Functional Mean Flow in Hilbert Space

Zhiqi Li Yuchen Sun Greg Turk Bo Zhu
Georgia Institute of Technology
zli3167@gatech.edu, yuchen.sun.eecs@gmail.com, turk@cc.gatech.edu, bo.zhu@gatech.edu

Abstract

We present Functional Mean Flow (FMF) as a one-step generative model defined in infinite-dimensional Hilbert space. FMF extends the one-step Mean Flow framework [15] to functional domains by providing a theoretical formulation for Functional Flow Matching and a practical implementation for efficient training and sampling. We also introduce an x_1-prediction variant that improves stability over the original u-prediction form. The resulting framework is a practical one-step Flow Matching method applicable to a wide range of functional data generation tasks such as time series, images, PDEs, and 3D geometry.

1. Introduction

Functional generative models (e.g., [4, 9, 27]) represent data in the form of continuous functions [9], where the underlying generative process is modeled as a probability distribution defined over function spaces [10]. Compared with standard generative models defined in discrete space, the main advantage of a functional model lies in its ability to subsample coordinates while maintaining a continuous functional representation, effectively decoupling memory and runtime cost from data resolution. This property enables training and sampling at arbitrary spatial or temporal resolutions. For example, *Infty-Diff* [4] employs non-local integral operators to map between Hilbert space, achieving up to an $4\times$ subsampling rate without compromising quality.

As many other generative models, such as Diffusion [20, 50, 51] or Flow Matching [34, 35, 48], the performance of functional generative models is also limited by the need for many sampling steps during inference. To address this bottleneck, recent work explores one-step or few-step methods that directly approximate the endpoint transport. Among them, *Mean Flow* [16] provides a principled approach by predicting the time-averaged velocity instead of the instantaneous velocity used in standard Flow Matching. This design captures the overall transport in a single update, enabling efficient one-step sampling and achieving 60%–90% better FID than previous one-step models.

Extending one-step generation to Functional Flow Matching is fundamentally challenging in two aspects: (1) the infinite-dimensional Hilbert-space setting makes modeling highly non-trivial, as finite-dimensional intuitions no longer apply and the modeling confronts the inconsistency between marginal and conditional flows, making it infeasible to generalize the finite-dimensional mean-velocity formulation to infinite-dimensional functional spaces; (2)

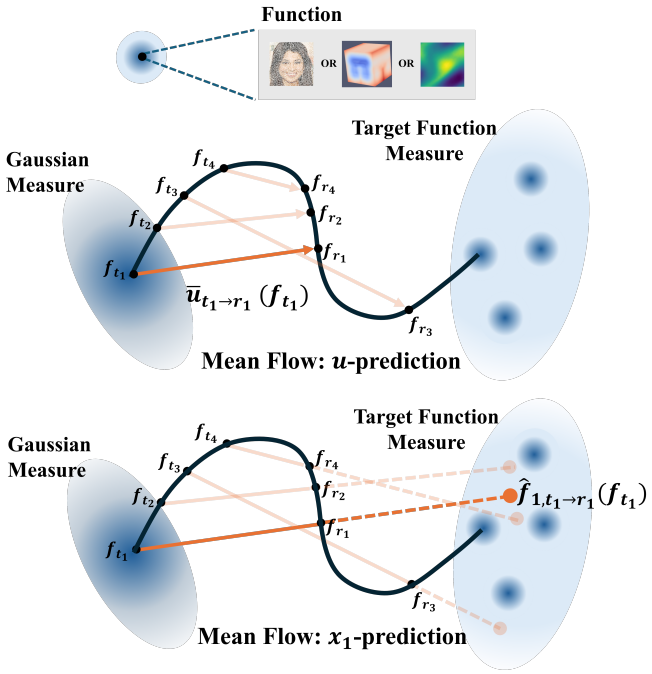


Figure 1. Illustration of Functional Mean Flow. The figure shows a 2D projection of the infinite-dimensional function space. During generation, the flow transports a Gaussian measure to the target function measure. The u-prediction FMF models the mean velocity $\bar{u}_{t\to r}(f_t)$ between any two points f_t and f_r along the flow trajectory, while the x_1-prediction FMF estimates the expected position $\hat{f}_{1,t\to r}(f_t)$ reached by continuing the mean velocity $\bar{u}_{t\to r}(f_t)$ for the remaining distance $1-t$. Both u- and x_1-prediction FMFs support one-step generation, formulated respectively as $f_1 = f_0 + \bar{u}_{0\to 1}(f_0)$ and $f_1 = \hat{f}_{1,0\to 1}(f_0)$.

functional derivatives and operator-valued velocity fields introduce numerical instability, complicating optimization and adversely affecting convergence across different functional generation tasks.

To address these challenges, we derive a new transport formulation based on the Fréchet derivative of two-parameter flows, which establishes the Mean Flow formulation in infinite-dimensional spaces and resolves the theoretical inconsistency between conditional and marginal dynamics. In addition, we reformulate the learning objective as an equivalent conditional loss with a stop-gradient approximation and introduce an x_1-prediction variant that predicts the expected endpoint by extrapolating the mean velocity, instead of predicting the mean velocity itself. These developments together constitute our proposed framework, **Functional Mean Flow (FMF)**, which enables stable and efficient one-step functional generation across a wide range of tasks in infinite-dimensional spaces.

We summarize our contributions as follows:

1. We derived the infinite-dimensional mean-velocity formulation, establishing a mathematically sound framework for one-step generation in Hilbert space.
2. We introduce, for the first time, the x_1-prediction variant of Mean Flow and show that it exhibits improved stability over the original u-prediction formulation on certain tasks, thereby broadening the applicability of the Mean Flow framework.
3. We demonstrate the effectiveness of the proposed method across a range of functional tasks, including time series modeling, image generation, PDEs, and 3D shape generation.

2. Functional Flow Matching

Functional Flow Matching (FFM) [27] extends classical Flow Matching from finite-dimensional Euclidean spaces to infinite-dimensional function spaces. Let $(\mathcal{F}, \langle\cdot,\cdot\rangle_{\mathcal{F}})$ be a separable Hilbert space of functions with the Borel σ-algebra $\mathcal{B}(\mathcal{F})$, and let $\mu_0 = \mathcal{N}(m_0, C_0)$ be a reference Gaussian measure on $\mathcal{F}$ with mean $m_0 \in \mathcal{F}$ and covariance operator $C_0 : \mathcal{F} \to \mathcal{F}$, FFM learns a time-dependent velocity field $u : [0,1] \times \mathcal{F} \to \mathcal{F}$ that transports μ_0 to a target distribution $\mu_1 = \nu$ through a continuous path of measures $(\mu_t)_{t\in[0,1]}$ satisfying the weak continuity equation

$$\begin{aligned}&\int_0^1 \int_{\mathcal{F}} \big(\partial_t \psi(g,t) + \langle u_t(g), \nabla_g \psi(g,t)\rangle_{\mathcal{F}}\big) \mathrm{d}\mu_t(g)\mathrm{d}t = 0,\\ &\mu_t|_{t=0} = \mu_0, \mu_t|_{t=1} = \mu_1,\end{aligned} \tag{1}$$

for all appropriate test functions $\psi : \mathcal{F} \times [0,1] \to \mathbb{R}$.

Sampling $f_0 \sim \mu_0$, one obtains a generated function by integrating

$$\frac{\mathrm{d}f_t}{\mathrm{d}t} = u(t, f_t), \quad f_t|_{t=0} = f_0, \tag{2}$$

whose terminal state satisfies $f_1 \sim \nu$. For velocity field u_t, the associated flow $\phi_t : \mathcal{F} \to \mathcal{F}$ are defined as maps satisfying $f_t = \phi_t(f_0)$ for all f_0 and f_t in Equation 2. The flow ϕ_t satisfies the functional differential equation

$$\frac{\partial}{\partial_t}\phi_t = u_t \circ \phi_t, \qquad \phi_0 = \mathrm{Id}_{\mathcal{F}}, \tag{3}$$

where $\mathrm{Id}_{\mathcal{F}}$ denotes the identity operator on $\mathcal{F}$. The path of measures $(\mu_t)_{t\in[0,1]}$ can be generated by the pushforward of the flow $\mu_t = (\phi_t)_\sharp \mu_0$, thereby extending the continuous transport formulation to infinite-dimensional Hilbert spaces.

To make the training $\mathcal{L}(\theta) = \mathbb{E}_{t,g\sim\mu_t}\big[\|u_t(g) - u_t^\theta(g)\|_{\mathcal{F}}^2\big]$ tractable, where the reference marginal velocity field u_t cannot be computed analytically, FFM introduces conditional velocity u_t^f conditioned on the target function $f \sim \nu$ and corresponding conditional paths of measures $(\mu_t^f)_{t\in[0,1]}$ that interpolate between μ_0 and a f-centered measure μ_1^f. Marginalizing these conditionals yields the global measure path and velocity

$$\begin{aligned}\mu_t(A) &= \int_{\mathcal{F}} \mu_t^f(A)\mathrm{d}\nu(f),\\ u_t(g) &= \int_{\mathcal{F}} u_t^f(g)\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)\mathrm{d}\nu(f),\end{aligned} \tag{4}$$

for arbitrary $A \in \mathcal{B}(\mathcal{F})$ where $\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}$ is the Radon–Nikodym derivative. In practice, the conditional paths μ_t^f are typically chosen to be Gaussian measure $\mu_t^f = \mathcal{N}(m_t^f, (\sigma_t^f)^2 C_0)$ with $m_t^f = tf$, $\sigma_t^f = 1 - (1-\sigma_{\min})t$ and a small positive number $\sigma_{\min}$. The conditional velocity and its associated flow admits a closed form

$$\begin{aligned}\phi_t^f(f_0) &= \sigma_t^f f_0 + m_t^f = (1-(1-\sigma_{\min})t)f_0 + tf,\\ u_t^f(g) &= \frac{\dot\sigma_t^f}{\sigma_t^f}(g - m_t^f) + \dot m_t^f = \frac{1-\sigma_{\min}}{1-(1-\sigma_{\min})t}(tf - g) + f.\end{aligned} \tag{5}$$

Although the theory requires $\sigma_{\min} > 0$, in practice setting $\sigma_{\min} = 0$ causes no adverse effects [4].

The model is then trained via the conditional loss

$$\mathcal{L}_c(\theta) = \mathbb{E}_{t,f\sim\nu,g\sim\mu_t^f}\left[\|u_t^f(g) - u_\theta(t,g)\|_{\mathcal{F}}^2\right], \tag{6}$$

which can be proved equivalent to the marginal loss $\mathcal{L}(\theta)$ up to a constant. For completeness, the corresponding theorems from [27] on Functional Flow Matching are provided in the Appendix A.

3. Functional Mean Flow

Similar to conventional Flow Matching, Functional Flow Matching also suffers from the drawback that inference requires many integration steps. To address this limitation, we

Figure 2. Representing data as functions enables the same model to synthesize images at arbitrary resolutions with different noise levels. The model is trained only on randomly sampled 1/4 subsets of pixels from 256×256 CelebA-HQ images and performs one-step generation. Left to right: 64×64, 128×128, 256×256, 512×512, and 1024×1024.

extend Mean Flow [16] to the infinite-dimensional function space for one-step generation. In addition, We further propose an $\mathbf{x_1}$-prediction variant of Mean Flow, which predicts the intersection between the extrapolated mean velocity line and the terminal point at $t = 1$, different from the original $\mathbf{u}$-prediction formulation of Mean Flow. This x_1-prediction variant exhibits improved stability on certain task as shown in section 4.

3.1. FMF with u-prediction

We first define a two-parameter flow as $\phi_{t\to r} = \phi_r \circ \phi_t^{-1}$, where the inverse map ϕ_t^{-1} is guaranteed to exist by the uniqueness of the ODE solution of Equation 2. Based on $\phi_{t\to r}$, the mean velocity $\bar{u}_{t\to r} : \mathcal{F} \to \mathcal{F}$ is defined as

$$\bar{u}_{t\to r} = \frac{1}{r-t}(\phi_{t\to r} - \mathrm{Id}_{\mathcal{F}}). \tag{7}$$

In Functional Mean Flow, our goal is to learn the target velocity $\bar{u}_{t\to r}$ through a loss $\mathcal{L}^M(\theta) = \mathbb{E}_{t,r,g\sim\mu_t}\left[\|\bar{u}_{t\to r}(g) - \bar{u}^\theta_{t\to r}(g)\|^2_{\mathcal{F}}\right]$. However, since the reference mean velocity $\bar{u}_{t\to r}$ has no closed-form expression, similar to Functional Flow Matching, we aim to reformulate the training objective in terms of a conditional field. The reformulation for Functional Flow Matching relies on the consistency between conditional and marginal velocity fields (Theorem A.1). In our formulation, however, this consistency breaks down, as $\bar{u}_{t\to r}$ and $\phi_{t\to r}$ do not admit a consistent corresponding conditional field representation.

Statement 1 (Mismatch Between Flow and Marginals of Conditional Flow)**.** *In general, the marginal flow* $\phi^{(1)}_{t\to r}(g) = \int_{\mathcal{F}} \phi^f_{t\to r}(g)\frac{\mathrm{d}\mu^f_t}{\mathrm{d}\mu_t}(g)\mathrm{d}\nu(f)$ *obtained by taking the expectation over the conditional two-parameter flows* $\phi^f_{t\to r} = \phi^f_r \circ (\phi^f_t)^{-1}$ *is not equivalent to the two-parameter flow* $\phi^{(2)}_{t\to r} = \phi_r \circ (\phi_t)^{-1}$*. Here, the superscripts* $^{(1)}$ *and* $^{(2)}$ *denote two different ways of computing the marginal two-parameter flow. (see Appendix B.1 for proof.)*

To address this issue, we first derive an equivalent reformulation of the mean velocity $\bar{u}_{t\to r}$, which relies on the following theorem:

Theorem 3.1 (Initial-Time Derivative of Two-Parameter Flow)**.** *Assume that the dataset measure* ν *satisfies* $\int_{\mathcal{F}} \|f\|^2_{\mathcal{F}}\mathrm{d}\nu(f) < \infty$*, and the conditions of Functional Flow Matching [27] hold. With the conditional flow and conditional velocity chosen in Equation 5, the corresponding two-parameter flow* $\phi_{t\to r}(g)$ *is differentiable with respect to* t *and Fréchet differentiable with respect to* g *and satisfies, for any* $0 < t < r < 1$

$$\frac{\partial}{\partial t}\phi_{t\to r}(g) = -D\phi_{t\to r}(g)[u_t(g)], \tag{8}$$

where $D\phi_{t\to r}(g) : \mathcal{F} \to \mathcal{F}$ *is the Fréchet derivative of* $\phi_{t\to r}$ *at* g*. This theorem follows from Lemmas B.1,B.2 and B.3 in Appendix B.2. (see Appendix B.3 for proof.)*

With Theorem 3.1 and the definition of $\bar{u}_{t\to r}$ in Equation 7, the mean velocity $\bar{u}_{t\to r}$ can be expressed as

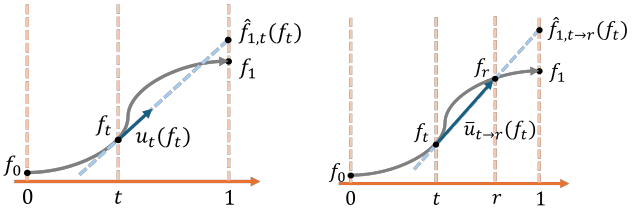
$$\begin{aligned}\bar{u}_{t\to r}(g) &\overset{①}{=} (r-t)\frac{\partial}{\partial t}\left[\frac{1}{r-t}(\phi_{t\to r} - \mathrm{Id}_{\mathcal{F}})(g)\right] - \frac{\partial}{\partial t}\phi_{t\to r}(g)\\ &\overset{②}{=} (r-t)\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\phi_{t\to r}(g)[u_t(g)]\\ &\overset{③}{=} (r-t)\left(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\bar{u}_{t\to r}(g)[u_t(g)]\right) + u_t(g),\end{aligned} \tag{9}$$

where ① follows from the product rule, ② is obtained by substituting Equation 8, and ③ is obtained by substituting Equation 7.

In the above expression of $\bar{u}_{t\to r}$, the right-hand side still depends on $\bar{u}_{t\to r}$ itself. Following [16, 52], we estimate this term using the current prediction of the model with a stop gradient operation and the velocity field u_t can be written as the marginal form of the conditional velocity u^f_t, and thus we define the conditional loss as

$$\begin{aligned}\mathcal{L}^M_c(\theta) = E_{t,r,g\sim\mu^f_t, f\sim\mu_1}\Big[\|(r-t)\mathrm{sg}\big(&\frac{\partial}{\partial t}\bar{u}_{t\to r}(g)\\ &+ D\bar{u}_{t\to r}(g)[u^f_t(g)]\big) + u^f_t(g) - \bar{u}^\theta_{t\to r}(g)\|^2_{\mathcal{F}}\Big],\end{aligned} \tag{10}$$

where sg means the stop gradient operation and $\bar{u}_{t\to r}(g)$ in sg is approximated by $\bar{u}^\theta_{t\to r}(g)$. The following theorem



(a) Functional Flow Matching. (b) Functional Mean flow.

Figure 3. From the x_1-prediction of Functional Flow Matching to the x_1-prediction of Functional Mean Flow. In the left figure, we illustrate the relationship between the u-prediction (predicting $u_t(f_t)$) and the x_1-prediction (predicting $\hat{f}_{1,t}(f_t)$) in flow matching, which satisfies $\hat{f}_{1,t}(f_t) = (1-t)u_t(f_t) + f_t$. Based on this relationship, we can analogously define the x_1-prediction of functional Mean Flow (predict $\hat{f}_{1,t\to r}(f_t)$), satisfying $\hat{f}_{1,t\to r}(f_t) = (1-t)u_{t\to r}(f_t) + f_t$.

establishes that the conditional loss $\mathcal{L}_c^M(\theta)$ is equivalent to the marginal loss $\mathcal{L}^M(\theta)$ up to a constant and can therefore be used to train the Functional Mean Flow model.

Theorem 3.2 (Equivalence of Mean Flow Conditional and Marginal Losses)**.** *Under the assumptions of Theorem 3.1, we have $\mathcal{L}_c^M(\theta) = \mathcal{L}^M(\theta) + C$ where C is independent of the model parameters θ. (see Appendix B.4 for proof.)*

3.2. FMF with x_1-prediction

In addition to the common u-prediction, standard Flow Matching also has an x_1-prediction variant, as shown in Figure 3(a). In Functional Flow Matching, the u-prediction estimates the velocity $u_t(f_t)$ at time t, while the x_1-prediction predicts the intersection $\hat{f}_{1,t}(f_t)$ between the extrapolated $u_t(f_t)$ and $t=1$, satisfying $\hat{f}_{1,t}(f_t) = (1-t)u_t(f_t) + f_t$. Similarly, in Functional Mean Flow (see Figure 3(b)), the u-prediction estimates the mean velocity between t and r, and we can define the x_1-prediction as the intersection of the extrapolated $\bar{u}_{t\to r}$ with $t=1$

$$\hat{f}_{1,t\to r} = (1-t)u_{t\to r} + \mathrm{Id}_{\mathcal{F}} = \frac{1-t}{r-t}\phi_{s\to t} - \frac{1-r}{r-t}\mathrm{Id}_{\mathcal{F}}. \tag{11}$$

For the x_1-prediction, the Functional Mean Flow loss is $\tilde{\mathcal{L}}^M(\theta) = \mathbb{E}_{t,r,g\sim\mu_t}\left[\|\hat{f}_{1,t\to r}(g) - \hat{f}^{\theta}_{1,t\to r}(g)\|^2_{\mathcal{F}}\right]$. As with the u-prediction, $\tilde{\mathcal{L}}^M(\theta)$ cannot be optimized directly, and we optimize its corresponding conditional loss instead

$$\tilde{\mathcal{L}}_c^M(\theta) = E_{t,r,g\sim\mu_t^f, f\sim\mu_1}\Big[\|\frac{r-t}{1-r}sg((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) + D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g) - g]) + \hat{f}^f_{1,t}(g) - \hat{f}^{\theta}_{1,t\to r}(g)\|^2_{\mathcal{F}}\Big], \tag{12}$$

where $\hat{f}^f_{1,t}(g)$ denotes the conditional value of $\hat{f}_{1,t}(g)$ with respect to f, analogous to how $u_t^f(g)$ serves as the conditional counterpart of $u_t(g)$. $\hat{f}^f_{1,t}(g)$ can be computed as follows (see Appendix B.5 for a detailed derivation):

$$\hat{f}^f_{1,t}(g) = \frac{\sigma_{\min}}{1-(1-\sigma_{\min})t}(g - tf) + f. \tag{13}$$

Similar to the equivalent reformulation of $\bar{u}_{t\to r}$ in Equation 9, the above x_1-prediction conditional loss is derived from the following equivalent reformulation of $\hat{f}_{1,t\to r}$ (see Appendix B.5 for derivation in details):

$$\hat{f}_{1,t\to r}(g) = \frac{r-t}{1-t}\big((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) + D\hat{f}_{1,t\to r}(g)[\hat{f}_{1,t}(g) - g]\big) + \hat{f}_{1,t}(g). \tag{14}$$

It can be shown that the x_1-prediction Functional Mean Flow also admits the following equivalent form:

Theorem 3.3 (Equivalence of Mean Flow Conditional and Marginal Losses for x_1-prediction)**.** *Under the assumptions of Theorem 3.1, we have $\tilde{\mathcal{L}}_c^M(\theta) = \tilde{\mathcal{L}}^M(\theta) + C$ where C is independent of the model parameters θ. (see subsection B.5 for proof.)*

In our experiments in section 4, we found that, in general, the u-prediction and x_1-prediction yield comparable results. However, in certain task, the u-prediction becomes highly unstable and fails to optimize, whereas the x_1-prediction demonstrates much better stability.

Remark. Although our x_1-prediction Mean Flow also predicts the endpoint, it differs from prior methods Consistency Models (CM) [52] and Flow Map Matching (FMM) [2]. CM and FMM predict the true future state f_r from the current function f_t, whereas our method, inspired by x_1-prediction Flow Matching, predicts the intersection of the velocity line with t=1. In addition, CM cannot fully utilize gradient information, and FMM optimizes quantities inside gradient operators, causing instability and high cost. Our x_1-prediction Mean Flow is theoretically equivalent to u-prediction Mean Flow and avoids these drawbacks.

3.3. Algorithm

Similar to Functional Flow Matching, Functional Mean Flow starts from functions sampled from a Gaussian measure, since white noise is undefined in infinite-dimensional spaces [4, 58]. The model also requires a function-to-function network, such as a Neural Operator (see section 4 for details on sampling from Gaussian measure and network). Similar to [16, 52], the gradient terms in Equation 10 and Equation 12 can be computed through the JVP operation within the optimization framework. Based on the above, we obtain the training and sampling algorithms for the u-prediction and x_1-prediction variants of Functional Mean Flow in Algorithm 1 and Algorithm 2. For clarity, we include Python code examples in the Appendix D.

Figure 4. Results on AFHQ, LSUN-Church, and FFHQ. The model is trained on a random 1/4 pixel subset of 256×256 images and evaluated at 256×256 and 512×512 via one-step generation.

Algorithm 1 Functional Mean Flow: Training

Input: dataset $\mathcal{D}$, initial model parameter θ, learning rate η, Gaussian measure sampler $\mathcal{N}(0, C_0)$, time sampler $\mathcal{T}$

repeat
  Sample $f \sim \mathcal{D}$, $f_0 \sim \mathcal{N}(0, C_0)$ and $t, r \sim \mathcal{T}$
  $g \leftarrow (1-(1-\sigma_{\min})t)f_0 + tf$
  if u-prediction **then**
    $u_t^f \leftarrow \frac{1-\sigma_{\min}}{1-(1-\sigma_{\min})t}(tf - g) + f$
    $\mathcal{L}(\theta) \leftarrow \|(r-t)\mathrm{sg}(\frac{\partial}{\partial t}\bar{u}^\theta_{t\to r}(g) + D\bar{u}^\theta_{t\to r}(g)[u_t^f]) + u_t^f - \bar{u}^\theta_{t\to r}(g)\|^2_{\mathcal{F}}$
  else if x_1-prediction **then**
    $\hat{f}^f_{1,t}(g) \leftarrow \frac{\sigma_{\min}}{1-(1-\sigma_{\min})t}(g - tf) + f$
    $\mathcal{L}(\theta) \leftarrow \|\frac{r-t}{1-r}\mathrm{sg}\big((1-t)\frac{\partial}{\partial t}\hat{f}^\theta_{1,t\to r}(g) + D\hat{f}^\theta_{1,t\to r}(g)[\hat{f}^f_{1,t}(g) - g]\big) + \hat{f}^f_{1,t}(g) - \hat{f}^\theta_{1,t\to r}(g)\|^2_{\mathcal{F}}$
  $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}(\theta)$
until convergence

Algorithm 2 Functional Mean Flow: Inference

Input: trained model parameter θ, Gaussian measure sampler $\mathcal{N}(0, C_0)$

Sample $f_0 \sim \mathcal{N}(0, C_0)$
if u-prediction **then**
  $f \leftarrow \bar{u}^\theta_{t\to r}(f_0) + f_0$
else if x_1-prediction **then**
  $f \leftarrow \hat{f}^\theta_{1,t\to r}(f_0)$

4. Experiment

To evaluate the generality and effectiveness of our approach, we conduct experiments on three distinct and representative tasks: real-world functional generation (including time-series data and Navier–Stokes solutions) [26, 27, 32, 45], function-based image generation [4, 29], and SDF-based 3D shape generation [58]. For all tasks, we adopt the neural architecture originally designed for multi-step generation, with only a minor modification that replaces the single time variable t with a pair (t, r) to meet the requirements of the FMF formulation (see Appendix C); the models are then trained with Algorithm 1. The experimental results demonstrate that our framework can be seamlessly integrated into various functional generation paradigms, enabling effortless adaptation of existing neural architectures for one-step generation. These include Neural Operators [26, 27, 32, 45], hybrid sparse–dense Neural Operators [4, 29], and point-based functional generation models [58].

4.1. Real-World Functional Generation

We now investigate the empirical performance of our FMF model on several real-world functional datasets. For fair comparison, we follow the same experimental setup as prior works [27] and adopt the Fourier Neural Operator (FNO) as the backbone to model $\bar{u}^\theta_{t\to r}(g)$ for u-prediction and $\hat{f}^\theta_{1,t\to r}(g)$ for x_1-prediction, which takes functions as both inputs and outputs. The network size and structural parameters are kept identical to previous implementations, and for initial Gaussian measure, a Gaussian processes with a Matérn kernel is used for parametrization (see Appendix C.1 for details).

Our functional datasets consist of two categories: (1) Five 1D statistical datasets with diverse correlation structures, including a daily temperature dataset (AEMET) [11], a gene expression time-series dataset (Genes) [40], an economic population time-series dataset (Pop.) [3], a GDP-per-capita dataset (GDP) [21], and a labor-force-size dataset (Labor) [22]; and (2) a 2D fluid dynamics dataset consist-

Table 1. Comparison of different functional generative method on 1D datasets. Statistical metrics (mean, variance, skewness, kurtosis, and autocorrelation) are reported across datasets. The best results for the 1-step and multi-step settings are highlighted in bold.

Dataset	Mean ↓	Variance ↓	Skewness ↓	Kurtosis ↓	Autocorrelation ↓	NFEs
AEMET						
FMF (u-pred)	**5.3e-1** (1.5e-1)	2.0e+0 (1.3e+0)	7.4e-2 (3.2e-2)	**1.4e-1** (5.7e-2)	**5.2e-4** (9.1e-6)	1
FMF (x_1-pred)	5.4e-1 (1.7e-1)	**1.8e+0** (9.8e-1)	**6.8e-2** (6.0e-2)	1.8e-1 (1.1e-1)	5.6e-4 (1.0e-5)	1
GANO	6.5e+1 (1.9e+2)	7.1e+1 (4.0e+1)	4.7e-1 (4.8e+0)	3.2e-1 (1.0e+0)	2.0e-3 (2.6e-3)	1
FFM-OT	**8.4e-2** (9.9e-2)	1.7e+0 (1.1e+0)	7.7e-2 (6.6e-2)	3.3e-2 (3.7e-2)	**3.0e-6** (4.0e-6)	668
FFM-VP	1.3e-1 (1.4e-1)	**1.5e+0** (1.2e+0)	**5.2e-2** (4.3e-2)	**1.7e-2** (1.6e-2)	6.0e-6 (7.0e-6)	488
FDDPM	2.6e-1 (3.0e-1)	3.5e+0 (1.0e+0)	1.1e-1 (4.2e-2)	3.9e-2 (3.0e-2)	5.0e-6 (5.0e-6)	1000
DDO	2.4e-1 (2.6e-1)	6.6e+0 (5.1e+0)	2.1e-1 (4.1e-2)	3.8e-2 (3.1e-2)	6.7e-6 (1.3e-4)	2000
Genes						
FMF (u-pred)	**1.6e-3** (8.3e-4)	**3.3e-4** (1.5e-4)	**3.6e-2** (9.6e-3)	**9.5e-2** (2.3e-2)	3.8e-3 (8.4e-4)	1
FMF (x_1-pred)	2.1e-3 (5.8e-4)	2.0e-3 (3.1e-4)	4.6e-2 (1.1e-2)	2.1e-1 (3.6e-2)	5.9e-3 (9.7e-4)	1
GANO	4.6e-2 (3.0e-3)	7.3e-3 (3.6e-4)	1.7e+0 (1.3e+0)	3.1e-1 (8.4e-2)	**2.0e-3** (1.2e-3)	1
FFM-OT	6.7e-4 (5.4e-4)	3.9e-3 (2.6e-4)	2.4e-1 (4.7e-2)	7.7e-2 (9.0e-3)	2.5e-4 (1.7e-4)	386
FFM-VP	**4.2e-4** (4.8e-4)	**7.3e-4** (3.5e-4)	**1.9e-1** (6.1e-2)	**4.3e-2** (1.2e-2)	**1.3e-4** (1.0e-4)	290
FDDPM	4.4e-4 (4.4e-4)	1.3e-3 (4.6e-4)	2.5e-1 (1.9e-1)	5.9e-2 (1.2e-2)	1.9e-4 (1.2e-4)	1000
DDO	4.2e-3 (1.5e-3)	1.2e-2 (3.6e-4)	3.0e-1 (5.7e-2)	1.3e-1 (1.8e-2)	1.0e-3 (2.3e-4)	2000
Pop.						
FMF (u-pred)	7.1e-4 (2.1e-4)	**1.4e-3** (2.3e-4)	**2.0e-1** (8.8e-2)	**6.4e+0** (7.1e+0)	7.2e-3 (9.0e-4)	1
FMF (x_1-pred)	**1.7e-4** (1.2e-4)	1.6e-3 (1.9e-4)	3.7e-1 (1.1e-1)	1.5e+1 (1.7e+1)	**1.1e-4** (3.9e-5)	1
GANO	4.7e-3 (2.4e-3)	1.6e-3 (1.5e-3)	1.0e+0 (9.2e-1)	2.3e+1 (3.7e+1)	1.6e-1 (2.8e-1)	1
FFM-OT	6.0e-4 (7.5e-4)	1.6e-4 (1.6e-4)	1.1e-1 (6.7e-2)	**1.8e+0** (1.2e+0)	7.0e-4 (3.4e-4)	662
FFM-VP	**5.4e-4** (7.6e-4)	3.0e-4 (2.9e-4)	1.7e-1 (4.4e-2)	2.1e+0 (9.2e-1)	8.9e-2 (9.1e-3)	494
FDDPM	6.6e-4 (6.1e-4)	**1.2e-4** (1.2e-4)	**9.4e-2** (6.5e-2)	2.5e+0 (2.2e+0)	**3.0e-5** (9.2e-6)	1000
DDO	2.3e-3 (1.3e-3)	2.2e-1 (8.3e-3)	4.3e-1 (1.5e-2)	5.2e+0 (1.5e-1)	5.0e-1 (1.0e-2)	2000
GDP						
FMF (u-pred)	1.2e-3 (6.8e-4)	**2.9e-3** (5.1e-4)	2.9e-1 (8.4e-2)	2.4e+0 (9.7e-1)	1.0e-3 (2.5e-4)	1
FMF (x_1-pred)	**1.1e-3** (7.8e-4)	4.0e-3 (6.7e-4)	**2.2e-1** (8.9e-2)	**1.8e+0** (5.8e-1)	**2.9e-4** (1.9e-5)	1
GANO	9.6e+2 (3.1e+3)	7.4e+2 (2.3e+3)	5.8e-1 (2.2e-1)	2.4e+0 (1.0e+0)	7.1e-2 (1.9e-1)	1
FFM-OT	2.8e-2 (2.8e-3)	5.3e-3 (1.2e-3)	6.6e-1 (2.9e-1)	9.2e+0 (1.6e+1)	6.1e-4 (4.6e-4)	536
FFM-VP	2.8e-2 (3.4e-3)	4.9e-3 (1.2e-3)	5.3e-1 (1.2e-1)	3.2e+0 (1.4e+0)	8.7e-2 (1.0e-2)	494
FDDPM	**6.0e-4** (6.5e-4)	**5.3e-4** (5.3e-4)	**5.1e-2** (2.6e-2)	**7.2e-1** (4.0e-1)	**1.8e-4** (4.3e-5)	1000
DDO	1.3e-2 (2.6e-3)	1.5e-1 (9.9e-3)	3.6e-1 (1.6e-2)	1.9e+0 (1.0e-1)	3.8e-1 (8.5e-3)	2000
Labor						
FMF (u-pred)	5.3e-6 (2.5e-6)	**7.1e-8** (1.2e-8)	3.3e-1 (7.9e-2)	1.3e+1 (5.6e+0)	**1.1e-2** (2.9e-3)	1
FMF (x_1-pred)	**5.1e-6** (2.4e-6)	1.2e-7 (2.0e-8)	**2.7e-1** (5.9e-2)	7.9e+0 (4.3e+0)	2.1e-2 (4.3e-3)	1
GANO	4.7e-5 (3.8e-5)	2.4e-7 (1.6e-7)	6.6e-1 (2.2e-1)	**5.7e+0** (3.0e+0)	3.3e-2 (1.1e-2)	1
FFM-OT	1.0e-2 (1.2e-4)	4.2e-7 (1.7e-7)	1.1e+0 (4.7e-1)	2.5e+1 (5.1e+1)	5.5e-2 (4.5e-3)	308
FFM-VP	9.6e-3 (6.1e-5)	3.5e-7 (7.8e-8)	1.1e+0 (1.2e-1)	**7.0e+0** (9.5e-1)	2.6e-2 (4.0e-3)	320
FDDPM	**6.4e-6** (4.1e-6)	**6.1e-8** (6.2e-8)	**2.5e-1** (1.6e-1)	7.5e+0 (7.1e+0)	**1.1e-2** (5.2e-3)	1000
DDO	1.3e-5 (5.6e-6)	3.6e-6 (4.4e-7)	7.3e-1 (5.9e-2)	7.2e+0 (1.4e-1)	3.7e-1 (1.6e-2)	2000

ing of numerical solutions to the Navier–Stokes equations on a 2D torus [31]. We compare our method against several function-based generative models, including the multi-step approaches FDDPM [26], DDO [32], and FFM [27] in both its OT and VP variants, as well as the one-step functional generation method GANO [45]. The quantitative and qualitative results are summarized in Table 1 and Table 2. For the 1D datasets, following [27], we compute a set of statistical functionals: mean, variance, skewness, kurtosis, and autocorrelation for the generated functions, and evaluate the MSE between them and the corresponding ground-truth statistics from the dataset. For the 2D dataset, we evaluate the MSE between the generated and ground-truth Navier–Stokes solutions in terms of both density and spectral representations [27, 32, 45].

Across both 1D and 2D settings, our method achieves the best performance among one-step functional generation methods, while performing comparably to the best multi-step baselines such as FFM. Detailed descriptions of the training procedures, inference configurations, and evaluation metrics can be found in Appendix C.1.

4.2. Image Generation Based on Functional

Infty-Diff [4, 29] observed that purely Neural Operator–based functional generation methods struggle to scale to large datasets. To overcome this limitation, Infty-Diff introduced a hybrid sparse–dense Neural Operator that efficiently learns from higher-resolution functional data (e.g., 256×256 images). The model first employs a sparse Neural Operator to flexibly represent functions sampled at random points, followed by a dense U-Net/UNO backbone that refines features on a dense grid obtained through k-nearest-neighbor (KNN) sampling. We follow the network design of [4] and adopt the same model capacity, with only minimal modifications to convert the original multi-step diffusion formulation into a single-step FMF generation framework, and for the initial Gaussian measure, we employ white noise with a mollifier kernel, consistent with their implementation (see Appendix C.2 for details).

Table 2. MSEs between the density and spectra of the real and generated samples on the Navier–Stokes dataset. The best results for the 1-step and multi-step settings are highlighted in bold.

	Density ↓	Spectrum ↓
FMF (u-pred)	9.7e-5	1.2e3
FMF (x_1-pred)	**8.0e-5**	**5.6e2**
GANO	2.5e-3	3.2e4
FFM-OT	**3.7e-5**	**9.3e1**
DDPM	9.9e-5	5.0e2
DDO	2.9e-2	1.6e5

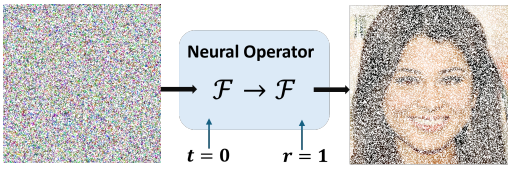


Figure 5. In Functional Mean Flow, both the input and output are modeled as continuous functions, enabling training and image generation to be defined over arbitrary pixel coordinates instead of being restricted to a discrete grid.

Although function-based image generation typically exhibits slightly lower perceptual fidelity compared to conventional pixel-based diffusion models, it provides significantly greater flexibility. As pointed out in Infty-Diff, functional-based models can be trained using only a random subset (e.g., one-quarter of the pixels) from a 256-resolution dataset, and can still generate images at arbitrary resolutions (e.g., 128, 256, or 512). Because the model operates directly in the functional space, its input and output can be defined on any pixel coordinates rather than being constrained to a fixed grid (see Figure 5).

Following Infty-Diff, we train our model on three unconditional image generation datasets, CelebA-HQ [24], FFHQ [25], and LSUN-Church [57], as well as one conditional generation dataset, AFHQ [6]. Qualitative results are shown in Figure 2 and Figure 4, with additional results provided in Appendix E.2. During training, the model observes only 25% of the pixels from 256×256 images, while at inference it generate images at 64, 128, 256, 512, and even 1024

Table 3. Evaluation of FID_{CLIP} [28] against previous infinite-dimensional approaches trained on coordinate subsets. For completeness, since several prior works report Inception FID, we additionally provide the Inception FID of our method, indicated with an asterisk (*). The best results for the 1-step and multi-step settings are highlighted in bold.

Method	Step	CelebAHQ-64	CelebAHQ-128	CelebAHQ-256	FFHQ-256	Church-256
D2F [9]	1	40.4*	–	–	–	–
GEM [8]	1	14.65	23.73	–	35.62	87.57
GASP [10]	1	9.29	27.31	–	24.37	37.46
FMF (Ours)	1	**3.48 (14.73*)**	**7.18 (30.35*)**	**9.17 (33.32*)**	**11.37 (37.67*)**	**26.57(35.63*)**
∞-Diff [4]	100	**4.57**	**3.02**	–	**3.87**	**10.36**
DPF [64]	1000	13.21*	–	–	–	–

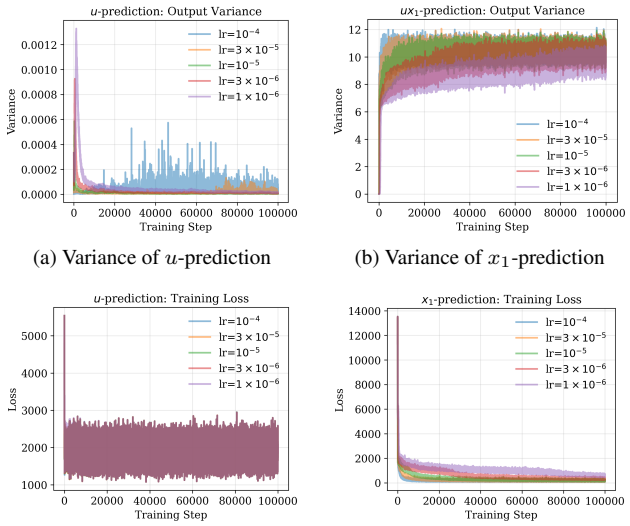
Table 4. Comparison of results across different resolutions. All results are trained on 256×256 images, using only $\frac{1}{4}$ of the pixels as input. Numbers are FID_{CLIP} [28]. The generation resolution is increased up to the maximum resolution of the dataset.

Dataset	64	128	256	512	1024
Unconditional Generation					
CelebA-HQ	3.48	5.86	9.17	9.70	10.96
FFHQ	4.42	7.70	11.37	12.34	–
LSUN-Church	12.07	17.89	26.51	–	–
Conditional Generation					
AFHQ	3.10	6.19	9.24	11.55	–

resolutions. We report quantitative comparisons with other function-based methods in Table 3, where following Infty-Diff we primarily evaluate using the FID_{CLIP} [28] metric to assess function-based generative methods. For completeness, we also report conventional FID scores [19] for reference. Since our function-based generation framework is inherently resolution-agnostic, we evaluate models trained on 256-resolution datasets at 64, 128, 256, 512, and 1024 resolutions, and report the corresponding FID_{CLIP} results in Table 4. Our model achieves state-of-the-art performance among one-step function-based methods and produces results comparable to the multi-step function-based generation of Infty-Diff. Additional details on the training setup, inference procedure, and the internal structure of Infty-Diff are provided in Appendix C.2.

4.3. 3D Shape Generation

To further validate the applicability of our method to function-based generation tasks, we extend it to SDF-based 3D shape generation, where shape generation is achieved by directly generating its SDF. We adopt the framework of Functional Diffusion [58], where both the input function and output function are represented by randomly sampled points and their corresponding function values: the input function f_c is represented by a set of context points $\{x_c^i\}_{i=1}^n$ with context values $\{v_c^i\}_{i=1}^n$, where $v_c^i = f_c(x_c^i)$, and the output function f_q is represented by query points



(a) Variance of u-prediction (b) Variance of x_1-prediction

(c) Loss of u-prediction (d) Loss of x_1-prediction

Figure 6. Training behavior of u- vs. x_1-prediction FMF under varying learning rates. The u-prediction model exhibits spatial-variance collapse and unstable losses, whereas the x_1-prediction model maintains stable variance and smooth optimization.

$\{x_q^j\}_{j=1}^m$ and their predicted query values $\{v_q^j\}_{j=1}^m$, where $v_q^j = f_q(x_q^j)$. Following the Perceiver [23] framework, Functional Diffusion performs cascaded cross- and self-attention between the context embedding and the learnable functional vector $\mathcal{X}$. It then applies cross-attention with the query points $\{x_q^j\}_{j=1}^m$ to generate the corresponding query values $\{v_q^j\}_{j=1}^m$, yielding the output function represented as $(\{x_q^j\}_{j=1}^m, \{v_q^j\}_{j=1}^m)$. We follow this network design and adopt the same model capacity for a fair comparison, and consistent with Functional Diffusion, we construct the initial Gaussian measure using linear interpolation over a coarse grid (see Appendix C.3 for details).

In our experiments, however, we found that this framework is not well-suited for u-prediction FMF: training becomes unstable even at small learning rates, with severe collapse. To illustrate this behavior, we perform a 2D experiment on MNIST [30], converted into signed distance fields (SDFs) and trained under the Functional Diffusion framework using FMF. We monitor the batch-averaged spatial variance $\frac{1}{m}\sum_{j=1}^m \big(v_q^j - \frac{1}{m}\sum_{j=1}^m v_q^j\big)^2$ of the network output: an SDF should satisfy $|\nabla f| = 1$ and therefore maintain nontrivial spatial variation. Once the variance vanishes and remains near zero for an extended period (collapse), the model degenerates into a constant field and cannot recover. The experimental results are shown in Figure 6, and see Appendix E.1 for details and evidence that a collapsed u-prediction model cannot generate meaningful outputs.

We further train our x_1-prediction FMF on the 3D ShapeNet-CoreV2 [5] dataset, following the same preprocessing method as [58], where each mesh is converted to a

voxelized SDF and then sampled into point–value pairs. As reported in [58], function-based 3D shape generation models can solve the challenging task of reconstructing an entire SDF field from as few as 64 surface points $\{C_l\}_{l=1}^{64}$. As demonstrated in Table 5 and Figure 7, our one-step formulation achieves this task with comparable accuracy, highlighting the robustness and effectiveness of the proposed x_1-prediction FMF. Details on the dataset processing, task and metrics description, model architecture, and training and inference procedures are provided in Appendix C.3.

Table 5. Quantitative comparison of reconstruction quality. The model is trained on the ShapeNet dataset, where the conditional input consists of 64 points sampled from the target surface. The model is required to reconstruct the surface based on these 64 points. Step denotes the number of inference steps.

Method	Step	Chamfer ↓	F-Score ↑	Boundary ↓
Ours	1	0.060	0.584	0.011
3DS2VS	18	0.144	0.608	0.016
FD	64	0.101	0.707	0.012

(↓ lower is better; ↑ higher is better.)

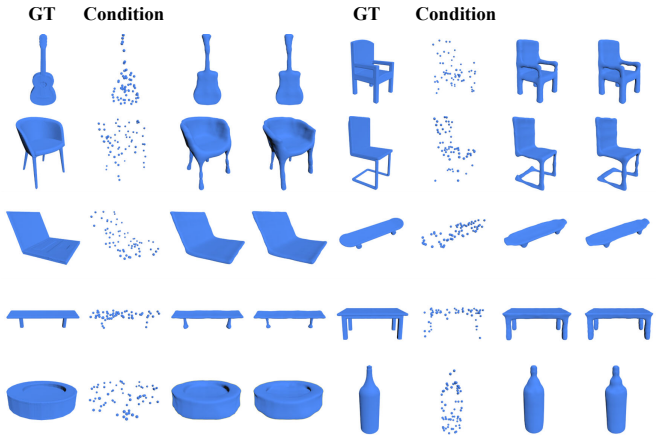


Figure 7. Results of 3D shape generation. This is a highly challenging task [58], where the generative model is **ONLY** conditioned on 64 randomly sampled points from the target surface and required to reconstruct the entire geometry. We apply the x_1-prediction FMF within the Functional Diffusion framework, reducing the original 64-step generation process to a single step. The **GT** column shows the ground-truth surfaces, while the **Condition** column visualizes the 64 conditioning points provided to the model.

5. Related Work

Functional Generation. Functional generation extends generative modeling to infinite-dimensional settings, drawing theoretical support from stochastic equations on Hilbert space [7]. It enables both training and sampling at arbitrary resolutions, making large-scale generation more computationally feasible. For instance, Infty-Brush [29] demonstrates controllable image generation at resolutions up to 4096 × 4096 pixels. Recent studies have investigated discrete-time diffusion models on Hilbert space [1, 14, 26, 32, 41, 58, 64], while concurrent works have explored their continuous-time counterparts [13, 18, 42]. Distinct from functional diffusion models, Functional Flow Matching [27] avoids injecting random noise during generation, enabling the production of high-quality samples with fewer NFE (Number of Function Evaluations). Beyond diffusion and flow-based approaches, researchers have also proposed functional GANs [39, 45] and functional energy-based models [33], further enriching the landscape of infinite-dimensional generative modeling. A recent concurrent work [61] extends Rectified Flow to function spaces and achieves strong results in one-step functional generation.

Few-step Diffusion/Flow Models. Reducing the sampling steps is vital to improve the efficiency of diffusion/flow models. Distillation techniques play a key role in enabling few-step generation. Several studies have explored distilling diffusion models [15, 37, 38, 46, 47, 56, 63] and flow models [35]. In parallel, consistency models [52] were introduced as independently trainable one-step generators that do not rely on distillation. Subsequent works have focused on enhancing their training stability and sample quality [17, 36, 49]. Inspired by consistency models, recent research has incorporated self-consistency principles into related frameworks, such as enforcing consistency in the velocity field of Flow Matching [55], Shortcut Model [12], and stochastic interpolation across time steps [62]. While standard consistency models rely on a single time variable, Flow Map Matching [2] learns displacement maps parameterized by two time variables. Mean Flow [16] further extend this idea by learning the average velocity over time via the time derivative of the Mean Flow identity, achieving state-of-the-art one-step generation performance on ImageNet.

6. Conclusion

We proposed Functional Mean Flow as a unified one-step flow matching framework in infinite-dimensional Hilbert space. We introduced an x_1-prediction variant of Mean Flow, which exhibits improved training stability and robustness over the original u-prediction formulation. Experiments on image-function synthesis, 3D signed distance field modeling, solving PDEs, and time-series prediction demonstrate the versatility and effectiveness of our method. Future work will explore broader functional modalities and further investigate the advantages of the x_1-prediction formulation beyond the current domains.

Acknowledgments

We express our gratitude to the anonymous reviewers for their insightful feedback. Georgia Tech authors acknowledge NSF IIS #2433322, ECCS #2318814, CAREER #2433307, IIS #2106733, OISE #2433313, and CNS #1919647 for funding support.

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8

[2] Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research (TMLR)*, 2025. 4, 8

[3] Jutta Bolt and Jan Luiten Van Zanden. Maddison-style estimates of the evolution of the world economy: A new 2023 update. *Journal of Economic Surveys*, 39(2):631–671, 2025. 5

[4] Sam Bond-Taylor and Chris G Willcocks. ∞-diff: Infinite resolution diffusion with subsampled mollified states. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 2, 4, 5, 6, 7

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[7] Giuseppe Da Prato and Jerzy Zabczyk. *Stochastic equations in infinite dimensions*. Cambridge university press, 2014. 8

[8] Yilun Du, Katie Collins, Josh Tenenbaum, and Vincent Sitzmann. Learning signal-agnostic manifolds of neural fields. 2021. 7

[9] Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *International Conference on Machine Learning (ICML)*, 2022. 1, 7

[10] Emilien Dupont, Yee Whye Teh, and Arnaud Doucet. Generative models as distributions of functions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022. 1, 7

[11] Manuel Febrero-Bande and Manuel Oviedo De La Fuente. Statistical computing in functional data analysis: The r package fda. usc. *Journal of statistical Software*, 51:1–28, 2012. 5

[12] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025. 8

[13] Giulio Franzese, Giulio Corallo, Simone Rossi, Markus Heinonen, Maurizio Filippone, and Pietro Michiardi. Continuous-time functional diffusion processes. In *Neural Information Processing Systems (NeurIPS)*, 2023. 8

[14] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10021–10030, 2023. 8

[15] Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. In *Neural Information Processing Systems (NeurIPS)*, 2023. 1, 8

[16] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Neural Information Processing Systems (NeurIPS)*, 2025. 1, 3, 4, 8

[17] Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. In *International Conference on Learning Representations (ICLR)*, 2025. 8

[18] Paul Hagemann, Sophie Mildenberger, Lars Ruthotto, Gabriele Steidl, and Nicole Tianjiao Yang. Multilevel diffusion: Infinite dimensional score-based diffusion models for image generation. *SIAM Journal on Mathematics of Data Science*, 2025. 8

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems (NeurIPS)*, 2017. 7

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Neural Information Processing Systems (NeurIPS)*, 2020. 1

[21] Robert Inklaar, Herman de Jong, Jutta Bolt, and Jan Luiten Van Zanden. Rebasing'maddison': new income comparisons and the shape of long-run economic development. 2018. 5

[22] International Monetary Fund. International financial statistics: Prices, production, and labor, labor force, 2022. 5

[23] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021. 7

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations (ICLR)*, 2018. 6

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[26] Gavin Kerrigan, Justin Ley, and Padhraic Smyth. Diffusion generative models in infinite dimensions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023. 5, 6, 8

[27] Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. In *International Conference on*

Artificial Intelligence and Statistics (AISTATS), 2023. 1, 2, 3, 5, 6, 8

[28] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fr\'echet inception distance. In *International Conference on Learning Representations (ICLR)*, 2023. 7

[29] Minh-Quan Le, Alexandros Graikos, Srikar Yellapragada, Rajarsi Gupta, Joel Saltz, and Dimitris Samaras. ∞-brush: Controllable large image synthesis with diffusion models in infinite dimensions. In *European Conference on Computer Vision (ECCV)*, 2024. 5, 6, 8

[30] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 7

[31] Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning chaotic dynamics in dissipative systems. In *Neural Information Processing Systems (NeurIPS)*, 2022. 6

[32] Jae Hyun Lim, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, et al. Score-based diffusion models in function space. *Journal of Machine Learning Research (JMLR)*, 2025. 5, 6, 8

[33] Jen Ning Lim, Sebastian Vollmer, Lorenz Wolf, and Andrew Duncan. Energy-based models for functional data using path measure tilting. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023. 8

[34] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 1

[35] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 8

[36] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations (ICLR)*, 2025. 8

[37] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *Neural Information Processing Systems (NeurIPS)*, 2024. 8

[38] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8

[39] Evangelos Ntavelis, Mohamad Shahbazi, Iason Kastanis, Radu Timofte, Martin Danelljan, and Luc Van Gool. Arbitrary-scale image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11533–11542, 2022. 8

[40] David A Orlando, Charles Y Lin, Allister Bernard, Jean Y Wang, Joshua ES Socolar, Edwin S Iversen, Alexander J Hartemink, and Steven B Haase. Global control of cell-cycle transcription by coupled cdk and network oscillators. *Nature*, 453(7197):944–947, 2008. 5

[41] Dogyun Park, Sihyeon Kim, Sojin Lee, and Hyunwoo J Kim. Ddmi: Domain-agnostic latent diffusion models for synthesizing high-quality implicit neural representations. *arXiv preprint arXiv:2401.12517*, 2024. 8

[42] Jakiw Pidstrigach, Youssef Marzouk, Sebastian Reich, and Sven Wang. Infinite-dimensional diffusion models for function spaces. *Journal of Machine Learning Research (JMLR)*, 2024. 8

[43] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10619–10629, 2022.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[45] Md Ashiqur Rahman, Manuel A. Florez, Anima Anandkumar, Zachary E. Ross, and Kamyar Azizzadenesheli. Generative adversarial neural operators. *Transactions on Machine Learning Research (TMLR)*, 2022. 5, 6, 8

[46] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. 8

[47] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision (ECCV)*, 2024. 8

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[49] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations (ICLR)*, 2023. 8

[50] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Neural Information Processing Systems (NeurIPS)*, 2019. 1

[51] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[52] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023. 3, 4, 8

[53] Andrew M Stuart. Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559, 2010.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[55] 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. *arXiv preprint arXiv:2407.02398*, 2024. 8

[56] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 8

[57] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[58] Biao Zhang and Peter Wonka. Functional diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4, 5, 7, 8

[59] Biao Zhang, Matthias Nießner, and Peter Wonka. 3dilg: Irregular latent grids for 3d generative modeling. *Advances in Neural Information Processing Systems*, 35:21871–21885, 2022.

[60] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

[61] Jianxin Zhang and Clayton Scott. Flow straight and fast in hilbert space: Functional rectified flow. *arXiv preprint arXiv:2509.10384*, 2025. 8

[62] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *International Conference on Machine Learning (ICML)*, 2025. 8

[63] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning (ICML)*, 2024. 8

[64] Peiye Zhuang, Samira Abnar, Jiatao Gu, Alex Schwing, Joshua M Susskind, and Miguel Angel Bautista. Diffusion probabilistic fields. In *International Conference on Learning Representations (ICLR)*, 2023. 7, 8