}} = \mathrm{D}E[1, v_t] - LE$. The transformed residual is:

$$R_{\text{cons},T} = \mathrm{D}E_T[1, v_t] - L_T E_T \overset{\text{(J),(C)}}{=} T^{-1}\mathrm{D}E[1, v_t] - T^{-1}LE = T^{-1}R_{\text{cons}}.$$

Thus, $R_{\text{cons}} = 0$ if and only if $R_{\text{cons},T} = 0$.

Phase Loss. The residual is $R_{\text{phase}} = E(1, x_1) - e^L E(0, x_0)$. The transformed residual is:

$$\begin{aligned} R_{\text{phase},T} &= E_T(1, x_1) - e^{L_T} E_T(0, x_0) \\ &= T^{-1}E(1, x_1) - (T^{-1}e^L T)(T^{-1}E(0, x_0)) \\ &= T^{-1}(E(1, x_1) - e^L E(0, x_0)) = T^{-1}R_{\text{phase}}. \end{aligned}$$

Again, the zero set of the loss is invariant. Since the norms of the residuals are scaled by the constant transformation T^{-1}, the set of global minimizers is preserved under this transformation. Therefore, the objectives only identify g up to an invertible linear transformation M. □

A.3 Proof of Proposition 1

Proof. To simplify the notation, let us define:

$$\begin{aligned} A(x_t) &= Lg(x_t) \\ B(x_t) &= \nabla g(x_t)\, v_t(x_t) \\ C(x_t, x_1) &= \nabla g(x_t)\, u_t(x_t \mid x_1) \end{aligned}$$

With this notation, the losses are $\mathcal{L}_{\text{marg}} = \mathbb{E}_{x_t\sim p_t}[\|A(x_t) - B(x_t)\|^2]$ and $\mathcal{L}_{\text{cond}} = \mathbb{E}_{x_1\sim q,x_t\sim p_t(\cdot|x_1)}[\|A(x_t) - C(x_t, x_1)\|^2]$.

We expand the squared norms inside the expectations:

$$\mathcal{L}_{\text{marg}} = \int p_t(x_t) \left(\|A\|^2 - 2\langle A, B\rangle + \|B\|^2\right) dx_t$$

$$\mathcal{L}_{\text{cond}} = \iint q(x_1)\, p_t(x_t \mid x_1) \left(\|A\|^2 - 2\langle A, C\rangle + \|C\|^2\right) dx_t\, dx_1$$

We will now compare the terms of these two expansions one by one.

(i) First Term ($\|A\|^2$): The first term of $\mathcal{L}_{\text{cond}}$ is $\iint q(x_1)\, p_t(x_t \mid x_1)\, \|A(x_t)\|^2\, dx_t\, dx_1$. Since $A(x_t)$ does not depend on x_1, we can use the law of iterated expectation or simply rearrange the integral:

$$\iint q(x_1)\, p_t(x_t \mid x_1)\, \|A(x_t)\|^2 \, dx_t \, dx_1 = \int \left(\int q(x_1)\, p_t(x_t \mid x_1)\, dx_1 \right) \|A(x_t)\|^2 \, dx_t$$
$$= \int p_t(x_t)\, \|A(x_t)\|^2 \, dx_t$$

This is identical to the first term of $\mathcal{L}_{\text{marg}}$.

(ii) Cross Term ($-2\langle A, \cdot \rangle$): The cross term of $\mathcal{L}_{\text{cond}}$ is $\iint q(x_1)\, p_t(x_t \mid x_1) \big(-2\langle A(x_t), C(x_t, x_1)\rangle \big)\, dx_t \, dx_1$. We analyze the integral:

$$\begin{aligned}
&\iint q(x_1)\, p_t(x_t \mid x_1)\, \langle A(x_t), C(x_t, x_1)\rangle \, dx_t \, dx_1 \\
&= \int \left\langle A(x_t), \int q(x_1)\, p_t(x_t \mid x_1)\, C(x_t, x_1)\, dx_1 \right\rangle dx_t \\
&= \int \left\langle A(x_t), \int q(x_1)\, p_t(x_t \mid x_1)\, \nabla g(x_t)\, u_t(x_t \mid x_1)\, dx_1 \right\rangle dx_t \\
&= \int \left\langle A(x_t), \nabla g(x_t) \int q(x_1)\, p_t(x_t \mid x_1)\, u_t(x_t \mid x_1)\, dx_1 \right\rangle dx_t
\end{aligned}$$

By definition, the marginal velocity field $v_t(x_t)$ is the expectation of the conditional field $u_t(x_t \mid x_1)$ over the posterior $p(x_1 \mid x_t) = \frac{q(x_1) p_t(x_t|x_1)}{p_t(x_t)}$. So, $v_t(x_t) = \int u_t(x_t \mid x_1) \frac{q(x_1) p_t(x_t|x_1)}{p_t(x_t)}\, dx_1$. Multiplying by $p_t(x_t)$ gives $p_t(x_t) v_t(x_t) = \int q(x_1)\, p_t(x_t \mid x_1)\, u_t(x_t \mid x_1)\, dx_1$. Substituting this back into our expression:

$$\begin{aligned}
\ldots &= \int \langle A(x_t), \nabla g(x_t)\, (p_t(x_t) v_t(x_t))\rangle \, dx_t \\
&= \int \langle A(x_t), p_t(x_t) B(x_t)\rangle \, dx_t \\
&= \int p_t(x_t) \langle A(x_t), B(x_t)\rangle \, dx_t
\end{aligned}$$

This shows that the cross terms of $\mathcal{L}_{\text{cond}}$ and $\mathcal{L}_{\text{marg}}$ are also identical.

(iii) Final Quadratic Term ($\|\cdot\|^2$): The final term of $\mathcal{L}_{\text{cond}}$ is $\mathbb{E}_{x_1, x_t}[\|C(x_t, x_1)\|^2]$. We use the law of total variance: for a random variable Z, $\mathbb{E}[\|Z\|^2] = \|\mathbb{E}[Z]\|^2 + \text{Var}(Z)$. We apply this by first conditioning on x_t.

$$\begin{aligned}
\mathbb{E}_{x_1, x_t}[\|C\|^2] &= \mathbb{E}_{x_t \sim p_t}\left[\mathbb{E}_{x_1 \sim p(x_1|x_t)}[\|C(x_t, x_1)\|^2]\right] \\
&= \mathbb{E}_{x_t}\left[\|\mathbb{E}_{x_1|x_t}[C(x_t, x_1)]\|^2 + \text{Var}_{x_1|x_t}(C(x_t, x_1))\right]
\end{aligned}$$

Let's compute the inner conditional expectation:
$\mathbb{E}_{x_1|x_t}[C(x_t, x_1)] = \mathbb{E}_{x_1|x_t}[\nabla g(x_t) u_t(x_t \mid x_1)] = \nabla g(x_t) \mathbb{E}_{x_1|x_t}[u_t(x_t \mid x_1)] = \nabla g(x_t) v_t(x_t) = B(x_t)$.
Substituting this back:

$$\begin{aligned}
\mathbb{E}_{x_1, x_t}[\|C\|^2] &= \mathbb{E}_{x_t}\left[\|B(x_t)\|^2 + \text{Var}_{x_1|x_t}(C(x_t, x_1))\right] \\
&= \mathbb{E}_{x_t}[\|B(x_t)\|^2] + \mathbb{E}_{x_t}[\text{Var}_{x_1|x_t}(C(x_t, x_1))]
\end{aligned}$$

The first part, $\mathbb{E}_{x_t}[\|B(x_t)\|^2] = \int p_t(x_t) \|B(x_t)\|^2 \, dx_t$, is exactly the final term of $\mathcal{L}_{\text{marg}}$. The second part is the discrepancy term $\Delta(g)$:

$$\begin{aligned}
\Delta(g) &= \mathbb{E}_{x_t}[\text{Var}_{x_1|x_t}(C(x_t, x_1))] \\
&= \mathbb{E}_{x_t}\left[\mathbb{E}_{x_1|x_t}\left[\|C(x_t, x_1) - \mathbb{E}_{x_1|x_t}[C(x_t, x_1)]\|^2\right]\right] \\
&= \mathbb{E}_{x_t}\left[\mathbb{E}_{x_1|x_t}\left[\|C(x_t, x_1) - B(x_t)\|^2\right]\right] \\
&= \mathbb{E}_{x_1, x_t}\left[\|C(x_t, x_1) - B(x_t)\|^2\right] \\
&= \iint q(x_1)\, p_t(x_t \mid x_1) \|\nabla g(x_t) u_t(x_t \mid x_1) - \nabla g(x_t) v_t(x_t)\|^2 \, dx_t \, dx_1 \\
&= \iint q(x_1)\, p_t(x_t \mid x_1) \|\nabla g(x_t)(u_t(x_t \mid x_1) - v_t(x_t))\|^2 \, dx_t \, dx_1
\end{aligned}$$

Conclusion: Assembling all the terms, we have:

$$\begin{aligned}
\mathcal{L}_{\text{cond}} &= \underbrace{\mathbb{E}_{x_t}[\|A\|^2]}_{\text{Term 1}} - \underbrace{2\mathbb{E}_{x_t}[\langle A, B\rangle]}_{\text{Term 2}} + \underbrace{\left(\mathbb{E}_{x_t}[\|B\|^2] + \Delta(g)\right)}_{\text{Term 3}} \\
&= \left(\mathbb{E}_{x_t}[\|A\|^2] - 2\mathbb{E}_{x_t}[\langle A, B\rangle] + \mathbb{E}_{x_t}[\|B\|^2]\right) + \Delta(g) \\
&= \mathcal{L}_{\text{marg}} + \Delta(g)
\end{aligned}$$

Since $\Delta(g)$ is the expectation of a squared norm, it is non-negative, which proves the theorem. □

A.4 Proof of Proposition 2

Proof. The proof relies on the law of iterated expectation. Let $f(x_t)$ be any measurable function of x_t. The expectation of $f(x_t)$ over the marginal distribution $p_t(x_t)$ is:

$$\mathbb{E}_{x_t\sim p_t}[f(x_t)] = \int_{\mathbb{R}^d} f(x_t)p_t(x_t)\,dx_t$$

Now, we substitute the definition of the marginal path density, $p_t(x_t) = \int_{\mathbb{R}^d} q(x_1)p_t(x_t|x_1)\,dx_1$:

$$\mathbb{E}_{x_t\sim p_t}[f(x_t)] = \int_{\mathbb{R}^d} f(x_t)\left(\int_{\mathbb{R}^d} q(x_1)p_t(x_t|x_1)\,dx_1\right)dx_t$$

We can combine the terms inside a double integral:

$$\mathbb{E}_{x_t\sim p_t}[f(x_t)] = \iint_{\mathbb{R}^d\times\mathbb{R}^d} f(x_t)q(x_1)p_t(x_t|x_1)\,dx_1\,dx_t$$

By Fubini's theorem, we can exchange the order of integration since the integrand is non-negative (or integrable):

$$\mathbb{E}_{x_t\sim p_t}[f(x_t)] = \int_{\mathbb{R}^d} q(x_1)\left(\int_{\mathbb{R}^d} f(x_t)p_t(x_t|x_1)\,dx_t\right)dx_1$$

This expression can be recognized as a nested expectation. The inner integral is the expectation of $f(x_t)$ over the conditional distribution $p_t(\cdot|x_1)$, and the outer integral is the expectation over the data distribution $q(x_1)$:

$$\begin{aligned}
\int_{\mathbb{R}^d} q(x_1)\left(\mathbb{E}_{x_t\sim p_t(\cdot|x_1)}[f(x_t)]\right)dx_1 &= \mathbb{E}_{x_1\sim q}\left[\mathbb{E}_{x_t\sim p_t(\cdot|x_1)}[f(x_t)]\right] \\
&= \mathbb{E}_{x_1\sim q, x_t\sim p_t(\cdot|x_1)}[f(x_t)]
\end{aligned}$$

We have thus shown the general identity $\mathbb{E}_{x_t\sim p_t}[f(x_t)] = \mathbb{E}_{x_1\sim q, x_t\sim p_t(\cdot|x_1)}[f(x_t)]$.

To prove the theorem, we simply choose $f(x_t)$ to be the squared residual of the marginal loss:

$$f(x_t) = \left\|\mathcal{L}g(x_t) - \nabla_x g(x_t)\,v_t(x_t)\right\|^2$$

By its definition, $\mathcal{L}_{marg} = \mathbb{E}_{x_t\sim p_t}[f(x_t)]$. Applying the identity we just derived gives:

$$\mathcal{L}_{marg} = \mathbb{E}_{x_1\sim q, x_t\sim p_t(\cdot|x_1)}\left[\left\|\mathcal{L}g(x_t) - \nabla_x g(x_t)\,v_t(x_t)\right\|^2\right]$$

This completes the proof. □

B Experimental Details

This section complements Section 5 of the main paper by providing full details needed for reproducibility. We describe dataset prepration, model architecture and parametrization, training schedules, and computational resources.

B.1 Parameterization of the affine lift

We parameterize $\tilde{L}$ with the following block structure.

$$\tilde{L} \;=\; \begin{bmatrix} 0 & 0 & \mathbf{0} \\ 1 & 0 & \mathbf{0} \\ \mathbf{b}_g & \mathbf{A}_{gt} & \mathbf{A}_{gg} \end{bmatrix} \tag{17}$$

This parameterization guarantees correct time evolution by design and yields affine dynamics for the observables: $\dot{g} = \mathbf{b}_g + \mathbf{A}_{gt}t + \mathbf{A}_{gg}g$. The learned parameters are the weights ϕ, ψ of the endocer g_phi and decoder g_ψ^{-1} and the matrix blocks $(\mathbf{b}_g, \mathbf{A}_{gt}, \mathbf{A}_{gg})$.

Algorithm 1: Koopman–CFM Training (simulation-free; fixed teacher, precomputed pairs)

Input: Fixed teacher velocity $v_{\text{CFM}}(t,x)$; encoder g_ϕ; decoder g_ψ^{-1}; affine generator $\tilde{L}$; precomputed buffer $\mathcal{B} = \{(x_0, x_1)\}$.
Definition: Lifted coordinate $\tilde{z}(t,x) := [\,1,\ t,\ g_\phi(t,x)\,]^\top$.
for *each minibatch* **do**
  Sample $x_1 \sim q_1$, $t \sim \mathcal{U}(0,1)$, then draw $x_t \sim p_t(\cdot \mid x_1)$;
  $\mathcal{L}_{\text{cons}} \leftarrow \left\| \tilde{L}\,\tilde{z}(t,x_t) \ - \ Dg_\phi(t,x_t)[\,1,\ v_{\text{CFM}}(t,x_t)\,] \right\|^2$;
  Sample (x_0, x_1) from buffer $\mathcal{B}$;
  $\mathcal{L}_{\text{phase}} \leftarrow \left\| \exp(\tilde{L})\,\tilde{z}(0,x_0) \ - \ \tilde{z}(1,x_1) \right\|^2$;
  $\mathcal{L}_{\text{target}} \leftarrow \ell_{\text{img}}\Big(g_\psi^{-1}\big(\exp(\tilde{L})\,\tilde{z}(0,x_0) \big),\ x_1 \Big)$;
  $\mathcal{L}_{\text{recon}} \leftarrow \ell_{\text{img}}\Big(g_\psi^{-1}\big(\tilde{z}(1,x_1) \big),\ x_1 \Big)$;
  $\mathcal{L} \leftarrow \lambda_c \mathcal{L}_{\text{cons}} + \lambda_p \mathcal{L}_{\text{phase}} + \lambda_t \mathcal{L}_{\text{target}} + \lambda_r \mathcal{L}_{\text{recon}}$;
  Update $\{\phi, \psi, \tilde{L}\}$ by backprop on $\mathcal{L}$;

Algorithm 2: One-Step Koopman Sampling (matrix exponential + decode)

Input: Trained (g_ϕ, g_ψ^{-1}, L); prior $p_0 = \mathcal{N}(0, I)$.
Input: Lifted coordinate $z(t,x) := [\,1,\ t,\ g_\phi(t,\ x)\,]^\top$.
Precompute $E \leftarrow \exp(L)$;
Sample $x_0 \sim p_0$;
return $\hat{x}_1 \leftarrow g_\psi^{-1}\Big(E\, z(0,\ x_0) \Big)$;

Data. We evaluate our approach on three datasets of increasing difficulty. MNIST contains 60,000 training and 10,000 test grayscale images of handwritten digits at resolution 28×28. FFHQ (Flickr-Faces-HQ) was downscaled to 32×32 resolution, from which we use all 70,000 RGB images. Finally, CIFAR-10 provides 50,000 training and 10,000 test images at resolution 32×32 across 10 object classes. This progression from simple digits to natural faces and general object classes allows us to systematically study the performance of our method as task complexity increases.

Model Architecture. For all datasets, we employ a consistent backbone architecture: a SongUNet used as both encoder and decoder. To reduce the overall parameter count, we restrict the encoder output and decoder input to a single channel. Moreover, to obtain explicit control over the Koopman dimension, we optionally append a linear projection from the flattened UNet output to the target latent dimension.

Training Details. Before training our pipeline, we pre-trained an OT-CFM model following the reference implementation provided in the torchcfm code examples. From this model, we generated between 10^4 and 10^6 (x_0, x_1) pairs (see Table 3 for exact counts per dataset), which served as inputs for computing the target loss. All models were trained using the Adam optimizer under identical training protocols across datasets. Experiments were carried out on NVIDIA A40, H100, and A100 GPUs. Additional hyperparameters, including learning rates, batch sizes, and training schedules, are reported in Table 3.

C Ablations

This section expands the analysis of Section 5 by presenting ablations that clarify the role of each loss term, the effect of Koopman dimension, the impact of consistency on trajectory fidelity, and the interpretability of modes.

	MNIST	FFHQ	CIFAR-10
CFM iterations	200k	800k	800k
Batch size	128	256	124
Learning rate	0.0001	0.0001	0.0001
Koopman iterations	70k	600k	800k
Target weight (w/o $\mathcal{L}_{\text{cons}}$ – w/ $\mathcal{L}_{\text{cons}}$)	1.0 – 1.0	1.0 – 0.01	1.0 – 0.01
Operator Dimension	1026	1026	1026
UNet Output Channels	1	1	1
UNet Base Channels	64	64	64
UNet Channels Multiplier	[1,2,2]	[1,2,2,2]	[1,2,2,2]
Linear Projection	✓	✗	✗

Table 3: Training hyperparameters for Koopman–CFM on MNIST, FFHQ, and CIFAR-10. *Linear projection* refers to the projection head at the UNet encoder output (resp. decoder input). Since loss terms are not of the same order of magnitude, the target loss was reweighted by the given parameter. *Koopman iterations* denote the number of iterations for the overall pipeline, while *CFM iterations* correspond to the underlying CFM model.

C.1 Impact of Loss Terms

Table 4 shows the effect of adding loss components across datasets. Phase and reconstruction alone yield poor FIDs, as they impose no constraint in image space. Adding the target loss improves fidelity by supervising decoded samples. Adding the consistency loss (weight 0.01) slightly worsens FID (e.g., FFHQ 7.5 $\rightarrow$ 10.1), since it regularizes the model to follow the teacher's nonlinear trajectories rather than shortcutting through straighter ones. This increases trajectory faithfulness at the cost of marginally higher endpoint error. We argue this tradeoff is beneficial: while endpoint-only distillation can optimize FID, it fails to capture the true generative flow (cf. Table 5, Fig. 5). Consistency-trained models achieve competitive FIDs while uniquely enabling spectral decomposition and robust downstream performance.

Table 4: Loss ablation across datasets showing the effect of incrementally adding loss components

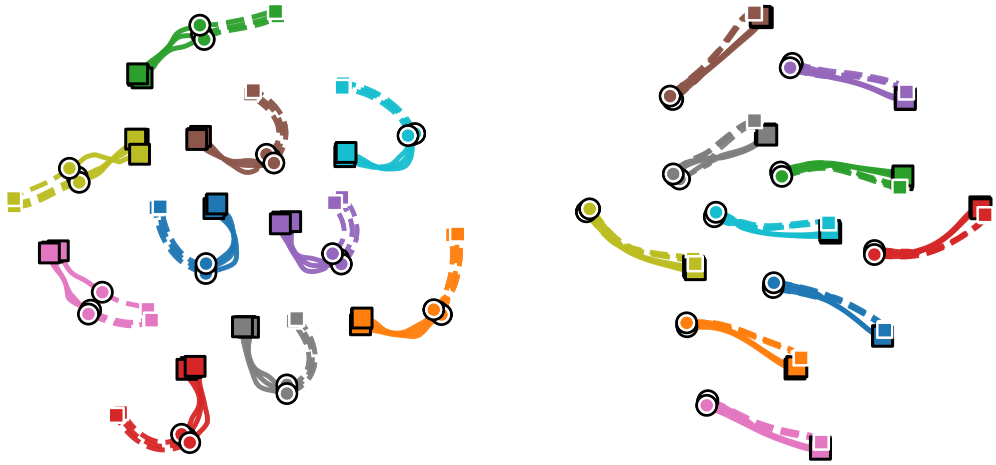
Dataset	$\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{phase}}$	$\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{phase}} + \mathcal{L}_{\text{target}}$	$\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{phase}} + 0.01\mathcal{L}_{\text{target}} + \mathcal{L}_{\text{cons}}$
MNIST	143.5	6.43	11.6
FFHQ	41	7.5	10.1
CIFAR-10	64.5	16.7	14.1

C.2 Trajectory Fidelity with and without Consistency Loss

Table 5: MSE between trajectory rollouts between CFM and Koopman dynamics in latent space: we generate 1000 full trajectories $\{x_t\}$ via CFM encode in the Koopman latent space $g(t, x_t)$ and compare them with Koopman rollouts $g(x_t) = e^{Lt} g(t = 0, x_0)$.

Dataset	Min	Max	Mean MSE	Std Dev
FFHQ (w/ consistency)	3.0×10^{-6}	1.3×10^{-5}	5.0×10^{-6}	1.0×10^{-6}
FFHQ (w/o consistency)	5.24×10^{-4}	2.66×10^{-3}	1.30×10^{-3}	2.95×10^{-4}
CIFAR-10 (w/ consistency)	4.0×10^{-6}	3.7×10^{-5}	1.0×10^{-5}	4.0×10^{-6}
CIFAR-10 (w/o consistency)	3.46×10^{-4}	7.01×10^{-3}	1.74×10^{-3}	6.36×10^{-4}

We illustrate in Figure 5, the impact of consistency on trajectory fidelity. Notably, the consistency trained models trajectories closely tracks the teacher's nonlinear path. In contrast, the purely distilled model trajecotry diverges significantly, learning an unaligned shortcut, but with correct boundaries.



Without consistency loss

With consistency loss

Figure 5: t-SNE visualization of CFM and Koopman trajectories in the embedding space on FFHQ. The consistency loss makes Koopman rollouts (dotted) follow the teacher dynamics (continuous) more closely. This is seen both in the proximity of trajectories and in the alignment of their endpoints. Circles mark starting points and squares mark end points.

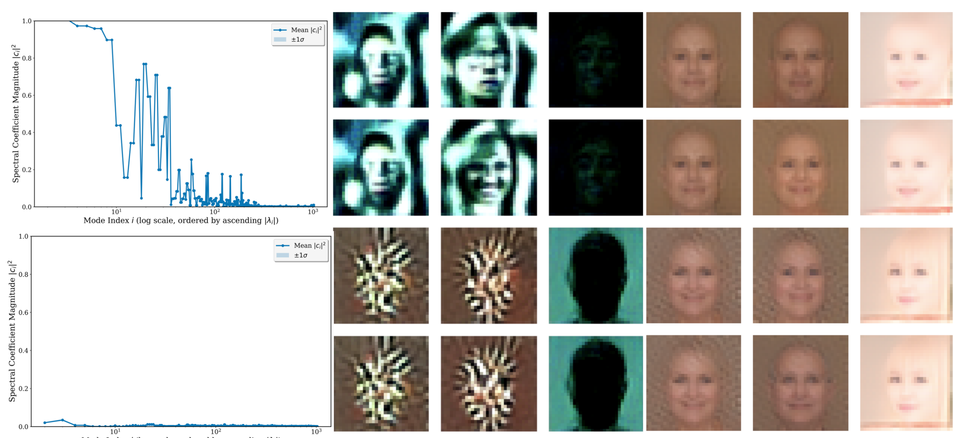


Figure 6: Left: First three columns are some decoded modes of the generator trained without consistency loss, and the next three are those obtained from the generator trained with consistency. Right: Mean coefficients $|c_i|^2$ projected on the generator modes ordered by corresponding eigenvalue magnitude $|\lambda_i|^2$. Top corresponds to the spectrum along the modes obtained from training with consistency and bottom to those obtained from training without consistency

This confirms that while a better FID can be achieved by ignoring the teacher's dynamics, doing so prevents the model from learning a faithful representation of the generative process.

C.3 Modes with and without Consistency

Figure 6 illustrates how consistency qualitatively changes the learned Koopman modes. Without consistency, individual modes tend to decode into entire faces—effectively full puzzle pieces—which suggests poor disentanglement, as each mode redundantly encodes the whole sample. By contrast, with consistency, the modes behave like localized "patch bases," decomposing faces into locally meaningful components (e.g., hair, eyes), reminiscent of Fourier modes assembling a signal from parts. This interpretation is reinforced by the spectral profile on the right of Fig. 6: with consistency, coefficients decay smoothly with eigenvalue magnitude, closely resembling a Fourier spectrum, whereas without consistency the spectrum remains flat, indicating the absence of structured decomposition. Together with the perturbation experiments in Fig. 6 and the downstream robustness results in Fig. Figure 7.a, Figure 7.b, Figure 7.c, these findings confirm that consistency is crucial for learning a Koopman operator that provides an interpretable, basis-like decomposition of generative dynamics.

C.4 Functional Robustness on Downstream Tasks

Finally, we evaluate if this interpretable structure translates to functional robustness on challenging downstream tasks: inpainting, super-resolution, and denoising. These tasks test the model's ability to perform conditional generation, which depends on the quality of its learned dynamics. For a corrupted input encoded to $z_{1,corr}$, we reconstruct by adding noise at $t = 0$ and evolving it through the learned process:

$$z_{0,corr} = e^{-L} z_{1,corr} \quad ; \quad x_{recon} = g_{\psi}^{-1}(e^{L}(z_{0,corr} + \text{noise}))$$

As shown in Figure 7, the consistency-trained model significantly outperforms the ablation model across all tasks. This superior performance is a direct consequence of the structured, Fourier-like basis described above. Because its learned dynamics can induce local, patch-based semantic modifications, the model is uniquely equipped to solve tasks that require local reasoning, like inpainting a missing patch. The purely distilled model fails and simply reproduces the same image, showing that it learned the noise to data map, but not much about the underlying image data distribution.

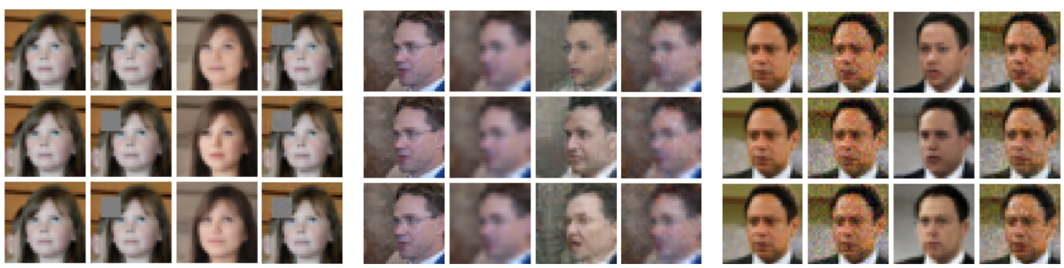


Figure 7.a Inpainting   Figure 7.b Super-Resolution   Figure 7.c Denoising

Figure 7: Performance on structured generative tasks. For each task, we show the input, the corrupted image, the result from our consistency-trained model, and the result from the ablation model. Our model consistently produces coherent, high-fidelity results, while the ablation model fails.

D Uncurated Samples

This section supplements Section 5 by showing uncurated generations and reporting wall-clock sampling times, illustrating the tradeoffs between, speed, fidelity and interpretability.

E Recovering Koopman Modes in Pixel-Space Dynamics: extended results

We provide extended results from the experiment of 5.5.

Results and Discussion: As shown in Figure 9, the results seems to provide a validation of our approach. The model trained **with** the consistency loss ($\mathcal{L}_{\text{cons}}$) unlocks a degree of compositional control over the CFM teacher. We observe the ability to apply specific attributes—such as adding or removing hats and glasses—while preserving the subject's identity. This demonstrates a high degree of **compositional capacity**.

Furthermore, this experiment gives rise to the concept of a learned **"attribute catalogue."** It seems like the visual appearance of the decoded modes (Column 12) can hint at their functional effect, indeed we see that hats and glasses seems to be associated with the mode that appears to be wearing a hat and glasses. This may allow to browse a dictionary of learned Koopman modes and using them

MNIST (uncurated) FFHQ (uncurated) CIFAR-10 (uncurated)

Figure 8: Uncurated samples from our Koopman generative model across three datasets. All samples are obtained via our one-step strategy.

Figure 9: The consistency loss is critical for semantic control. Each row shows an optimized noise perturbation steering the original CFM model. **Top:** Our model, trained with $\mathcal{L}_{\text{cons}}$, achieves compositional edits like adding a adding or removing hats and glasses while preserving identity. **Bottom:** An identical model trained without $\mathcal{L}_{\text{cons}}$ fails, producing only noise. This ablation proves that our consistency loss is essential for learning a dynamically-aligned and interpretable latent space

to steer CFM sampling towards desired edits. This optimization procedure yields non-trivial noise perturbation x_{pert}.

Crucially, the ablation study shown in the bottom block of Figure 9 is definitive. The model trained **without** $\mathcal{L}_{\text{cons}}$ completely fails to produce any coherent semantic changes. This stark contrast proves that the observed compositional control is not an accidental property but a direct consequence of enforcing infinitesimal consistency along the entire generative trajectory. It is the $\mathcal{L}_{\text{cons}}$ term that forces the Koopman latent space to become a dynamically faithful and interpretable map of the teacher's own generative manifold.

F Extended survey on interpretability of generative models

There is a rich body of work on understanding how generative models transform noise into data. Early research on VAEs and GANs focused on analyzing their latent spaces. Variational Autoencoders were used to learn *disentangled* representations of data Bengio et al. (2013), i.e., latent codes that separate the underlying generative factors of variation Higgins et al. (2016); Burgess et al. (2018); Kim & Mnih (2018); Khemakhem et al. (2020). The success of Generative Adversarial Networks Goodfellow et al.

(2014) prompted similar studies Chen et al. (2016). Because the latent space of GANs is not explicitly structured, research focused on identifying directions that correspond to interpretable generative factors, enabling controlled image editing Jahanian et al. (2020); Härkönen et al. (2020); Voynov & Babenko (2020); Shen & Zhou (2021). The rise of diffusion and flow models as state-of-the-art generative methods naturally raised the question of whether such interpretability techniques could be extended to these models. However, their iterative generation process and the prevalence of complex, learnable control mechanisms Zhang et al. (2023) have not yielded equally simple or powerful methods for interpretation and editing. Existing approaches tend to be more complicated and lack the conceptual clarity and usability of those developed for VAEs and GANs Kwon et al. (2022); Yang et al. (2023); Meng et al. (2022); Kulikov et al. (2024). In contrast, our method preserves the dynamical-systems view of these models while enabling simple and interpretable latent-space manipulations.