In this appendix, we provide background theorems of Functional Flow Matching (Appendix A), theoretical derivations of our method (Appendix B),implementation details (Appendix C), example Python code (Appendix D), and additional experiments and results (Appendix E).

A. Related Theorem in Functional Flow Matching

In this section, we elaborate on key theorems from [27], which provide the theoretical foundation for Functional Flow Matching. Functional Flow Matching aims to learn u_t^θ by minimizing $\mathcal{L}(\theta) = \mathbb{E}_{t,g\sim\mu_t}\left[\|u_t(g) - u_t^\theta(g)\|_{\mathcal{F}}^2\right]$. However, since the reference function $u_t(g)$ here does not exist in closed form, Functional Flow Matching construct a conditional velocity field $u_t^f(g)$ to serve as the optimization target for $u_t^\theta(g)$ instead.

The conditional velocity $u_t^f(g)$ induces a flow ϕ_t^f by Equation 3 that push-forward μ_0 to $\mu_t^f = (\phi_t^f)_\sharp \mu_0$. With μ_t^f, the marginal measure path μ_t and the marginal velocity field u_t can be obtained by taking the expectation with respect to the data measure ν by Equation 4. However, the connection between the marginal measure path μ_t and the velocity field u_t, derived from the expectations of their conditional counterparts μ_t^f and u_t^f, is non-trivial. Under some smoothness conditions, [27] established this relationship in the following theorem.

Theorem A.1. *Assume that $\int_0^1 \int_{\mathcal{F}} \int_{\mathcal{F}} \|u_t^f(g)\| \mathrm{d}\mu_t^f(g) \mathrm{d}\nu(f) \mathrm{d}t < \infty$. If $\mu_t^f \ll \mu_t$ for ν-a.e. f and almost every $t \in [0,1]$, then the vector field*

$$u_t(g) = \int_{\mathcal{F}} u_t^f(g) \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g) \mathrm{d}\nu(f) \tag{15}$$

generates the marginal path of measures $(\mu_t)_{t\in[0,1]}$ which are defined as $\mu_t(A) = \int_{\mathcal{F}} \mu_t^f(A) \mathrm{d}\nu(f)$ via Equation 4, $\forall A \in \mathcal{B}(\mathcal{F})$. That is, $(u_t)_{t\in[0,1]}$ and $(\mu_t)_{t\in[0,1]}$ jointly satisfy the continuity Equation 1. Here, $\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}$ denotes the Radon–Nikodym derivative of the conditional measure with respect to the marginal.

In the theorem above, the condition $\mu_t^f \ll \mu_t$ denotes that μ_t^f is absolutely continuous with respect to μ_t, while ν-a.e. indicates that the statement holds for ν-almost every f. However, the requirement $\mu_t^f \ll \mu_t$ for ν-a.e. f and almost every $t \in [0,1]$ is generally difficult to guarantee, since the marginal distribution μ_t itself is hard to know. To address this issue, [27] established the following theorem, which reformulates this condition in terms of the known, constructible conditional measures μ_t^f.

Theorem A.2. *Consider a probability measure ν on $\mathcal{F}$ and a collection of measures μ_t^f parameterized by $f \in \mathcal{F}$. Suppose that the collection of parameterized measures μ_t^f is ν-a.e. mutually absolutely continuous. Define the marginal measure μ_t via Equation 4. Then, $\mu_t^f \ll \mu_t$ for ν-a.e. f.*

Furthermore, [27] employed the Feldman–Hájek theorem [7] together with Lemma 6.15 of [53] to show that, under the conditional velocity and corresponding conditional measures chosen in Equation 5, the assumption in the above theorem that the measures μ_t^f are ν-a.e. mutually absolutely continuous holds as long as the data distribution is supported on the Cameron–Martin space of C_0, i.e., $\nu(C_0^{1/2}(\mathcal{F})) = 1$. In practice, one typically does not verify whether the data distribution is supported on $C_0^{1/2}(\mathcal{F})$; instead, this condition is often enforced implicitly through data preprocessing [4, 33]. However, in the experiments reported in [27], the authors observed that such preprocessing was not strictly necessary.

Using the constructed conditional velocity, the conditional loss can be written as $\mathcal{L}_c(\theta) = \mathbb{E}_{t,f,g\sim\mu_t^f}\left[\|u_t^f(g) - u_t^\theta(g)\|_{\mathcal{F}}^2\right]$ in Equation 6. Based on the above Theorem A.1, [27] proved that optimizing $\mathcal{L}(\theta)$ is equivalent to optimizing $\mathcal{L}_c(\theta)$, where the optimization reference function $u_t^f(g)$ is known, thus allowing directly optimizing $\mathcal{L}_c(\theta)$ to obatin $u_t^\theta(g)$.

Theorem A.3. *Assume that the true and model vector fields are square-integrable, i.e., $\int_0^1 \int_{\mathcal{F}} \|u_t(g)\|_{\mathcal{F}}^2 \mathrm{d}\mu_t(g) \mathrm{d}t < \infty$ and $\int_0^1 \int_{\mathcal{F}} \|u_t^\theta(g\|_{\mathcal{F}}^2 \mathrm{d}\mu_t(g) \mathrm{d}t < \infty$. Then, $\mathcal{L}(\theta) = \mathcal{L}_c(\theta) + C$, where $C \in \mathbb{R}$ is a constant independent of θ.*

B. Missing Proofs

In the following analysis, for notational simplicity and without causing ambiguity, we will use the same symbol $\|\cdot\|$ to denote both the norm $\|h\|_{\mathcal{F}}$ of a function $h \in \mathcal{F}$ and the operator norm $\|h\|_{\mathcal{L}(\mathcal{F},\mathcal{F})}$ of a bounded linear map $O : \mathcal{F} \to \mathcal{F}$. For instance, $Dv_t(g)$ is a bounded linear operator, mapping $h \in \mathcal{F}$ to $Dv_t(g)[h] \in \mathcal{F}$, so its norm is written as $\|Dv_t(g)\|$ for simplicity.

B.1. Proof of Statement 1

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

Proof. The theorem essentially reveals a discrepancy between the mean of instantaneous velocity fields and the mean of non-instantaneous flow trajectories.

To prove that $\phi^{(1)}_{t\to r}(g) \neq \phi^{(2)}_{t\to r}(g)$, it suffices to show that $\frac{\mathrm{d}}{\mathrm{d}r}\phi^{(1)}_{t\to r}(g) \neq \frac{\mathrm{d}}{\mathrm{d}r}\phi^{(2)}_{t\to r}(g)$ at $t=0$. We therefore compute and compare $\frac{\mathrm{d}}{\mathrm{d}r}\phi^{(1)}_{t\to r}(g)$ and $\frac{\mathrm{d}}{\mathrm{d}r}\phi^{(2)}_{t\to r}(g)$ when $t=0$.

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}r}\phi^{(1)}_{t\to r}(g)\\
&\overset{①}{=} \frac{\mathrm{d}}{\mathrm{d}r}\int_{\mathcal{F}} \phi^f_{t\to r}(g)\frac{\mathrm{d}\mu^f_t}{\mathrm{d}\mu_t}(g)\mathrm{d}\nu(f)\\
&= \int_{\mathcal{F}} \frac{\mathrm{d}}{\mathrm{d}r}(\phi^f_{t\to r}(g))\frac{\mathrm{d}\mu^f_t}{\mathrm{d}\mu_t}(g)\mathrm{d}\nu(f)\\
&\overset{②}{=} \int_{\mathcal{F}} u^f_r(\phi^f_{t\to r}(g))\frac{\mathrm{d}\mu^f_t}{\mathrm{d}\mu_t}(g)\mathrm{d}\nu(f)\\
&\overset{③}{=} \int_{\mathcal{F}} u^f_r(\phi^f_r(g))\mathrm{d}\nu(f)\\
&\overset{④}{=} \int_{\mathcal{F}} (f-(1-\sigma_{\min})g)\mathrm{d}\nu(f)\\
&= \int_{\mathcal{F}} f\mathrm{d}\nu(f) - (1-\sigma_{\min})g\\
&= m_\nu - (1-\sigma_{\min})g,
\end{aligned} \tag{16}$$

where m_ν denotes the expectation of the dataset distribution ν, which is a constant depending only on the dataset. In the above derivation, ① follows from the definition of $\phi^{(1)}_{t\to r}$, ② follows from the definition of the two-parameter flow introduced in section 3, ③ substitutes $t=0$ and uses the facts that $\mu^f_0 = \mu_0$ and $\phi^f_{0\to r} = \phi^f_r$, and ④ follows from the specific choice of the conditional flow defined in Equation 5.

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}r}\phi^{(2)}_{t\to r}(g)\\
&\overset{①}{=} \frac{\mathrm{d}}{\mathrm{d}r}\phi_r\circ\phi_t^{-1}(g)\\
&\overset{②}{=} u_r(\phi_r\circ\phi_t^{-1}(g))\\
&\overset{③}{=} u_r(\phi_r(g)).
\end{aligned} \tag{17}$$

In the above derivation, ① follows from the definition of $\phi^{(2)}_{t\to r}$, ② follows from the definition of the flow in Equation 3, and ③ substitutes $t=0$ and uses the facts that $\phi_0 = \mathrm{Id}_{\mathcal{F}}$.

For $\frac{\mathrm{d}}{\mathrm{d}r}\phi^{(1)}_{t\to r}(g) - \frac{\mathrm{d}}{\mathrm{d}r}\phi^{(2)}_{t\to r}(g)$ we have:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}r}\phi^{(1)}_{t\to r}(g) - \frac{\mathrm{d}}{\mathrm{d}r}\phi^{(2)}_{t\to r}(g)\\
&= m_\nu - (1-\sigma_{\min})g - u_r(\phi_r(g)).
\end{aligned} \tag{18}$$

Therefore, for $\phi^{(1)}_{t\to r} = \phi^{(2)}_{t\to r}$ to hold, a necessary condition is $\dot{\phi}_r(g) = u_r(\phi_r(g)) = m_\nu - (1-\sigma_{\min})g, \forall g\in\mathcal{F}$ and $t\in[0,1]$, which can be solved as

$$\phi_r(g) = (m_\nu - (1-\sigma_{\min})g)r + g, \forall g\in\mathcal{F},\ t\in[0,1]. \tag{19}$$

This implies that $\nu = \mu_1 = (\phi_1)_\sharp\mu_0 = \mathcal{N}(m_\nu, \sigma^2_{\min}C_0)$, which contradicts the arbitrariness of the dataset distribution ν. Therefore, the equality $\phi^{(1)}_{t\to r} = \phi^{(2)}_{t\to r}$ cannot be satisfied. □

B.2. Supporting Lemmas for Theorem 3.1

As the computation of the Fréchet derivative of $\phi_{t\to r}$ is required in Theorem 3.1, we begin by establishing the following lemma, which asserts the Fréchet differentiability of $\phi_{t\to r}$.

Lemma B.1 (Fréchet differentiability of $\phi_{t\to r}$ in Hilbert space)**.** *For every radius $R>0$ and $\mathcal{B}_R = \{g\in\mathcal{F} | \|g\| < R\}$, assume $\{u_t\}_{t\in[0,1]}$ satisfies:*

(A1) *(Continuity) $u(t,x)$ is measurable and integrable in t and Lipschitz continuous in $x\in\mathcal{B}_R$, which means there exists integrable $L_R\in L^1(0,1)$, i.e., $\int_0^1 |L_R(t)|\mathrm{d}t<\infty$ such that*

$$\|u_t(x)-u_t(y)\| \le L_R(t)\|x-y\|, \forall x,y\in\mathcal{B}_R; \tag{20}$$

(A2) *(Bounded Fréchet differentiability) For each t, $u_t \in C^1(\mathcal{F};\mathcal{F})$ is continuously Fréchet differentiable, and there exists integrable $M_R\in L^1(0,1)$, i.e., $\int_0^1 |M_R(t)|\mathrm{d}t<\infty$ such that*

$$\|Du_t(x)\| \le M_R(t), \forall x\in\mathcal{B}_R. \tag{21}$$

Then the associated two-parameter flow $\phi_{t\to r}(g)$ is continous for t and r and is Fréchet differentiable for all $g\in\mathcal{B}_R$ satisfying $\phi_{t\to\tau}(g)\in\mathcal{B}_R$ for all $\tau\in[t,r]$. Let $J_r(g) = D\phi_{t\to r}(g)$. Then $J_r(g)$ satisfies the equation

$$\frac{\partial}{\partial r}J_r(g) = Du_r(\phi_{t\to r}(g))\circ J_r(g), \quad J_t(g) = I, \tag{22}$$

and $D\phi_{t\to r}(g)$ is continuous with respect to g.

Proof. Under (A1), the existence and uniqueness theorem for ODEs on Banach spaces ensures that for each $g\in\mathcal{B}_R$, Equation 3 admits a unique solution ϕ_t, and $\phi_t(g)$ is continuous with respect to both g and t. Consequently, the two-parameter flow $\phi_{t\to r}(g) = \phi_r(\phi_t^{-1}(g))$ is continuous with respect to t, r, and g.

Define $\tilde{\mathcal{B}}^{t,r}_R = \{g\in\mathcal{B}_R \mid \phi_{t\to\tau}(g)\in\mathcal{B}_R, \forall\tau\in[t,r]\}$. For any $x,y\in\tilde{\mathcal{B}}^{t,r}_R$, we have:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}r}\|\phi_{t\to r}(x)-\phi_{t\to r}(y)\|^2\\
&= 2\langle\phi_{t\to r}(x)-\phi_{t\to r}(y), \frac{\mathrm{d}}{\mathrm{d}r}\phi_{t\to r}(x) - \frac{\mathrm{d}}{\mathrm{d}r}\phi_{t\to r}(y)\rangle\\
&\le 2\|\phi_{t\to r}(x)-\phi_{t\to r}(y)\|\|\frac{\mathrm{d}}{\mathrm{d}r}\phi_{t\to r}(x) - \frac{\mathrm{d}}{\mathrm{d}r}\phi_{t\to r}(y)\|\\
&\le 2\|\phi_{t\to r}(x)-\phi_{t\to r}(y)\|\|u_r(\phi_{t\to r}(x)) - u_r(\phi_{t\to r}(y))\|\\
&\le 2L_R(r)\|\phi_{t\to r}(x)-\phi_{t\to r}(y)\|^2.
\end{aligned} \tag{23}$$

The above inequality can be solved as $\|\phi_{t\to r}(x) - \phi_{t\to r}(y)\| \le e^{\int_t^r L_R(\tau)d\tau}\|x-y\|$ which implies that $\phi_{t\to r}$ is Lipschitz continuous on $\tilde{\mathcal{B}}_R^{t,r}$.

Now we proceed to prove that $\phi_{t\to r}$ is Fréchet differentiable in $\tilde{\mathcal{B}}_R^{t,r}$. For any $g \in \tilde{\mathcal{B}}_R^{t,r}$ and $h \in \mathcal{F}$, since $\phi_{t\to r}$ is Lipschitz continuous on $\tilde{\mathcal{B}}_R^{t,r}$, it follows that when ϵ is sufficiently small, we can have $g + \epsilon h \in \tilde{\mathcal{B}}_R^{t,r}$. Consider the difference quotient $\eta_\tau^\epsilon = \frac{\phi_{t\to\tau}(g+\epsilon h) - \phi_{t\to\tau}(g)}{\epsilon}$. Denote the difference as $\Delta_\tau^\epsilon = \phi_{t\to\tau}(g+\epsilon h) - \phi_{t\to\tau}(g)$. Since the flow $\phi_{t\to r}$ can be expressed as the integral of its velocity field $\phi_{t\to r}(g) = g + \int_t^r u_\tau\big(\phi_{t\to\tau}(g)\big)\mathrm{d}\tau$, we thus have:

$$\begin{aligned}\eta_r^\epsilon &= \frac{\phi_{t\to r}(g+\epsilon h) - \phi_{t\to r}(g)}{\epsilon} \\ &= h + \frac{1}{\epsilon}\int_t^r u_\tau\big(\phi_{t\to\tau}(g+\epsilon h)\big) - u_\tau\big(\phi_{t\to\tau}(g)\big)\mathrm{d}\tau \\ &= h + \frac{1}{\epsilon}\int_t^r u_\tau\big(\phi_{t\to\tau}(g) + \Delta_\tau^\epsilon\big) - u_\tau\big(\phi_{t\to\tau}(g)\big)\mathrm{d}\tau.\end{aligned} \tag{24}$$

Since each u_τ is continuously Fréchet differentiable, we can integrate the derivative Du_τ along the line segment connecting $x \in \mathcal{F}$ and $x + v \in \mathcal{F}$ as

$$u_\tau(x+v) - u_\tau(x) = \int_0^1 Du_\tau(x+\theta v)[v]\mathrm{d}\theta. \tag{25}$$

By taking $x = \phi_{t\to\tau}(g)$ and $v = \Delta_\tau^\epsilon$ in the above expression, we obtain:

$$\begin{aligned}\eta_r^\epsilon &= h + \frac{1}{\epsilon}\int_t^r u_\tau\big(\phi_{t\to\tau}(g) + \Delta_\tau^\epsilon\big) - v_\tau\big(\phi_{t\to\tau}(g)\big)\mathrm{d}\tau \\ &= h + \frac{1}{\epsilon}\int_t^r \int_0^1 Du_\tau(\phi_{t\to\tau}(g) + \theta\Delta_\tau^\epsilon)[\Delta_\tau^\epsilon]\mathrm{d}\theta\mathrm{d}\tau \\ &= h + \int_t^r \int_0^1 Du_\tau(\phi_{t\to\tau}(g) + \theta\Delta_\tau^\epsilon)[\frac{\Delta_\tau^\epsilon}{\epsilon}]\mathrm{d}\theta\mathrm{d}\tau \\ &= h + \int_t^r K_\tau^\epsilon[\eta_\tau^\epsilon]\mathrm{d}\tau,\end{aligned} \tag{26}$$

where we write $K_\tau^\epsilon = \int_0^1 Du_\tau(\phi_{t\to\tau}(g) + \theta\Delta_\tau^\epsilon)\mathrm{d}\theta$. By assumption (A2), we have $|K_\tau^\epsilon| \le M_R(\tau)$ a.e. Therefore, by applying the Grönwall's inequality, we obtain:

$$\sup_{\tau\in[t,r]} \|\eta_\tau^\epsilon\| \le \|h\| e^{\int_t^r M_R(\xi)\mathrm{d}\xi}. \tag{27}$$

Hence, η_τ^ϵ is uniformly bounded for all $\tau \in [t,r]$. Since $\Delta_\tau^\epsilon = \epsilon\eta_\tau^\epsilon$, we have $\Delta_\tau^\epsilon \to 0$ as $\epsilon \to 0$. Because Du_τ is continuous, we have $K_\tau^\epsilon = \int_0^1 Du_\tau\big(\phi_{t\to\tau}(g) + \theta\Delta_\tau^\epsilon\big)\mathrm{d}\theta \to K_\tau^0 = \int_0^1 Du_\tau\big(\phi_{t\to\tau}(g)\big)\mathrm{d}\theta$ pointwise in τ as $\epsilon \to 0$. Moreover, by assumption (A2), the following inequality holds:

$$\|K_\tau^\epsilon\| \le \int_0^1 \|Du_\tau(\phi_{t\to\tau}(g) + \theta\Delta_\tau^\epsilon)\|\mathrm{d}\theta \le M_R(\tau). \tag{28}$$

Therefore, by the Dominated Convergence Theorem, we have $\int_t^r \|K_\tau^\epsilon - K_\tau^0\|\mathrm{d}\tau \to 0$.

Finally, we use the above results to prove the convergence of η_r^ϵ. It suffices to show that the family η_r^ϵ forms a Cauchy sequence. For sufficiently small $\epsilon, \epsilon' > 0$, we compute the difference

$$\begin{aligned}\eta_r^\epsilon - \eta_r^{\epsilon'} &= \int_t^r K_\tau^\epsilon[\eta_\tau^\epsilon]\mathrm{d}\tau - \int_t^r K_\tau^{\epsilon'}[\eta_\tau^{\epsilon'}]\mathrm{d}\tau \\ &= \int_t^r K_\tau^\epsilon[\eta_\tau^\epsilon - \eta_\tau^{\epsilon'}]\mathrm{d}\tau + \int_t^r (K_\tau^\epsilon - K_\tau^{\epsilon'})[\eta_\tau^{\epsilon'}]\mathrm{d}\tau.\end{aligned} \tag{29}$$

Applying the triangle inequality yields

$$\begin{aligned}\|\eta_r^\epsilon - \eta_r^{\epsilon'}\| \le &\int_t^r \|K_\tau^\epsilon\|\|\eta_\tau^\epsilon - \eta_\tau^{\epsilon'}\|\mathrm{d}\tau \\ &+ \int_t^r \|K_\tau^\epsilon - K_\tau^{\epsilon'}\|\|\eta_\tau^{\epsilon'}\|.\mathrm{d}\tau.\end{aligned} \tag{30}$$

Since $\int_t^r \|K_\tau^\epsilon - K_\tau^0\|\mathrm{d}\tau \to 0$, define $A_{\epsilon,\epsilon'} = |h|\exp^{\int_t^r M(\xi)\mathrm{d}\xi}\int_t^r \|K_\tau^\epsilon - K_\tau^0\| + \|K_\tau^{\epsilon'} - K_\tau^0\|\mathrm{d}\tau$, then $A_{\epsilon,\epsilon'} \to 0$ and $\int_t^r \|K_\tau^\epsilon\|\|\eta_\tau^\epsilon - \eta_\tau^{\epsilon'}\|\mathrm{d}\tau \le A_{\epsilon,\epsilon'}$. Let $q_r = \|\eta_\tau^\epsilon - \eta_\tau^{\epsilon'}\|$, then $q_r \le A_{\epsilon,\epsilon'} + \int_t^r q_\tau M_R(\tau)\mathrm{d}\tau$. Applying the integral form of Grönwall's inequality gives

$$q_r \le A_{\epsilon,\epsilon'} e^{\int_t^r M_R(\tau)d\tau} \tag{31}$$

Since $A_{\epsilon,\epsilon'} \to 0$ and $\int_t^r M_R(\tau)\mathrm{d}\tau$ is bounded, we obtain $q_r = \|\eta_\tau^\epsilon - \eta_\tau^{\epsilon'}\| \to 0$, which shows that η_r^ϵ is a Cauchy sequence and hence convergent. Therefore, $\phi_{t\to r}(g)$ is Gâteaux differentiable. Moreover, since K_τ^ϵ is independent of the direction of h, the quantity $A_{\epsilon,\epsilon'}$ is independent of the direction of h which ensures uniform convergence of the directional difference quotients on the unit ball. The limit η_τ^0 satisfies the linear integral equation $\eta_r^0 = h + \int_t^r Du_\tau(\phi_{t\to\tau}(g))[\eta_\tau^0]\mathrm{d}\tau$. This is because $K_\tau^\epsilon[\eta_\tau^\epsilon] \to K_\tau^0[\eta_\tau^0] = Du_\tau(\phi_{t\to\tau}(g))[\eta_\tau^0]$ and $\|K_\tau^\epsilon[\eta_\tau^\epsilon]\| \le M_R(\tau)\|h\|e^{\int_t^r M_R(\xi)\mathrm{d}\xi}$ with $M_R \in L^1(0,1)$, and then the Dominated Convergence Theorem allows us to pass the limit in Equation 26 and obtain the linear integral equation of η_τ^0. This integral equation follows directly that η_r^0 is linear in h. By applying Grönwall's inequality, we can easily further obtain a uniform bound on $\|\eta_r^0\|$, implying that η_r^0 defines a bounded linear operator in h. Hence, $\phi_{t\to r}(g)$ is in fact Fréchet differentiable. Define the Fréchet derivative as $\eta_r^0 = D\phi_{t\to r}(g)$, and then with $J_\tau(g) = D\phi_{t\to\tau}(g)$ we have

$$J_r(g)[h] = h + \int_t^r Du_\tau\big(\phi_{t\to\tau}(g)\big)[J_\tau(g)[h]]\mathrm{d}\tau. \tag{32}$$

Applying Grönwall's inequality to the above, we obtain that for all $\forall\tau \in [t,r]$, $\|J_\tau(g)\| \le e^{\int_t^\tau M_R(\xi)\mathrm{d}\xi}$. Hence, $Du_\tau\big(\phi_{t\to\tau}(g)\big)[J_\tau(g)[h]]$ is integrable over $[t,r]$, which

implies that $J_r(g)[h]$ is absolutely continuous with respect to r. Therefore, we have:

$$\frac{\mathrm{d}}{\mathrm{d}r}J_r(g) = Du_r\big(\phi_{t\to r}(g)\big)\circ J_r(g), \quad J_t(g) = \mathrm{Id}_{\mathcal{F}}. \tag{33}$$

For any $g \in \tilde{\mathcal{B}}_R^{t,r}$, since $\phi_{t\to r}$ is Lipschitz continuous, there exists a neighborhood of g, denoted by $B_{g,\epsilon} = \{h \in \mathcal{F} | \|g-h\| < \epsilon\}$, such that $B_{g,\epsilon} \subset \tilde{\mathcal{B}}_R^{t,r}$. For any $g' \in B_{g,\epsilon}$, using the integral equation above, we have:

$$\begin{aligned}
&J_r(g) - J_r(g') \\
&= \int_t^r Du_\tau\big(\phi_{t\to\tau}(g)\big)\circ J_\tau(g)\mathrm{d}\tau \\
&- \int_t^r Du_\tau\big(\phi_{t\to\tau}(g')\big)\circ J_\tau(g')\mathrm{d}\tau \\
&= \int_t^r Du_\tau\big(\phi_{t\to\tau}(g)\big)\circ (J_\tau(g) - J_\tau(g')) \\
&+ (Du_\tau\big(\phi_{t\to\tau}(g)\big) - Du_\tau\big(\phi_{t\to\tau}(g')\big))\circ J_\tau(g')\mathrm{d}\tau.
\end{aligned} \tag{34}$$

Taking norms on both sides of the above equation, we have

$$\begin{aligned}
&\|J_r(g) - J_r(g')\| \\
&\le \int_t^r \|Du_\tau\big(\phi_{t\to\tau}(g)\big)\|\|J_\tau(g) - J_\tau(g')\|\mathrm{d}\tau \\
&+ \int_t^r \|Du_\tau\big(\phi_{t\to\tau}(g)\big) - Du_\tau\big(\phi_{t\to\tau}(g')\big)\|\|J_\tau(g')\|\mathrm{d}\tau.
\end{aligned} \tag{35}$$

Set $\epsilon_{g'}(r) = \int_t^r \|Du_\tau\big(\phi_{t\to\tau}(g)\big) - Du_\tau\big(\phi_{t\to\tau}(g')\big)\|\|J_\tau(g')\|\mathrm{d}\tau$. Since both Du_τ and $\phi_{t\to\tau}$ continuous, we have $\|Du_\tau\big(\phi_{t\to\tau}(g)\big) - Du_\tau\big(\phi_{t\to\tau}(g')\big)\| \to 0$, as $g' \to 0$. Moreover, because$\|Du_\tau\big(\phi_{t\to\tau}(g)\big) - Du_\tau\big(\phi_{t\to\tau}(g')\big)\|\|J_\tau(g')\| \le 2M_R(\tau)e^{\int_t^\tau M_R(\xi)\mathrm{d}\xi}$ by the Dominated Convergence Theorem, we have $\epsilon_{g'}(r) \to 0$ as $g' \to g$. Meanwhile, $\|J_r(g) - J_r(g')\|$ satisfies:

$$\begin{aligned}
&\|J_r(g) - J_r(g')\| \\
&\le \int_t^r \|Du_\tau\big(\phi_{t\to\tau}(g)\big)\|\|J_\tau(g) - J_\tau(g')\|\mathrm{d}\tau \\
&+ \int_t^r \|Du_\tau\big(\phi_{t\to\tau}(g)\big) - Du_\tau\big(\phi_{t\to\tau}(g')\big)\|\|J_\tau(g')\|\mathrm{d}\tau \\
&\le \int_t^r M_R(\tau)\|J_\tau(g) - J_\tau(g')\|\mathrm{d}\tau + \epsilon_{g'}(r).
\end{aligned} \tag{36}$$

By applying Grönwall's inequality, we have:

$$\|J_r(g) - J_r(g')\| \le \epsilon_{g'}(r)e^{\int_t^r M_R(\tau)\mathrm{d}\tau}. \tag{37}$$

Since $\epsilon_{g'}(r) \to 0$ as $g' \to g$, it follows that $\|J_r(g) - J_r(g')\| \to 0$. Hence, $J_r(g)$ is continuous at g. □

Since u_t is not directly available, and during training we construct the conditional velocity field u_t^f and represent u_t as expectation of u_t^f as Equation 4, we need to reformulate the assumptions in Theorem B.1 in terms of u_t^f.

Lemma B.2. *Let $\mu_t^f \ll \mu_t$ for ν-a.e. f and almost every $t \in [0,1]$ and thus define $\rho_t^f(g) := \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)$, which is validated by Theorem A.1. Assume that for ν-a.e. f and a.e. $t \in [0,T]$ and any radius $R > 0$, there exist measurable nonnegative functions $A_{R,f}(t)$, $B_{R,f}(t)$, $C_{R,f}(t)$, $E_{R,f}(t)$, $L_{R,f}(t)$, $M_{R,f}(t)$:*

(B1) *(Continuity) $u_t^f(g)$ and $\rho_t^f(g)$ are measurable in t and uniform Lipschitz in $g \in \mathcal{B}_R$: $\forall x, y \in \mathcal{B}_R$*

$$\begin{aligned}
\|u_t^f(x) - u_t^f(y)\| &\le L_{R,f}(t)\|x-y\|, \\
|\rho_t^f(x) - \rho_t^f(y)| &\le E_{R,f}(t)\|x-y\|.
\end{aligned} \tag{38}$$

(B2) *(C^1 in g with bounds) $u_t^f \in C^1(\mathcal{F};\mathcal{F})$ and $\rho_t^f \in C^1(\mathcal{F};\mathbb{R})$ are continuously Fréchet differentiable and $\forall g \in \mathcal{B}_R$*

$$\|Du_t^f(g)\| \le M_{R,f}(t), \qquad \|D\rho_t^f(g)\| \le A_{R,f}(t). \tag{39}$$

(B3) *(Integrable envelopes) $\forall g \in \mathcal{B}_R$, $\|u_t^f(g)\| \le B_{R,f}(t) \in L^1(0,1)$ and $0 \le \rho_t^f(g) \le C_{R,f}(t)$ are bounded and $L_R(t) \in L^1(0,1)$ and $M_R(t) \in L^1(0,1)$ and $U_R(t) \in L^1(0,1)$ are well-defined and integrable, which are defined as*

$$\begin{aligned}
L_R(t) &:= \int_{\mathcal{F}} \big(L_{R,f}(t)C_{R,f}(t) \\
&\qquad + E_{R,f}(t)B_{R,f}(t)\big)\mathrm{d}\nu(f), \\
M_R(t) &:= \int_{\mathcal{F}} \big(M_{R,f}(t)C_{R,f}(t) \\
&\qquad + A_{R,f}(t)B_{R,f}(t)\big)\mathrm{d}\nu(f), \\
U_R(t) &:= \int_{\mathcal{F}} B_{R,f}(t)C_{R,f}(t)\mathrm{d}\nu(f).
\end{aligned} \tag{40}$$

Then the marginal field u_t satisfies the hypotheses (A1)–(A2) *as*

$$\begin{aligned}
\|u_t(x) - u_t(y)\| \le L_R(t)\|x-y\|, \forall x, y \in \mathcal{B}_R, \\
\|Du_t(g)\| \le M_R(t), \forall g \in \mathcal{B}_R.
\end{aligned}$$

Proof. For any $x, y \in \mathcal{B}_R$, compute $u_t(x) - u_t(y)$.

$$\begin{aligned}
&u_t(x) - u_t(y) \\
&= \int_{\mathcal{F}} u_t^f(x)\rho_t^f(x)\mathrm{d}\nu(f) - \int_{\mathcal{F}} u_t^f(y)\rho_t^f(y)\mathrm{d}\nu(f) \\
&= \int_{\mathcal{F}} [u_t^f(x) - u_t^f(y)]\rho_t^f(x)\mathrm{d}\nu(f) \\
&\quad + \int_{\mathcal{F}} u_t^f(y)[\rho_t^f(x) - \rho_t^f(y)]\mathrm{d}\nu(f).
\end{aligned} \tag{41}$$

Taking norms on both sides gives:

$$\begin{aligned}\|u_t(x)-u_t(y)\| &\le \int_{\mathcal{F}} L_{R,f}(t)\|x-y\|C_{R,f}(t)\mathrm{d}\nu(f)\\ &\quad+\int_{\mathcal{F}} B_{R,f}(t)E_{R,f}(t)\|x-y\|\mathrm{d}\nu(f)\\ &= L_R(t)\|x-y\|.\end{aligned} \tag{42}$$

By (B3) we have $L_R \in L^1(0,1)$. Moreover, since $u_t^f(g)$ and $\rho_t^f(g)$ are measurable and bounded, $u_t(g)$ is measurable in t. Since $\|u_t^f(g)\| \le B_{R,f}(t)$ and $\|u_t(g)\| \le \int_{\mathcal{F}} \|u_t^f(g)\|\rho_t^f \mathrm{d}\nu(f) \le \int_{\mathcal{F}} B_{R,f}(t)C_{R,f}(t)\mathrm{d}\nu(f) = U_R(t) \in L^1(0,1)$, we have that $u_t(g)$ is integrable with respect to t. Hence, (A1) holds.

To prove that u_t is Fréchet differentiable, we explicitly write out $Du_t(g)$ and then show that this $Du_t(g)$ indeed serves as the Fréchet derivative of u_t, thereby establishing the Fréchet differentiability of u_t:

$$Du_t(g) = \int (Du_t^f(g)\rho_t^f(g) + u_t^f(g)\otimes D\rho_t^f(g))\mathrm{d}\nu(f), \tag{43}$$

where $\otimes$ is the tensor product symbol and $u_t^f(g)\otimes D\rho_t^f(g)) : \mathcal{F}\to\mathcal{F}$ is calculated as $(u_t^f(g)\otimes D\rho_t^f(g)))[h] = D\rho_t^f(g)[h]u_t^f(g)$.

Since $u_t^f\rho_t^f$ is Fréchet differentiable, the difference $u_t^f(g+h)\rho_t^f(g+h) - u_t^f(g)\rho_t^f(g)$ can be written in integral form, and by the product rule for differentiation, we have:

$$\begin{aligned}&u_t^f(g+h)\rho_t^f(g+h) - u_t^f(g)\rho_t^f(g)\\ &= \int_0^1 D(u_t^f\rho_t^f)(g+\theta h)[h]\mathrm{d}\theta\\ &= \int_0^1 (Du_t^f(g+\theta h)\rho_t^f(g+\theta h)\\ &\qquad + u_t^f(g+\theta h)\otimes D\rho_t^f(g+\theta h))[h]\mathrm{d}\theta.\end{aligned} \tag{44}$$

Set the difference $R_f(h) = u_t^f(g+h)\rho_t^f(g+h) - u_t^f(g)\rho_t^f(g) - (Du_t^f(g)\rho_t^f(g) + u_t^f(g)\otimes D\rho_t^f(g))[h]$, which could be used to represent the difference $u_t^f(g+h)\rho_t^f(g+h) - u_t^f(g)\rho_t^f(g) - Du_t(g)[h]$ as $u_t^f(g+h)\rho_t^f(g+h) - u_t^f(g)\rho_t^f(g) - Du_t(g)[h] = \int_{\mathcal{F}} R_f(h)\mathrm{d}\nu(f)$. Here, $\|h\|$ is small enough to ensure $\|g+h\| < R$. Then we calculate $R_f(h)$ as

$$\begin{aligned}&R_f(h)\\ &= \int_0^1 (Du_t^f(g+\theta h)\rho_t^f(g+\theta h)\\ &\qquad + u_t^f(g+\theta h)\otimes D\rho_t^f(g+\theta h))[h]\mathrm{d}\theta\\ &- \int_0^1 (Du_t^f(g)\rho_t^f(g) + u_t^f(g)\otimes D\rho_t^f(g))[h]\mathrm{d}\theta\\ &= \int_0^1 ((Du_t^f(g+\theta h) - Du_t^f(g))\rho_t^f(g+\theta h)\\ &+ Du_t^f(g)(\rho_t^f(g+\theta h) - \rho_t^f(g))\\ &+ (u_t^f(g+\theta h) - u_t^f(g))\otimes D\rho_t^f(g+\theta h)\\ &+ u_t^f(g)\otimes(D\rho_t^f(g+\theta h) - D\rho_t^f(g)))[h]\mathrm{d}\theta.\end{aligned} \tag{45}$$

Taking norms on both sides gives:

$$\begin{aligned}&\|R_f(h)\|\\ &\le \int_0^1 (\|Du_t^f(g+\theta h) - Du_t^f(g)\|\|\rho_t^f(g+\theta h)\|\\ &+ \|Du_t^f(g)\|\|\rho_t^f(g+\theta h) - \rho_t^f(g)\|\\ &+ \|u_t^f(g+\theta h) - u_t^f(g)\|\|D\rho_t^f(g+\theta h)\|\\ &+ \|u_t^f(g)\|\|D\rho_t^f(g+\theta h) - D\rho_t^f(g))\|\|h\|\mathrm{d}\theta\\ &\le \int_0^1 4M_{R,f}(t)C_{R,f}(t) + 4A_{R,f}(t)B_{R,f}(t)\mathrm{d}\theta\\ &= [4M_{R,f}(t)C_{R,f}(t) + 4A_{R,f}(t)B_{R,f}(t)]\|h\|.\end{aligned} \tag{46}$$

Thus, since $M_R(t)$ is well defined, it follows that $4M_{R,f}(t)C_{R,f}(t) + 4A_{R,f}(t)B_{R,f}(t)$ is integrable with respect to $\mathrm{d}\nu(f)$. Moreover, because $u_t^f\rho_t^f$ is Fréchet differentiable, we have $\frac{\|R_f(h)\|}{\|h\|}\to 0$. Hence, by the Dominated Convergence Theorem, $\int_{\mathcal{F}} \frac{\|R_f(h)\|}{\|h\|}\mathrm{d}\nu(f)\to 0$, which implies that $\int_{\mathcal{F}} R_f(h)\mathrm{d}\nu(f) = o(\|h\|)$, that is,

$$\begin{aligned}&\int_{\mathcal{F}} R_f(h)\mathrm{d}\nu(f)\\ &= \int_{\mathcal{F}} u_t^f(g+h)\rho_t^f(g+h) - u_t^f(g)\rho_t^f(g)\\ &- (Du_t^f(g)\rho_t^f(g) + u_t^f(g)\otimes D\rho_t^f(g))[h]\mathrm{d}\nu(f)\\ &= u_t(g+h) - u_t(g) - Du_t(g)[h] = o(\|h\|).\end{aligned} \tag{47}$$

Hence, $Du_t(g)$ is the Fréchet derivative of $u_t(g)$, and thus u_t is Fréchet differentiable.

Taking norms on both sides of Equation 43 gives:

$$\begin{aligned}\|Du_t(g)\| &\le \int_{\mathcal{F}} (\|Du_t^f(g)\|\|\rho_t^f(g)\| + \|u_t^f(g)\|\|D\rho_t^f(g))\|\mathrm{d}\nu(f)\\ &\le \int_{\mathcal{F}} M_{R,f}(t)C_{R,f}(t) + B_{R,f}(t)A_{R,f}(t)\mathrm{d}\nu(f).\\ &= M_R(t)\end{aligned} \tag{48}$$

Since $M_R(t)$ is well-defined and integrable, condition (A2) holds. □

Following the same reasoning as in [27], our next step is to incorporate the specific choices of the conditional path and conditional velocity in Equation 5, in order to translate the assumptions on u_t^f and ρ_t^f in Theorem B.2 into the corresponding assumptions that the dataset must satisfy.

Lemma B.3. *We choose the conditional measures and conditional velocity in Equation 5 as*

$$\begin{aligned}
&\mu_t^f = \mathcal{N}(m_t^f, (\sigma_t^f)^2 C_0),\\
&m_t^f = tf, \sigma_t^f = 1-(1-\sigma_{\min})t,\\
&u_t^f(g) = \frac{\dot\sigma_t^f}{\sigma_t^f}(g-m_t^f)+\dot m_t^f = I_t g + J_t f,\\
&I_t = -\frac{1-\sigma_{\min}}{1-(1-\sigma_{\min})t}, J_t = 1-tI_t,
\end{aligned} \tag{49}$$

where $\sigma_{\min} > 0$. Assume that

(C1) *(Finite Second Moment) The data distribution ν satisfies $\int_{\mathcal{F}} \|f\|^2 \mathrm{d}\nu(f) < \infty$.*

Then the assumptions (B1)–(B3) in Theorem B.2 holds.

Proof. First, we compute $\rho_t^f(g)$, where $\rho_t^f(g) = \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}$ denotes the Radon–Nikodym derivative between measures μ_t^f and μ_t. In infinite-dimensional spaces, since there is no Lebesgue measure, the Radon–Nikodym derivative must be taken with respect to a reference measure. We choose the reference measure $\rho_0^f = \rho_0 = \mathcal{N}(0, C_0)$ and first compute the Radon–Nikodym derivatives $\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_0}$ and $\frac{\mathrm{d}\mu_t}{\mathrm{d}\mu_0}$ with respect to this reference measure. Through Cameron–Martin theorem, $\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_0}$ and $\frac{\mathrm{d}\mu_t}{\mathrm{d}\mu_0}$ can be calculated as

$$\begin{aligned}
\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_0} &= e^{\left(\left\langle \frac{m_t^f}{\sigma_t^2}, g\right\rangle - \frac{1}{2}\left\|\frac{m_t^f}{\sigma_t}\right\|^2\right)},\\
\frac{\mathrm{d}\mu_t}{\mathrm{d}\mu_0} &= \int_{\mathcal{F}} e^{\left(\left\langle \frac{m_t^{f'}}{\sigma_t^2}, g\right\rangle - \frac{1}{2}\left\|\frac{m_t^{f'}}{\sigma_t}\right\|^2\right)} \mathrm{d}\nu(f').
\end{aligned} \tag{50}$$

By the Radon–Nikodym ratio formula, we have:

$$\begin{aligned}
\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t} &= \frac{\mathrm{d}\mu_t^f/\mathrm{d}\mu_0}{\mathrm{d}\mu_t/\mathrm{d}\mu_0}\\
&= \frac{e^{(\langle a_t^f, g\rangle - b_t^f)}}{\int_{\mathcal{F}} e^{(\langle a_t^{f'}, g\rangle - b_t^{f'})}\mathrm{d}\nu(f')},
\end{aligned} \tag{51}$$

where $a_t^f = \frac{m_t^f}{\sigma_t^2}$ and $b_t^f = \frac{1}{2}\left\|\frac{m_t^f}{\sigma_t}\right\|^2 = \frac{1}{2}\|a_t^f\|^2$.

Let $s_t^f(g) = e^{(\langle a_t^f, g\rangle - b_t^f)}$ and $Z_t(g) = \int_{\mathcal{F}} s_t^{f'}(g)\mathrm{d}\nu(f')$. We then compute $D\rho_t^f(g) = D\left(\frac{s_t^f(g)}{Z_t(g)}\right)$. First, $Ds_t^f(g)[h] = s_t^f(g)\langle a_t^f, h\rangle$, while $DZ_t(g)[h]$ is given by

$$\begin{aligned}
DZ_t(g)[h] &= \int_{\mathcal{F}} Ds_t^{f'}(g)[h]\nu(df')\\
&= \int_{\mathcal{F}} s_t^{f'}(g)\langle a_t^{f'}, h\rangle \nu(df')\\
&= Z_t(g)\langle \bar a_t(g), h\rangle,
\end{aligned} \tag{52}$$

where $\bar a_t(g) = \int_{\mathcal{F}} a_t^{f'}\rho_t^{f'}(g)\mathrm{d}\nu(f')$ is defined as the $\rho_t^{f'}(g)$-weighted average of $a_t^{f'}$ with respect to ν. Then $D\rho_t^f(g)[h]$ can be calculated as

$$\begin{aligned}
D\rho_t^f(g)[h] &= \frac{Ds_t^f(g)[h]Z_t(g) - s_t^f(g)DZ_t(g)[h]}{Z_t(g)^2}\\
&= \frac{s_t^f(g)}{Z_t(g)}(\langle a_t^f, h\rangle - \langle \bar a_t, h\rangle)\\
&= \rho_t^f(g)\langle a_t^f - \bar a_t(g), h\rangle,
\end{aligned} \tag{53}$$

Next, we estimate the bounds for ρ_t^f and $D\rho_t^f(g)$. Since $a_t^f = \frac{m_t^f}{(\sigma_t^f)^2} = \frac{t}{(\sigma_t^f)^2}f$ is bounded by $|a_t^f| \le \sigma_{\min}^{-2}|f|$ and the inequality $Rr - \frac{1}{2}r^2 \le \frac{1}{2}R^2$ holds for all $r > 0$, we have

$$\begin{aligned}
s_t^f(g) &= e^{(\langle a_t^f, g\rangle - b_t^f)}\\
&\le e^{(R\|a_t^f\| - \frac{1}{2}\|a_t^f\|^2)}\\
&\le e^{\frac{1}{2}R^2}.
\end{aligned} \tag{54}$$

For $Z_t(g)$ we have

$$\begin{aligned}
Z_t(g) &= \int_{\mathcal{F}} s_t^{f'}(g)\mathrm{d}\nu(f')\\
&= \int_{\mathcal{F}} e^{(\langle a_t^{f'}, g\rangle - b_t^{f'})}\mathrm{d}\nu(f')\\
&\ge \int_{\mathcal{F}} e^{(-R\|a_t^{f'}\| - \frac{1}{2}\|a_t^{f'}\|^2)}\mathrm{d}\nu(f').
\end{aligned} \tag{55}$$

Let $c_R(t) = \int_{\mathcal{F}} e^{(-R\|a_t^{f'}\| - \frac{1}{2}\|a_t^{f'}\|^2)}\mathrm{d}\nu(f')$ and $C_{R,f}(t) = \frac{e^{\frac{R^2}{2}}}{c_R(t)}$. Then we have:

$$\rho_t^f(g) \le C_{R,f}(t). \tag{56}$$

Since $\int_{\mathcal{F}} \|f\|^2 \mathrm{d}\nu(f) < \infty$, let $F_1 = \int_{\mathcal{F}} \|f\| \mathrm{d}\nu(f)$ and $F_2 = \int_{\mathcal{F}} \|f\|^2 \mathrm{d}\nu(f)$. Then we have

$$\begin{aligned}
\|D\rho_t^f(g)\| &= \sup_{\|h\|=1} \|D\rho_t^f(g)[h]\|\\
&\le \rho_t^f(g)(\|a_t^f\| + \|\bar a_t(g)\|)\\
&\le \rho_t^f(g)(\sigma_{\min}^{-2}\|f\| + \int_{\mathcal{F}} \|a_t^{f'}\|\rho_t^{f'}(g)\mathrm{d}\nu(f'))\\
&\le C_{R,f}(t)(\sigma_{\min}^{-2}\|f\| + \sigma_{\min}^{-2}C_{R,f}(t)F_1).
\end{aligned} \tag{57}$$

Define the g-independent quantity $A_{R,f}(t)$ as $A_{R,f}(t) = C_{R,f}(t)(\sigma_{\min}^{-2}\|f\| + \sigma_{\min}^{-2}C_{R,f}(t)F_1)$ and hence we have:

$$\|D\rho_t^f(g)\| \le A_{R,f}(t), \tag{58}$$

$\forall x, y \in B_R$, by writing $\rho_t^f(x) - \rho_t^f(y) = \int_0^1 D\rho_t^f(y+\theta(x-y))[x-y]\mathrm{d}\theta$, we have

$$\begin{aligned}\|\rho_t^f(x) - \rho_t^f(y)\| &\le \int_0^1 \|D\rho_t^f(y+\theta(x-y))\|\|x-y\|\mathrm{d}\theta \\ &\le A_{R,f}(t)\|x-y\|.\end{aligned} \tag{59}$$

Thus we can write $E_{R,f}(t) = A_{R,f}(t)$ and have:

$$\|\rho_t^f(x) - \rho_t^f(y)\| \le E_{R,f}(t)\|x-y\|. \tag{60}$$

For the velocity field $u_r^f(g)$, we have:

$$\begin{aligned}\|u_r^f(g)\| &= \|I_t g + J_t f\| \\ &\le \|I_t\|\|g\| + \|J_t\|\|f\| \\ &\le \sigma_{\min}^{-1}R + \|f\|(1+\sigma_{\min}^{-1}).\end{aligned} \tag{61}$$

Thus, by setting $B_{R,f}(t) = \sigma_{\min}^{-1}R + \|f\|(1+\sigma_{\min}^{-1})$, we have:

$$\|u_r^f(g)\| \le B_{R,f}(t). \tag{62}$$

Since $Du_t^f(g)[h] = I_t h$, it follows that $\|Du_t^f(g)\| = |I_t| < \sigma_{\min}^{-1}$. By taking $M_{R,f}(t) = \sigma_{\min}^{-1}$, we have:

$$\|Du_t^f(g)\| \le M_{R,f}(t). \tag{63}$$

Since $u_t^f(x) - u_t^f(y) = I_t(x-y)$, by taking $L_{R,f}(t) = \sigma_{\min}^{-1}$, we have:

$$\begin{aligned}\|u_t^f(x) - u_t^f(y)\| &\le |I_t|\|x-y\| \\ &\le \sigma_{\min}^{-1}\|x-y\| \\ &= L_{R,f}(t)\|x-y\|.\end{aligned} \tag{64}$$

Then we express $L_R(t)$, $M_R(t)$ and $U_R(t)$ respectively.

$$\begin{aligned}&L_R(t) \\ &= \int_{\mathcal{F}} \sigma_{\min}^{-1}\frac{e^{\frac{R^2}{2}}}{c_R(t)} + \frac{e^{\frac{R^2}{2}}}{c_R(t)}(\sigma_{\min}^{-2}\|f\| \\ &+ \sigma_{\min}^{-2}\frac{e^{\frac{R^2}{2}}}{c_R(t)}F_1)(\sigma_{\min}^{-1}R + \|f\|(1+\sigma_{\min}^{-1}))\mathrm{d}\nu(f) \\ &= (\sigma_{\min}^{-1}\frac{e^{\frac{R^2}{2}}}{c_R(t)} + \sigma_{\min}^{-3}\frac{e^{R^2}}{c_R^2(t)}F_1 R) \\ &+ (\sigma_{\min}^{-3}\frac{e^{\frac{R^2}{2}}}{c_R(t)}R + \sigma_{\min}^{-2}(1+\sigma_{\min}^{-1})\frac{e^{\frac{R^2}{2}}}{c_R(t)}F_1)F_1 \\ &+ \sigma_{\min}^{-2}(1+\sigma_{\min}^{-1})\frac{e^{\frac{R^2}{2}}}{c_R(t)}F_2.\end{aligned} \tag{65}$$

Moreover, since $c_R(t)$ has a strictly positive lower bound for all $t \in [0,1]$

$$\begin{aligned}c_R(t) &= \int_{\mathcal{F}} e^{(-R\|a_t^{f'}\| - \frac{1}{2}\|a_t^{f'}\|^2)}\mathrm{d}\nu(f') \\ &= \int_{\mathcal{F}} e^{(-R\|\frac{tf}{\sigma_t^2}\| - \frac{1}{2}\|\frac{tf}{\sigma_t^2}\|^2)}\mathrm{d}\nu(f') \\ &\ge \int_{\mathcal{F}} e^{(-R\|\frac{1}{\sigma_{\min}^2}\|\|f'\| - \frac{1}{2}\|\frac{1}{\sigma_{\min}^2}\|^2\|f'\|^2)}\mathrm{d}\nu(f') \\ &> 0.\end{aligned} \tag{66}$$

Therefore, there exists $M_1 > 0$ such that $L_R(t) < M_1$ for all $t \in [0,1]$, and hence $L_R(t) \in L^1(0,1)$.

And $M_R(t)$ and $U_R(t)$ can be computed similarly and likewise,since $c_R(t)$ has a strictly positive lower bound for all $t \in [0,1]$, there exists $M_2 > 0$ such that $M_R(t) < M_2$ for all $t \in [0,1]$ and there exists $U_2 > 0$ such that $U_R(t) < U_2$ for all $t \in [0,1]$, and thus $M_R(t) \in L^1(0,1)$ and $U_R(t) \in L^1(0,1)$. □

B.3. Proof of Theorem 3.1

Theorem 3.1 (Initial-Time Derivative of Two-Parameter Flow) Assume that the dataset measure ν satisfies $\int_{\mathcal{F}} |f|^2\mathrm{d}\nu(f) < \infty$, and the conditions of Functional Flow Matching [27] hold. With the conditional flow and conditional velocity chosen in Equation 5, the corresponding marginal 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 $0 < t < r < 1$

$$\frac{\partial}{\partial t}\phi_{t\to r}(g) = -D\phi_{t\to r}(g)[u_t(g)], \tag{67}$$

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.

Proof. Based on Lemmas B.1, B.2, and B.3, we know that for any $R > 0$, if $g \in B_R$ and $\phi_{t\to\tau}(g) \in B_R$ for all $\tau \in [t,r]$, then $\phi_{t\to\tau}$ is Fréchet differentiable at g, $\phi_{t\to\tau}$ is continous at t and $D\phi_{t\to\tau}$ is continous at g. Moreover, by Lemmas B.2, since u_t is bounded, for any $g \in \mathcal{F}$ we can always choose some $R > 0$ such that $\phi_{t\to\tau}(g) \in B_R$ for all $\tau \in [t,r]$, which ensures that $\phi_{t\to\tau}$ is differentiable at g, $\phi_{t\to\tau}$ is differentiable at t and $D\phi_{t\to\tau}$ is continous at g.

Since $\phi_{t\to r}(g)$ is continuous at t, for sufficiently small $\epsilon > 0$ we have $\phi_{t-\epsilon\to r}(g) \in B_R$. We then compute the left difference quotient of $\phi_{t\to r}(g)$ with respect to time t as:

$$\begin{aligned}&\frac{\phi_{t\to r}(g) - \phi_{t-h\to r}(g)}{h} \\ &= \frac{\phi_{t\to r}(g) - \phi_{t\to r}(\phi_{t-\epsilon\to t}(g))}{h} \\ &= \frac{D\phi_{t\to r}(g)[g - \phi_{t-\epsilon\to t}(g)] + o(\|g - \phi_{t-\epsilon\to t}(g)\|)}{h}.\end{aligned} \tag{68}$$

Let $\delta_h^-(\tau) = \phi_{t-\epsilon\to\tau}(g) - g$, $\tau \in [t-\epsilon, t]$. Since $\phi_{t-\epsilon\to\tau}(g) - g = \int_{t-h}^{\tau} u_\xi(\phi_{t-h\to\xi}(g))\mathrm{d}\xi$, we have:

$$\begin{aligned}\frac{\delta_h^-(\tau)}{h} &= \frac{1}{h}\int_{t-h}^{\tau} u_\xi(\phi_{t-h\to\xi}(g))\mathrm{d}\xi \\ &= \frac{1}{h}\int_{t-h}^{\tau} u_\xi(\phi_{t-h\to\xi}(g)) - u_\xi(g)\mathrm{d}\xi \\ &\quad + \frac{1}{h}\int_{t-h}^{\tau} u_\xi(g)\mathrm{d}\xi.\end{aligned} \tag{69}$$

We want to determine the limit of $\frac{\delta_h^-(t)}{h}$ as $h \to 0$. Let $\frac{\delta_h^-(t)}{h} = R_h(t) + E_h(t)$, where $E_h(\tau) = \frac{1}{h}\int_{t-h}^{\tau} u_\xi(g)\mathrm{d}\xi$ and $R_h(\tau) = \frac{1}{h}\int_{t-h}^{\tau} u_\xi(\phi_{t-h\to\xi}(g)) - u_\xi(g)\mathrm{d}\xi$. Based on Lemmas B.2 and B.3, $u_t(g)$ is integrable and measurable. Thus based on Lebesgue differentiation theorem, we have $\|\frac{\int_{t-h}^{t} u_\xi(g)\mathrm{d}\xi}{h} - u_t(g)\| \to 0$ a.e. on t, which means $E_h(t) \to u_t(g)$ a.e. on t. Next we prove $R_h(t) \to 0$.

Based on Lemmas B.1, B.2, and B.3, u_ξ is Lipschitz continuous $|u_\xi(\phi_{t-h\to\xi}(g)) - u_\xi(g)| \le L_R(\xi)|\phi_{t-h\to\xi}(g) - g|$ and $L_R \in L^1(0,1)$. Therefore we have

$$\begin{aligned}&R_h(\tau) \\ &= \|\int_{t-h}^{\tau} u_\xi(\phi_{t-h\to\xi}(g)) - u_\tau(g)\mathrm{d}\xi\| \\ &\le \frac{1}{h}\int_{t-h}^{\tau} \|u_\xi(\phi_{t-h\to\xi}(g)) - u_\xi(g)\|\mathrm{d}\xi \\ &\le \int_{t-h}^{\tau} L_R(\xi)\frac{\|\phi_{t-h\to\xi}(g) - g\|}{h}\mathrm{d}\xi \\ &= \int_{t-h}^{\tau} L_R(\xi)\frac{\|\delta_h^-(\xi)\|}{h}\mathrm{d}\xi.\end{aligned} \tag{70}$$

Combined with Equation 69, we have inequality for $\|\frac{\delta_h^-(\tau)}{h}\|$:

$$\begin{aligned}&\|\frac{\delta_h^-(\tau)}{h}\| \\ &\le \int_{t-h}^{\tau} L_R(\xi)\frac{\|\delta_{h,\xi}^-\|}{h}\mathrm{d}\xi + \|E_h(\tau)\|.\end{aligned} \tag{71}$$

$\|E_h(\tau)\|$ is bounded arount $h = 0$ for convergence. By Grönwall's inequality we have

$$\|\frac{\delta_h^-(\tau)}{h}\| \le \|E_h(\tau)\|e^{\int_{t-h}^{\tau} L_R(\xi)\mathrm{d}\xi}. \tag{72}$$

Thus $R_h(t)$ can be calculated as

$$\begin{aligned}R_h(t) &\le \int_{t-h}^{t} L_R(\tau)\frac{\|\delta_h^-(\tau)\|}{h}d\tau \\ &\le \int_{t-h}^{t} L_R(\tau)\|E_h(\tau)\|e^{\int_{t-h}^{\tau} L_R(\xi)\mathrm{d}\xi}d\tau \\ &\le e^{\int_{t-h}^{t} L_R(\xi)\mathrm{d}\xi}\int_{t-h}^{t} L_R(\tau)\frac{1}{h}\int_{t-h}^{\tau}\|u_\xi(g)\|\mathrm{d}\xi d\tau \\ &\le e^{\int_{t-h}^{t} L_R(\xi)\mathrm{d}\xi}\int_{t-h}^{t} L_R(\tau)d\tau\frac{1}{h}\int_{t-h}^{t}\|u_\xi(g)\|\mathrm{d}\xi.\end{aligned} \tag{73}$$

When $h \to 0$, $\int_{t-h}^{t} L_R(\xi)\mathrm{d}\xi \to 0$ as $L_R(\xi) \in L^1(0,1)$ integrable and $\frac{1}{h}\int_{t-h}^{t}\|u_\xi(g)\|\mathrm{d}\xi \to \|u_t(g)\|$ by Lebesgue differentiation theorem for $\|u_t(g)\|$ is measurable and integrable on t. Therefore, we have $R_h(t) \to 0$.

Now we have $\frac{\phi_{t\to r}(g) - \phi_{t-h\to r}(g)}{h} = -D\phi_{t\to r}(g)[u_t(g)] + o(u_t(g))$. Similarly, we can also prove that $\frac{\phi_{t+h\to r}(g) - \phi_{t\to r}(g)}{h} = -D\phi_{t\to r}(g)[u_t(g)] + o(u_t(g))$. Therefore $\phi_{t\to r}(g)$ is differentiable with respect to t and $\frac{\partial}{\partial t}\phi_{t\to r}(g) = -D\phi_{t\to r}(g)[u_t(g)]$. □

B.4. Proof of Theorem 3.2

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.

Proof. First, since we are working in a real Hilbert space, for any $f, g \in \mathcal{F}$, we have

$$\begin{aligned}&\|\bar{u}_{t\to r}(g) - \bar{u}_{t\to r}^\theta(g)\|^2 \\ &= \langle \bar{u}_{t\to r}(g) - \bar{u}_{t\to r}^\theta(g), \bar{u}_{t\to r}(g) - \bar{u}_{t\to r}^\theta(g)\rangle \\ &= \|\bar{u}_{t\to r}(g)\|^2 + \|\bar{u}_{t\to r}^\theta(g)\|^2 - 2\langle \bar{u}_{t\to r}(g), \bar{u}_{t\to r}^\theta(g)\rangle,\end{aligned} \tag{74}$$

and similarly

$$\begin{aligned}&||(r-t)(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) \\ &+ D\bar{u}_{t\to r}(g)[u_t^f(g)]) + u_t^f(g) - \bar{u}_{t\to r}^\theta(g)||^2 \\ &= \|(r-t)(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\bar{u}_{t\to r}(g)[u_t^f(g)]) + u_t^f(g)\|^2 \\ &+ \|\bar{u}_{t\to r}^\theta(g)\|^2 - 2\langle (r-t)(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) \\ &\qquad + D\bar{u}_{t\to r}(g)[u_t^f(g)]) + u_t^f(g), \bar{u}_{t\to r}^\theta(g)\rangle.\end{aligned} \tag{75}$$

Note that the first term in both expressions is independent of the model parameters, so we focus on analyzing the remaining two terms. First, we show that the second term in both expressions is identical, i.e., $\mathbb{E}_{t,r,g\sim\mu_t}[\|\bar{u}_{t\to r}^\theta(g)\|^2] =$

$\mathbb{E}_{t,r,g\sim\mu_t^f,f\sim\mu_1}[\|\bar{u}_{t\to r}^{\theta}(g)\|^2]$

$$\begin{aligned}
&\mathbb{E}_{t,r,g\sim\mu_t}[\|\bar{u}_{t\to r}^{\theta}(g)\|^2] \\
&= \int_0^1\int_0^1\int_g \|\bar{u}_{t\to r}^{\theta}(g)\|^2 \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{①}{=} \int_0^1\int_0^1\int_f\int_g \|\bar{u}_{t\to r}^{\theta}(g)\|^2 \mathrm{d}\mu_t^f(g)\mathrm{d}\nu(f)\mathrm{d}t\mathrm{d}r \\
&= \mathbb{E}_{t,r,g\sim\mu_t^f,f\sim\mu_1}[\|\bar{u}_{t\to r}^{\theta}(g)\|^2],
\end{aligned} \tag{76}$$

where ① follows from the relationship between μ_t and μ_t^f given in Equation 4.

Then, we show that the third term in both expressions is identical, i.e., $\langle \bar{u}_{t\to r}(g), \bar{u}_{t\to r}^{\theta}(g)\rangle = \langle (r-t)(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\bar{u}_{t\to r}(g)[u_t^f(g)]) + u_t^f(g), \bar{u}_{t\to r}^{\theta}(g)\rangle$

$$\begin{aligned}
&\mathbb{E}_{t,r,g\sim\mu_t}[\langle \bar{u}_{t\to r}(g), \bar{u}_{t\to r}^{\theta}(g)\rangle] \\
&= \int_0^1\int_0^1\int_g \langle \bar{u}_{t\to r}(g), \bar{u}_{t\to r}^{\theta}(g)\rangle \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{①}{=} \int_0^1\int_0^1\int_g \langle (r-t)\big(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\bar{u}_{t\to r}(g)[u_t(g)]\big) \\
&\quad + u_t(g), \bar{u}_{t\to r}^{\theta}(g)\rangle \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{②}{=} \int_0^1\int_0^1\int_g \langle (r-t)\big(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) \\
&\quad + D\bar{u}_{t\to r}(g)[\int_{\mathcal{F}} u_t^f(g)\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)\mathrm{d}\nu(f)]\big) \\
&\quad + \int_{\mathcal{F}} u_t^f(g)\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)\mathrm{d}\nu(f), \bar{u}_{t\to r}^{\theta}(g)\rangle \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{③}{=} \int_0^1\int_0^1\int_{\mathcal{F}}\int_g \langle (r-t)\big(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\bar{u}_{t\to r}(g)[u_t^f(g)]\big) \\
&\quad + u_t^f(g), \bar{u}_{t\to r}^{\theta}(g)\rangle \frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}(g)\mathrm{d}\mu_t(g)\mathrm{d}\nu(f)\mathrm{d}t\mathrm{d}r \\
&= \int_0^1\int_0^1\int_{\mathcal{F}}\int_g \langle (r-t)\big(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\bar{u}_{t\to r}(g)[u_t^f(g)]\big) \\
&\quad + u_t^f(g), \bar{u}_{t\to r}^{\theta}(g)\rangle \mathrm{d}\mu_t^f(g))\mathrm{d}\nu(f)\mathrm{d}t\mathrm{d}r \\
&= \mathbb{E}_{t,r,g\sim\mu_t^f,f\sim\mu_1}[\langle (r-t)(\frac{\partial}{\partial t}\bar{u}_{t\to r}(g) + D\bar{u}_{t\to r}(g)[u_t^f(g)]) \\
&\quad + u_t^f(g), \bar{u}_{t\to r}^{\theta}(g)\rangle],
\end{aligned} \tag{77}$$

where ① follows by substituting Equation 9, ② applies the relationship between u_t and u_t^f given in Equation 4, and ③ uses the exchangeability between Bochner integrals and inner products, together with the Fubini–Tonelli theorem. Therefore, we have $\mathcal{L}_c^M(\theta) = \mathcal{L}^M(\theta) + C$ where C is independent of the model parameters θ. □

B.5. Proof of Theorem 3.3 and Derivation of Equation 13 and Equation 14

We first present the derivation of Equation 13. Given the relationship between u_t and $\hat{f}_{1,t}(g)$ as $\hat{f}_{1,t}(g) = (1-t)u_t(g) + g$, together with the relationship between $u_t(g)$ and $u_t^f(g)$ in Equation 4, we have:

$$\begin{aligned}
\hat{f}_{1,t}(g) &= (1-t)u_t(g) + g \\
&= (1-t)\int_{\mathcal{F}} u_t^f(g)\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}\mathrm{d}\nu(f) + g \\
&= \int_{\mathcal{F}} ((1-t)u_t^f(g) + g)\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}\mathrm{d}\nu(f).
\end{aligned} \tag{78}$$

Similar to $u_{t\to r}^f(g)$, $\hat{f}_{1,t\to r}^f(g)$ is defined through

$$\hat{f}_{1,t}(g) = \int_{\mathcal{F}} \hat{f}_{1,t}^f(g)\frac{\mathrm{d}\mu_t^f}{\mathrm{d}\mu_t}\mathrm{d}\nu(f), \tag{79}$$

so we take $\hat{f}_{1,t}^f(g) = (1-t)u_t^f(g) + g$. Substituting the expression of $u_{t\to r}^f(g)$ from Equation 5 gives:

$$\begin{aligned}
\hat{f}_{1,t}(g) &= (1-t)u_t^f(g) + g \\
&= (1-t)\frac{1-\sigma_{\min}}{1-(1-\sigma_{\min})t}(tf-g) + (1-t)f + g \\
&= (\frac{(1-t)(1-\sigma_{\min})}{1-(1-\sigma_{\min})t} - 1)(tf-g) + f \\
&= f - \frac{\sigma_{\min}}{1-(1-\sigma_{\min})t}(tf-g).
\end{aligned} \tag{80}$$

Then we present the derivation of Equation 14. Since $\bar{u}_{t\to r}$ satisfies Equation 9, substituting the relationship between $\bar{u}_{t\to r}$ and $\hat{f}_{1,t\to r}$ given in Equation 11 yields:

$$\begin{aligned}
\frac{\hat{f}_{1,t\to r}(g) - g}{1-t} = (r-t)\big(&\frac{\partial}{\partial t}\frac{\hat{f}_{1,t\to r}(g) - g}{1-t} \\
&+ D\frac{\hat{f}_{1,t\to r}(g) - g}{1-t}[u_t(g)]\big) + u_t(g),
\end{aligned} \tag{81}$$

which gives:

$$
\begin{aligned}
&\hat{f}_{1,t\to r}(g) \\
&= (1-t)((r-t)\big(\frac{\partial}{\partial t}\frac{\hat{f}_{1,t\to r}(g)-g}{1-t} \\
&+ D\frac{\hat{f}_{1,t\to r}(g)-g}{1-t}[u_t(g)]\big) + u_t(g)) + g \\
&= (1-t)((r-t)\big(\frac{(1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) + (\hat{f}_{1,t\to r}(g)-g)}{(1-t)^2} \\
&+ \frac{D\hat{f}_{1,t\to r}(g)[u_t(g)]-u_t(g)}{1-t}\big) + u_t(g)) + g \\
&= (r-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) + \frac{r-t}{1-t}(\hat{f}_{1,t\to r}(g)-g) \\
&+ (r-t)D\hat{f}_{1,t\to r}(g)[u_t(g)] + (1-r)u_t(g) + g \\
&\overset{①}{=} (r-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) + \frac{r-t}{1-t}(\hat{f}_{1,t\to r}(g)-g) \\
&+ (r-t)D\hat{f}_{1,t\to r}(g)[\frac{\hat{f}_{1,t}(g)-g}{1-t}] + (1-r)\frac{\hat{f}_{1,t}(g)-g}{1-t} + g \\
&= \frac{r-t}{1-t}((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) + D\hat{f}_{1,t\to r}(g)[\hat{f}_{1,t}(g)-g]) \\
&+ \frac{1-r}{1-t}\hat{f}_{1,t}(g) + \frac{r-t}{1-t}\hat{f}_{1,t\to r}(g),
\end{aligned}
\tag{82}
$$

where ① is obtained by substituting the relationship between the flow-matching x_1-prediction target $\hat{f}_{1,t}(g)$ and $u_t(g)$, given by $\hat{f}_{1,t}(g) = (1-t)u_t(g)+g$. Moving the term $\frac{r-t}{1-t}\hat{f}_{1,t\to r}(g)$ on the right-hand side of the above equation to the left-hand side and simplifying, we obtain:

$$
\begin{aligned}
\hat{f}_{1,t\to r}(g) = \frac{r-t}{1-t}((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) \\
+ D\hat{f}_{1,t\to r}(g)[\hat{f}_{1,t}(g)-g]) + \hat{f}_{1,t}(g).
\end{aligned}
\tag{83}
$$

Finally, we prove the Theorem 3.3

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

Proof. The proof of this theorem proceeds similarly to Theorem 3.2. First, for any $f, g \in \mathcal{F}$, expand the norm.

$$
\begin{aligned}
&\|\hat{f}_{1,t\to r}(g) - \hat{f}^{\theta}_{1,t\to r}(g)\|^2 \\
&= \langle \hat{f}_{1,t\to r}(g) - \hat{f}^{\theta}_{1,t\to r}(g), \hat{f}_{1,t\to r}(g) - \hat{f}^{\theta}_{1,t\to r}(g)\rangle \\
&= \|\hat{f}_{1,t\to r}(g)\|^2 + \|\hat{f}^{\theta}_{1,t\to r}(g)\|^2 - 2\langle \hat{f}_{1,t\to r}(g), \hat{f}^{\theta}_{1,t\to r}(g)\rangle,
\end{aligned}
\tag{84}
$$

and similarly

$$
\begin{aligned}
&\|\frac{r-t}{1-r}((1-t)\frac{\partial}{\partial_t}\hat{f}_{1,t\to r}(g) \\
&+ D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g)-g]) + \hat{f}^f_{1,t}(g) - \hat{f}^{\theta}_{1,t\to r}(g)\|_2^2 \\
&= \|\frac{r-t}{1-r}((1-t)\frac{\partial}{\partial_t}\hat{f}_{1,t\to r}(g) + D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g)-g]) \\
&+ \hat{f}^f_{1,t}(g)\|^2 + \|\hat{f}^{\theta}_{1,t\to r}(g)\|^2 - 2\langle\frac{r-t}{1-r}((1-t)\frac{\partial}{\partial_t}\hat{f}_{1,t\to r}(g) \\
&+ D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g)-g]) + \hat{f}^f_{1,t}(g), \hat{f}^{\theta}_{1,t\to r}(g)\rangle.
\end{aligned}
\tag{85}
$$

Note that the first term in both expressions is independent of the model parameters, so we focus on analyzing the remaining two terms. First, we show that the second term in both expressions is identical, i.e., $\mathbb{E}_{t,r,g\sim\mu_t}[\|\hat{f}^{\theta}_{1,t\to r}(g)\|^2] = \mathbb{E}_{t,r,g\sim\mu^f_t,f\sim\mu_1}[\|\hat{f}^{\theta}_{1,t\to r}(g)\|^2]$

$$
\begin{aligned}
&\mathbb{E}_{t,r,g\sim\mu_t}[\|\hat{f}^{\theta}_{1,t\to r}(g)\|^2] \\
&= \int_0^1\int_0^1\int_g \|\hat{f}^{\theta}_{1,t\to r}(g)\|^2 \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{①}{=} \int_0^1\int_0^1\int_f\int_g \|\hat{f}^{\theta}_{1,t\to r}(g)\|^2 \mathrm{d}\mu^f_t(g)\mathrm{d}\nu(f)\mathrm{d}t\mathrm{d}r \\
&= \mathbb{E}_{t,r,g\sim\mu^f_t,f\sim\mu_1}[\|\hat{f}^{\theta}_{1,t\to r}(g)\|^2],
\end{aligned}
\tag{86}
$$

where (①) follows from the relationship between μ_t and μ^f_t given in Equation 4.

Then, we show that the third term in both expressions is identical, i.e., $\langle \hat{f}_{1,t\to r}(g), \hat{f}^{\theta}_{1,t\to r}(g)\rangle = \langle\frac{r-t}{1-r}((1-t)\frac{\partial}{\partial_t}\hat{f}_{1,t\to r}(g) + D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g)-g]) +$

$$
\begin{aligned}
&\hat{f}^f_{1,t}(g), \hat{f}^\theta_{1,t\to r}(g)\rangle \\
&\mathbb{E}_{t,r,g\sim\mu_t}[\langle \hat{f}_{1,t\to r}(g), \hat{f}^\theta_{1,t\to r}(g)\rangle] \\
&= \int_0^1\int_0^1\int_g \langle \hat{f}_{1,t\to r}(g), \hat{f}^\theta_{1,t\to r}(g)\rangle \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{①}{=} \int_0^1\int_0^1\int_g \langle \frac{r-t}{1-t}((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) \\
&+ D\hat{f}_{1,t\to r}(g)[\hat{f}_{1,t}(g) - g]) + \hat{f}_{1,t}(g), \hat{f}^\theta_{1,t\to r}(g)\rangle \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{②}{=} \int_0^1\int_0^1\int_g \langle \frac{r-t}{1-t}((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) \\
&+ D\hat{f}_{1,t\to r}(g)[\int_{\mathcal{F}} \hat{f}^f_{1,t}(g)\frac{\mathrm{d}\mu^f_t}{\mathrm{d}\mu_t}\mathrm{d}\nu(f) - g]) \\
&+ \int_{\mathcal{F}} \hat{f}^f_{1,t}(g)\frac{\mathrm{d}\mu^f_t}{\mathrm{d}\mu_t}\mathrm{d}\nu(f), \hat{f}^\theta_{1,t\to r}(g)\rangle \mathrm{d}\mu_t(g)\mathrm{d}t\mathrm{d}r \\
&\overset{③}{=} \int_0^1\int_0^1\int_{\mathcal{F}}\int_g \langle \frac{r-t}{1-t}((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) \\
&+ D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g) - g]) \\
&+ \hat{f}^f_{1,t}(g), \hat{f}^\theta_{1,t\to r}(g)\rangle \frac{\mathrm{d}\mu^f_t}{\mathrm{d}\mu_t}\mathrm{d}\mu_t(g)\mathrm{d}\nu(f)\mathrm{d}t\mathrm{d}r \\
&= \int_0^1\int_0^1\int_{\mathcal{F}}\int_g \langle \frac{r-t}{1-t}((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) \\
&+ D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g) - g]) \\
&+ \hat{f}^f_{1,t}(g), \hat{f}^\theta_{1,t\to r}(g)\rangle \mathrm{d}\mu^f_t(g)\mathrm{d}\nu(f)\mathrm{d}t\mathrm{d}r \\
&= \mathbb{E}_{t,r,g\sim\mu^f_t, f\sim\mu_1}[\langle \frac{r-t}{1-t}((1-t)\frac{\partial}{\partial t}\hat{f}_{1,t\to r}(g) \\
&+ D\hat{f}_{1,t\to r}(g)[\hat{f}^f_{1,t}(g) - g]) + \hat{f}^f_{1,t}(g), \hat{f}^\theta_{1,t\to r}(g)\rangle],
\end{aligned}
\tag{87}
$$

where ① follows by substituting Equation 14, ② applies the relationship between u_t and u^f_t given in Equation 79, and ③ uses the exchangeability between Bochner integrals and inner products, together with the Fubini–Tonelli theorem. Therefore, we have $\tilde{\mathcal{L}}^{\mathcal{F}}_c(\theta) = \tilde{\mathcal{L}}^M(\theta) + C$ where C is independent of the model parameters θ. □

C. Model Architecture and Details of Dataset, Training and Sampling

C.1. Real-World Functional Generation

Models For the real-world functional generation experiments, including 1D time-series and 2D Navier–Stokes data, we follow the setup of [27] and compare with FFM, DDO, FDDPM and GANO. The implementations of GANO, DDPM, and DDO are directly adopted from [27]; please refer to [27] for additional details. FFM employs a 4-layer Fourier Neural Operator (FNO) implemented using the NeuralOperator library. Following [27], we use linearly interpolated spatial coordinates in $[0, 1]$ as explicit position embeddings and scale the temporal condition t by 10^{-3} as a time embedding. The spatial and temporal embeddings are concatenated with the input data, yielding a total input dimension of data channels+2. Our method adopts the same architecture as FFM but introduces two temporal conditions, t and r. Both are scaled and concatenated with spatial embeddings and input data, resulting in data channels+3 input dimensions. Dataset-specific configurations, including the number of Fourier modes, input channels, hidden channels, projection channels, spatial dimensionality, and total parameter count, are summarized in Table 6.

Table 6. FNO configuration for different datasets in real-world functional generation experiments. Each model uses a 4-layer FNO implemented with the `neuralop` library.

Dataset	Fourier Modes	Input Channels	Hidden Channels	Projection Channels	Spatial Dim.	Total Params
AEMET	64	4	256	128	1D	9.4M
Gene	16	4	256	128	1D	3.2M
Population	32	4	256	128	1D	5.3M
GDP	32	4	256	128	1D	5.3M
Labor	32	4	256	128	1D	5.3M
Navier–Stokes (2D)	(32, 32)	5	128	256	2D	35.9M

Dataset&Metrics Following [27], our experiments cover both 1D time-series and 2D Navier–Stokes functional datasets. The Navier–Stokes dataset consists of 2D incompressible fluid flow solutions on a 64×64 periodic grid, originally introduced by [31]. To reduce redundancy and improve training efficiency, we randomly sample 20,000 frames from the original dataset for training. The 1D time-series category includes five datasets: AEMET, Gene, Population, GDP, and Labor. The AEMET dataset contains 73 temperature curves recorded by weather stations in Spain between 1980 and 2009, each represented over 365 daily points. The Gene Expression dataset comprises 156 gene-activity time series measured across 20 time steps. The Population dataset provides population trajectories for 169 countries from 1950 to 2018 (69 time points). The GDP dataset records GDP-per-capita time series for 145 countries over the same 69-year span. The Labor dataset contains quarterly labor-force measurements from 2017 to 2022 (24 time points) for 35 countries.

Following [27], for the five 1D time-series datasets, we evaluate the quality of generated functions using a set of statistical functionals, including mean, variance, skewness, kurtosis, and autocorrelation. For each functional, we compute its value over all generated functions and compare it with the corresponding ground-truth statistics from the real dataset using mean squared error (MSE). This captures the model's ability to reproduce key statistical characteristics of temporal signals. For the 2D Navier–Stokes dataset, we employ two complementary distribution-level metrics. **Density MSE** measures the statistical discrepancy between the marginal value distributions of real and generated samples. Each dataset is flattened into scalar values representing pointwise function evaluations, from which continuous

probability densities are estimated via kernel density estimation (KDE). The mean-squared difference between the estimated densities quantifies how well the generated data reproduce the overall statistical distribution of function values. **Spectrum MSE** evaluates the discrepancy between the average Fourier energy spectra of real and generated samples. Each sample is transformed into the frequency domain using a 2D FFT, and spectral energies are aggregated over wavenumber bands and averaged across the dataset. The resulting mean-squared error reflects the model's ability to match the multi-scale energy distribution of the target fluid dynamics.

Training&Sampling During training and sampling, Gaussian processes with a Matérn kernel are used to sample the initial noise functions accurately. For the 1D datasets, we use a kernel length of 0.01 and a kernel variance of 0.1, while for the 2D Navier–Stokes dataset we use a kernel length of 0.01 and a kernel variance of 1.0. All models are trained using the Adam optimizer. The training and sampling procedures for GANO, DDPM, DDO, and FFN follow [27], and we refer readers to that work for implementation details. For our method, in the 1D setting we use an initial learning rate of 1×10^{-3}. For the AEMET dataset, the learning rate is reduced by a factor of 0.1 after 50 epochs, while no decay is applied for the other datasets. In the 2D setting, we use an initial learning rate of 5×10^{-4}, which is decayed by a factor of 0.5 every 40 epochs. Consistent with Mean Flow, we employ an adaptive loss function $\mathcal{L} = w|\Delta|_2^2$, $w = \frac{1}{(|\Delta|_2^2+c)^p}$ where Δ denotes the regression error, $c > 0$ is a small stabilizing constant (set to 10^{-3} in our experiments), and $p = 0.75$. The time variables t and r are sampled using a lognormal distribution with mean -0.4 and variance 0.01 for the 1D datasets, and uniformly from the interval $[0, 1]$ for the 2D dataset. Their values are swapped whenever $t > r$. By default, r is set equal to t with a probability of 0.25, except for the Population, GDP, and Labor datasets, where a probability of 0.125 is used.

C.2. Image Generation Based on Functional

Models As illustrated in Figure 8, we adopt the neural architecture of Infty-Diff[4], where both the input and output are continuous image functions represented by randomly sampled subsets of coordinates. To handle such sparse input–output mappings, the network consists of two components: a Sparse Neural Operator and a Dense U-Net/UNO.

The Sparse Neural Operator processes the irregularly sampled pixels and maps them into feature vectors on the same subsets of coordinates. These features are then interpolated onto a lower-resolution dense grid using k-nearest neighbor (KNN) interpolation with neighborhood size 3. On the dense grid, a Dense U-Net/UNO is applied to extract high-level representations. Following Infty-Diff's observation that U-Net and UNO yield comparable results, we employ the U-Net for simplicity. The dense U-Net operates on a 128^2 base grid for image datasets with a resolution of 256^2, with 128 base channels and five resolution levels whose channel multipliers are $[1, 2, 4, 8, 8]$. Self-attention modules are inserted after the 16^2 and 8^2 resolution stages to enhance global context aggregation. After dense processing, the resulting features are inversely interpolated back to the coordinate subsets using KNN. The reconstructed features are further refined through another Sparse Neural Operator, and the final output is obtained via a residual connection with the initial sparse features.

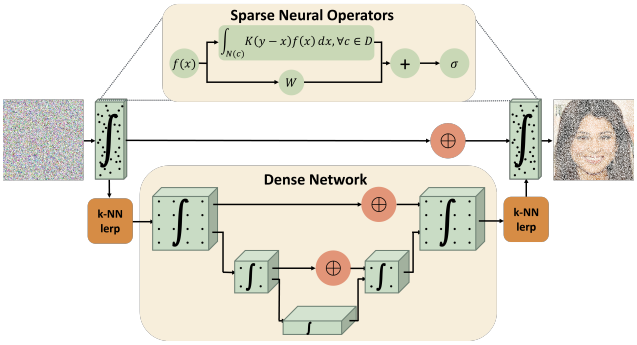


Figure 8. **Hybrid Sparse–Dense Neural Operator for Infty-Diff.** The model for functional-based image generation follows the hybrid sparse–dense Neural Operator design in Infty-Diff, where both the input and output are functions represented by randomly sampled pixels. The architecture consists of a Sparse Neural Operator and a Dense Network, and this figure presents the internal structure corresponding to the schematic in Figure 5

Following the implementation guidelines of Infty-Diff, we adopt a linear-kernel Sparse Neural Operator for efficiency. While Infty-Diff implements this operator with TorchSparse, our method computes gradients via PyTorch JVP, which is not compatible with TorchSparse. To address this issue, we emulate the sparse computation scheme of TorchSparse by masking feature maps. Each Sparse Operator module consists of an initial pointwise convolution layer, three linear-kernel convolution operator layers, and a final pointwise convolution layer. Each operator layer consists of a sparse depthwise convolution with 64 channels and a kernel size of 7 for 256^2-resolution images, followed by two pointwise convolution layers with 128 internal channels.

For time conditioning in both the Sparse Neural Operator and the Dense Network, we use positional embeddings [54] to encode the time variable following Mean Flow [16]. The resulting embeddings of t and r are added to replace the original time-embedding conditional input in Infty-Diff. In total, the network comprises $\sim 420M$ trainable parameters.

Dataset&Metrics We trained our models on three unconditional datasets CelebA-HQ [24], FFHQ [25], and LSUN-Church [57] and one conditional dataset AFHQ [6]. All datasets were resized to a resolution of 256×256 with LSUN-Church center-cropped along the shorter side. Following Infty-Diff [4], we randomly sampled one quarter of the pixels during training to validate functional-based generation. CelebA-HQ is a high-quality derivative of CelebA containing 30,000 high-resolution human face images. FFHQ includes 70,000 diverse face images, LSUN-Church provides around 126,000 images of churches, and AFHQ comprises 15,000 animal images from three categories: cats, dogs, and wild animals, which are used as conditional labels for conditional generation. To evaluate the models, we generated 50K samples from each trained model and compared them with the corresponding real datasets. Following [28], we computed the Fréchet Inception Distance (FID) [19] using CLIP features [44] extracted from the clip_vit_b_32 encoder, which correlates better with human perception of image quality, especially for multiple-resolution generation. We denote this metric as FID_{CLIP}. For completeness and comparison with previous works, we also report the standard FID computed using Inception-V3 features, denoted as FID.

Training&Sampling In the functional-based image generation experiments, we follow Infty-Diff [4] and use mollified white noise to approximate an infinite-dimensional Gaussian process sampled from a Gaussian measure $\mathcal{N}(0, C_0)$ as the initial noise function for training and sampling. Specifically, the white noise is convolved with a Gaussian kernel $k(\cdot)$ to ensure that the resulting samples lie in the Hilbert space $\mathcal{F}$. The mollification is expressed as

$$\begin{aligned} h(c) &= \int_{\mathbb{R}^n} K(c-y, l)x(y)dy \\ K(y, l) &= \frac{1}{(4\pi l)^{n/2}} e^{-\frac{|y|^2}{4l}}, \end{aligned} \tag{88}$$

where $l > 0$ is the smoothing parameter, which we set to one pixel width in our experiments. For image data, we take $n = 2$. Here, $x(y)$ denotes the original white noise before smoothing, and $h(c)$ represents the mollified function after applying the Gaussian kernel. We also apply the same Gaussian mollification to the training images in the dataset to improve regularity and ensure the data distribution satisfies the integrability requirements of the function space. Following Infty-Diff, the generated mollified output is then restored to a sharper image using a Wiener-filter–based approximate inverse defined in the Fourier domain as

$$\tilde{x}(\omega) = \frac{e^{-\omega^2 t}}{e^{-2(\omega^2 t)} + \epsilon^2} \hat{h}(\omega), \tag{89}$$

where ϵ denotes an estimate of the inverse signal-to-noise ratio (SNR). Here, $\hat{h}(\omega)$ and $\tilde{x}(\omega)$ denote the Fourier transforms of the mollified function $h(c)$ and the reconstructed output $x(c)$, respectively. This operation effectively recovers high-frequency details while maintaining numerical stability.

All models are trained on four NVIDIA L40s GPUs using the Adam optimizer for a total of 800K steps, with the learning rate gradually reduced from an initial value of 5.0×10^{-5} to a final value of 7.8×10^{-7}. The total batch size is set to 16 for all experiments. Consistent with Mean Flow, we employ an adaptive loss $\mathcal{L} = w|\Delta|_2^2$, where Δ denotes the regression error and $w = 1/(|\Delta|_2^2 + c)^p$ with a small constant $c > 0$ (e.g., 10^{-3}). We set $p = 0.5$ for all experiments. The time variables t and r are sampled uniformly from the interval $[0, 1]$, swapping their values whenever $t > r$, and setting $r = t$ with a probability of 0.5. Following Infty-Diff [4], we employ the manner of Diffusion Autoencoder [43] to mitigate stochasticity arising from the high variance of randomly sampled coordinate subsets.

C.3. 3D Shape Generation

Models. We use the architecture of Functional Diffusion [58] for 3D shape generation as shown in Figure 9. In this model, both the input and output functions are represented by randomly sampled spatial points and their corresponding function values, enabling a continuous functional representation independent of discretized grids. Specifically, the input function f_c is represented on context points $\{x_c^i\}_{i=1}^n$ with values $\{v_c^i\}_{i=1}^n$, where $v_c^i = f_c(x_c^i)$, and the output function f_q is represented on query points $\{x_q^j\}_{j=1}^m$ with values $\{v_q^j\}_{j=1}^m$ where $v_q^j = f_q(x_q^j)$. n and m are the number of context points and query points respectively. The Functional Diffusion framework naturally supports the case where the context and query points differ, allowing flexible mappings between input and output functions.

Following [58], we evenly partition the context set $(\{x_c^i\}_{i=1}^n, \{v_c^i\}_{i=1}^n)$ into d groups $I_1, \ldots, I_d$, where $\cup_{k=1}^d I_k = \{1, \ldots, n\}$. Each group is processed by an attention block consisting of a cross-attention and a self-attention module. The cross-attention module takes the embedding of $(\{x_c^i\}_{i\in I_k}, \{v_c^i\}_{i\in I_k})$ and a latent vector. The latent vector is propagated from the previous attention block, while the first block initializes it with a learnable latent variable $\mathcal{X}$ that represents the functional itself. For each group $(\{x_c^i\}_{i\in I_k}, \{v_c^i\}_{i\in I_k})$, a point-wise embedding is obtained by summing the Fourier positional encoding of spatial coordinates $\{x_c^i\}_{i\in I_k}$ and the embedding of the corresponding values $\{v_c^i\}_{i\in I_k}$. The resulting embeddings are concatenated with a conditional embedding, which may come from semantic labels or partially observed conditional points. In our experiments, we use 64 partially observed surface points as conditional inputs. These combined embeddings

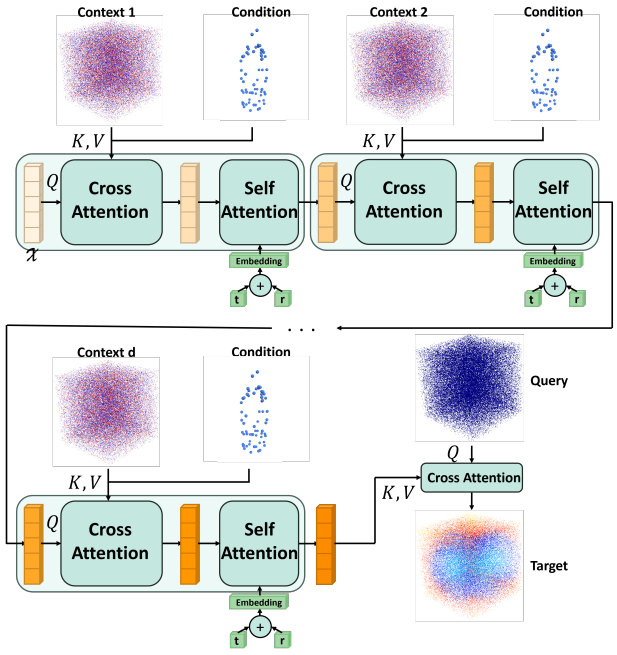


Figure 9. **The Architecture of Functional Diffusion**. Both the input and output of the model are functions, which are represented by randomly sampled spatial points and their corresponding function values. The input function is referred to as the context, and the output function as the query. Each context point and its associated value jointly form the representation of input function. The context points and their values first interact with the learnable vector $\mathcal{X}$ through cross-attention, followed by cascaded cross- and self-attention m odules that progressively yield the latent vector representing the output function. The latent vector then interacts with the query points through another cross-attention layer to predict the query values, together forming the output function. Following [58], the model is conditioned on 64 target surface points, from which it reconstructs the target surface.

are passed through K and V networks to produce the keys and values for the cross-attention, while the latent vector, transformed by a Q network, serves as the query. The temporal embedding follows Mean Flow [16], employing standard sinusoidal positional encodings [54]. The sum of the embeddings of t and r replaces the time-embedding input originally used in Functional Diffusion.

After cascading d attention blocks, the resulting latent vector encodes the representation of the target function. This latent is then used as the key and value input to a final cross-attention block, where the query corresponds to the Fourier positional encodings of the query points $\{x_q^j\}_{j=1}^m$. The attention output yields the predicted query values $\{v_q^j\}_{j=1}^m$, forming the representation of the generated function f_q with $\{x_q^j\}_{j=1}^m$. In our implementation, the embedding dimension is set to 784, the number of groups d is 24, and the attention layers adopt a multi-head mechanism with 8 heads, each with an internal head dimension of 64.

Task&Dataset&Metrics We follow the surface reconstruction task setup in Functional Diffusion [58], where the model is required to reconstruct a target surface given 64 observed points sampled from that surface. Specifically, the generative model is conditioned on these 64 target surface points to generate the SDF function of the corresponding complete surface from the initial noise. Consistent with Functional Diffusion [58], we use the ShapeNet-CoreV2 [5] dataset, which contains 57,000 3D models across 55 object categories. Following the same preprocessing procedure [58–60], we convert each ShapeNet mesh into a signed distance field (SDF) and randomly sample $n = 49152$ points from the domain $[0, 1]^3$ to obtain the context points and their corresponding SDF values. Separately, we sample another $m = 2048$ points and their SDF values to form the query points and query values, while a distinct set of surface points is sampled near the zero-level set of the SDF to serve as the conditioning input. During training, each data instance consists of context points $\{\tilde{x}_c^i\}_{i=1}^n$ and values $\{\tilde{v}_c^i\}_{i=1}^n$, query points $\{\tilde{x}_q^j\}_{j=1}^m$ and values $\{\tilde{v}_q^j\}_{j=1}^m$, and a randomly selected set of 64 surface points $\{C^l\}_{l=1}^{64}$ used as the conditional input. Note that $(\{\tilde{x}_c^i\}_{i=1}^n, \{\tilde{v}_c^i\}_{i=1}^n)$ and $(\{\tilde{x}_q^j\}_{j=1}^m, \{\tilde{v}_q^j\}_{j=1}^m)$ correspond to the same sample, the reference SDF field $f \sim \nu$ from the dataset, rather than the context and query functions used as the input g and output $\hat{f}_{1,t\to r}(g)$ in Functional Mean Flow training; the construction of the input and reference output functions from each data instance is detailed in the Training&Sampling paragraph.

We evaluate the model using Chamfer Distance, F1-score, and Boundary Loss, following [58–60]. The Chamfer Distance measures the average bidirectional distance between the generated and target point sets, while the F1-score quantifies the precision–recall trade-off of the reconstructed surface points. The Boundary Loss assesses geometric fidelity near the zero-level surface and is formally defined as $\text{Boundary}(f) = \frac{1}{|\mathcal{E}_\Omega|}\sum_{i\in\mathcal{E}_\Omega} |f(\mathbf{x}_i) - q(\mathbf{x}_i)|^2$ where $\mathcal{E}_\Omega$ denotes the set of sampled spatial points near the surface boundary, $f(\mathbf{x}_i)$ represents the predicted SDF value, and $q(\mathbf{x}_i)$ is the ground-truth SDF. This metric measures the mean squared deviation between predicted and true SDFs in the boundary region, capturing the fine-grained accuracy of surface reconstruction. For evaluation, both Chamfer Distance and F1-score are computed by uniformly sampling 50K points on each surface, whereas Boundary Loss is computed using 100K sampled points. We follow the same data split as [58], training the model on the training split and evaluating it on the test split.

Training&Sampling. In our 3D shape generation experiments, we follow Functional Diffusion [58] and approximate samples from the Gaussian measure by linearly interpolating white noise defined on a coarse 64^3 lattice over the

3D domain. This interpolated field serves as an efficient estimate of a Gaussian process sample, providing a computationally practical alternative to direct Gaussian process sampling and substantially improving sampling efficiency in the 3D setting. Since we adopt the x_1-prediction variant of Functional Mean Flow for model training, the neural network input is $g = (1-(1-\sigma_{\min})t)f_0 + tf$ in Equation 5, where f denotes the function sampled from the dataset. In practice, f corresponds to each function represented by the instance $(\{\tilde{x}_c^i\}_{i=1}^n, \{\tilde{v}_c^i\}_{i=1}^n)$ and $(\{\tilde{x}_q^j\}_{j=1}^m, \{\tilde{v}_q^j\}_{j=1}^m)$ from dataset. The network output is then used jointly with the predicted $\hat{f}_{1,t}^f(g)$ to compute the training loss Equation 12. Therefore, context points and context values that represent the input function g to the network are calculated as $x_c^i = \tilde{x}_c^i$ and $v_c^i = t\tilde{v}_c^i + (1-(1-\sigma_{\min})t)r_c^i$ from the the instance $(\{\tilde{x}_c^i\}_{i=1}^n, \{\tilde{v}_c^i\}_{i=1}^n)$ and $(\{\tilde{x}_q^j\}_{j=1}^m, \{\tilde{v}_q^j\}_{j=1}^m)$, whereas the query points representing the output function and the corresponding reference query values of $\hat{f}_{1,t}^f(g)$ are obtained following $x_q^j = \tilde{x}_q^j$ and $v_q^j = \tilde{v}_q^j + \sigma_{\min} r_q^j$, because $\hat{f}_{1,t}^f(g)$ from Equation 13 can be computed as

$$\begin{aligned}\hat{f}_{1,t}^f(g) &= \frac{\sigma_{\min}}{1-(1-\sigma_{\min})t}(g-tf)+f \\ &= \sigma_{\min} f_0 + f\end{aligned} \tag{90}$$

Here r_c^i and r_q^j represent the values of initial noise function $f_0 \sim \mu_0$ in Equation 5 and Equation 13 evaluated at the context and query points x_c^i and x_q^j, respectively. Besides the separate reference term $\hat{f}_{1,t}^f(g)$ in Equation 12, the term $D\hat{f}_{1,t\to r}(g)[\hat{f}_{1,t}^f(g) - g]$ also involves an $\hat{f}_{1,t}^f(g)$. This instance of $\hat{f}_{1,t}^f(g)$ should be evaluated using the context points, denoted as $(\{\tilde{x}_c^i\}_{i=1}^n, \{\tilde{v}_c^i + \sigma_{\min} r_c^i\})_{i=1}^n$, since it serves as part of the functional derivative with respect to the network input g.

The model is trained on four H200 GPUs using the Adam optimizer for a total of 200K steps. The learning rate is gradually reduced from an initial value of $1e-5$ to a final value of $1e-7$. The total batch size is set to 16 for all experiments. Consistent with Mean Flow, we employ an adaptive loss function $\mathcal{L} = w|\Delta|_2^2$, $w = \frac{1}{(|\Delta|_2^2+c)^p}$ where Δ denotes the regression error, $c > 0$ is a small stabilizing constant (set to 10^{-3} in our experiments), and $p = 0.5$. The time variables t and r are uniformly sampled from the interval $[0,1]$, with their values swapped whenever $t > r$, and r is set equal to t with a probability of 0.5.

During sampling, we randomly draw $n = 49152$ points from the domain $[0,1]^3$ as context points, and use a dense 128^3 grid as query points. After predicting the SDF values on the 128^3 grid, the final mesh surface is reconstructed using the Marching Cubes algorithm.

D. Example Python Implementation

D.1. Unified Implementation

D.1.1. Unified Implementation for u-prediction Functional Mean Flow

In the following training code, `gp_like(g)` is the Gaussian Process sampling function (for finite-dimensional cases it can be replaced by `randn_like()`), and `sample_t_r()` is the time sampling function. `u(g, r, t)` denotes the learned model whose input and output are functions in a specific representation, while `f` denotes a batch of training data under the same representation. The parameter `sigma_min` corresponds to $\sigma_{\min}$ in Equation 5, and `metric` denotes the loss function.

```
t, r = sample_t_r()
f_0 = gp_like(f)

coef = 1-sigma_min
g = (1 - coef*t) * f_0 + t * f
v = coef/(1-coef*t)*(t*f-g)+ f

u, dudt = jvp(u, (g, t, r), (v, 1, 0))
u_tgt = (r - t) * dudt + v
error = u - stopgrad(u_tgt)

loss = metric(error)
```

The following code is for inference

```
f_0 = gp_like(f)
f=u(f_0,0,1)+f_0
```

D.1.2. Unified Implementation for x_1-prediction Functional Mean Flow

In the following code, `gp_like(g)` is the Gaussian Process sampling function (for finite-dimensional cases it can be replaced by `randn_like()`), and `sample_t_r()` is the time sampling function. `x1(g, r, t)` denotes the learned model whose input and output are functions in a specific representation, while `f` denotes a batch of training data under the same representation. The parameter `sigma_min` corresponds to $\sigma_{\min}$ in Equation 14, and `metric` denotes the loss function.

```
t, r = sample_t_r()
f_0 = gp_like(f)

coef = 1-sigma_min
g = (1 - coef*t) * f_0 + t * f
f1_f = sigma_min/(1-coef*t)*(g-t*f)+ f
```

```
f1, df1dt = jvp(x1, (g, t, r), (f1_f-g, 1-t, 0))
f1_tgt = (r - t)/(1-r) * df1dt + f1_f
error = f1 - stopgrad(f1_tgt)

loss = metric(error)
```

The following code is for inference

```
f_0 = gp_like(f)
f=x1(f_0,0,1)
```

D.2. 3D SDF-Specific Implementation

As discussed in subsection C.3, each training instance in Functional Diffusion consists of a tuple $(\{\tilde{x}_c^i\}_{i=1}^n, \{\tilde{v}_c^i\}_{i=1}^n, \{\tilde{x}_q^j\}_{j=1}^m, \{\tilde{v}_q^j\}_{j=1}^m, \{C^l\}_{l=1}^{64})$, representing (context points, context values, query points, query values, and condition). The pairs $\{\tilde{x}_c^i\}_{i=1}^n, \{\tilde{v}_c^i\}_{i=1}^n$ and $\{\tilde{x}_q^j\}_{j=1}^m, \{\tilde{v}_q^j\}_{j=1}^m$ correspond to two different samplings of the same reference SDF function $f \sim \nu$, differing only in their spatial locations $\tilde{x}_c^i$ and $\tilde{x}_q^j$. It is therefore crucial in implementation to clearly distinguish between these two representations to avoid ambiguity during training. In practice, the input function $g \sim \mu_t^f$ is constructed as $(\{\tilde{x}_c^i\}_{i=1}^n, \{t\tilde{v}_c^i + (1-(1-\sigma_{\min})t)r_c^i\}_{i=1}^n)$, the separate reference function $\hat{f}_{1,t}^f(g)$ is evaluated as $(\{\tilde{x}_q^j\}_{j=1}^m, \{\tilde{v}_q^j + \sigma_{\min} r_q^j\}_{j=1}^m)$, and the reference function $\hat{f}_{1,t}^f(g)$ in $D\hat{f}_{1,t\to r}(g)[\hat{f}_{1,t}^f(g) - g]$ is evaluated as $(\{\tilde{x}_c^i\}_{i=1}^n, \{\tilde{v}_c^i + \sigma_{\min} r_c^i\}_{i=1}^n)$, where r_c^i and r_q^j denote the values of the initial noise function $f_0 \sim \mu_0$ in Equation 5 and Equation 13, evaluated at the context and query points x_c^i and x_q^j, respectively. Temporal conditioning follows the Mean Flow formulation [16], where sinusoidal embeddings of time variables t and r [54] are summed and injected in place of the original time embedding used in Functional Diffusion. After d attention blocks, the final latent encodes the target function representation. A concluding cross-attention layer then maps this latent to the query points, using their Fourier positional encodings as queries, to predict the output values $v_{q_{j=1}^m}^{j^m}$. In our implementation, we use an embedding dimension of 784, $d = 24$ groups, and multi-head attention with 8 heads, each of dimension 64.

In the code, `xc`, `vc`, `xq`, `vq`, and `cond` respectively denote the context point $\{\tilde{x}_c^i\}_{i=1}^n$, context value $\{\tilde{v}_c^i\}_{i=1}^n$, query point $\{\tilde{x}_q^j\}_{j=1}^m$, query value $\{\tilde{v}_q^j\}_{j=1}^m$, and condition $\{C^l\}_{l=1}^{64}$, respectively. As mentioned earlier, Functional Diffusion constructs the initial Gaussian measure using linear interpolation over a random value on coarse grid. In the training code, `s` specifies the coarse grid resolution, and `interpolate(rg, xc)` performs interpolation from the random grid values `rg` to the sample points `xc`. The function `sample_t_r()` is the time sampling function, and `x1(xc, g, xq, r, t, cond)` denotes the learned model, where `(xc, g)` represents the input function g, and `xq` specifies the query points for the output function. `B` is batch size.

```
    rg = torch.randn(B, 1,s, s, s)
    rc = interpolate(rg,xc)
    rq = interpolate(rg,xq)
    t, r = sample_t_r()
    coef = 1-sigma_min
    g = (1 - coef*t) * rc + t * vc
    f1_f_c = sigma_min*rc+vc
    f1_f_q = sigma_min*rq+vq
    x1_partial = partial(x1, xc = xc, xq = xq, cond=cond)
    f1, df1dt = jvp(x1_partial, (g, t, r), (f1_f_c-g,1-t,0))
    f1_tgt = (r - t)/(1-r) * df1dt + f1_f_q
    error = f1 - stopgrad(f1_tgt)
    loss = metric(error)
```

As mentioned earlier, during inference we use a dense 128^3 grid as query points. In the following code, these query points are denoted as `xqg`. The following code is for inference

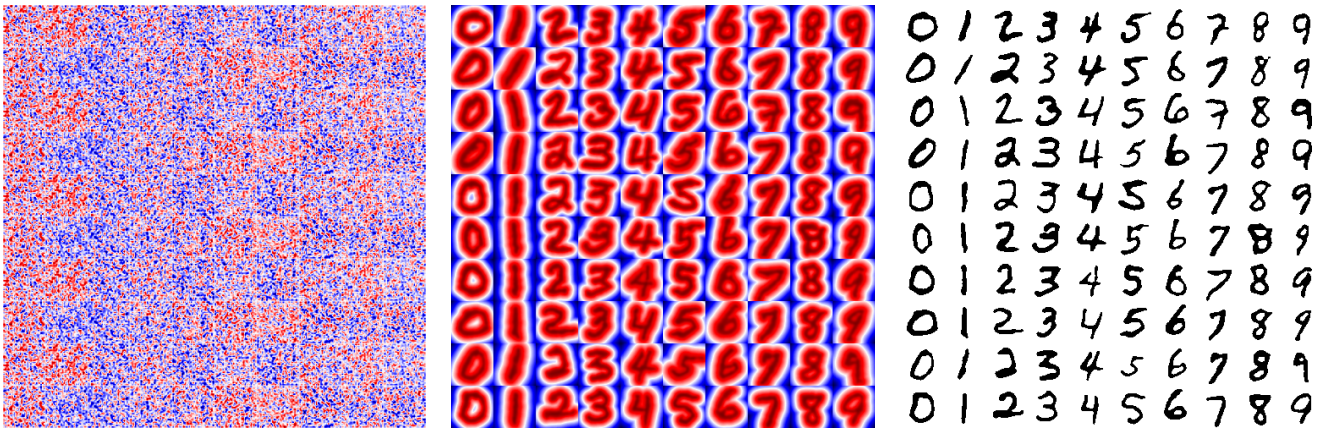
```
rg = torch.randn(B, 1,s, s, s)
rc = interpolate(rg,xc)
vqg = x1(xc, rc, xqg, 0, 1, cond)
```

E. Additional Results&Experiments

E.1. Instability of u-Prediction Mean Flow in Shape Generation with SDF

In other tasks, the performance of the u-prediction variant of Functional Mean Flow is generally comparable to that of the x_1-prediction version. However, in the shape generation experiments, we observe that the Functional Diffusion framework becomes highly unstable when trained with u-prediction Functional Mean Flow, indicating that u-prediction Functional Mean Flow is not well-suited for 3D shape generation within the Functional Diffusion framework.

To illustrate this finding, we conduct a 2D experiment using the MNIST [30] dataset, which is converted into signed distance fields (SDFs) and trained under the Functional Diffusion setup for 2D shape generation. The embedding dimension is set to 256, the number of groups d is 8, and each attention layer adopts a multi-head mechanism with 8 heads



(a) MNIST u-prediction (b) MNIST x_1-prediction (SDF) (c) MNIST x_1-prediction (binary)

Figure 10. Comparison of MNIST generation results across the u-prediction and x_1-prediction FMF variants within the Functional Diffusion framework. After 10K training steps, the x_1-prediction FMF produces valid SDFs (b), from which clear binary digits can be extracted via their zero-level sets (c). In contrast, the u-prediction model fails completely: once the output variance collapses, the model cannot update the initial noise into a meaningful SDF, as shown in (a) and rows correspond to learning rates from 10^{-4} to 10^{-6}.

and an internal head dimension of 64. The entire network contains approximately 19M learnable parameters. During training, we use a batch size of 64 and initialize the time embedding with a standard variance of 0.01. We set $t = r$ with 100% probability for time sampling, representing the most stable limiting form of Mean Flow

To monitor potential training failures, we track the batch-averaged spatial variance of the network outputs. A persistent collapse of this variance indicates instability, as an SDF is expected to satisfy $|\nabla f| = 1$ and thus maintain meaningful spatial variation. Once the variance approaches zero and remains there, the model's output effectively degenerates into a constant field, diverging from the ground-truth function and failing to recover. As illustrated in Figure 6, the x_1-prediction formulation remains stable across a wide range of learning rates, whereas the u-prediction model exhibits variance collapse even at relatively small learning rates.

After training for 10K steps, we visualize the generated samples in Figure 10. The number of sampling steps is fixed to 64. It can be observed that the x_1-prediction Mean Flow successfully generates valid SDFs, from which the zero-level sets can be extracted as clear binary handwritten digits. In contrast, the u-prediction variant consistently produces noisy outputs regardless of the learning rate.

E.2. Additional Results for Functional-Based Image Generation

In Figure 12, Figure 13, and Figure 11, we provide additional qualitative results on the FFHQ [25], LSUN-Church [57], and AFHQ [6] datasets, respectively. The same model is used to synthesize images at arbitrary resolutions under different noise levels. Notably, the model is trained only on randomly sampled 1/4 subsets of pixels from 256×256 images.

Figure 11. Additional results on AFHQ. The model is trained on randomly sampled 1/4 subsets of pixels from 256×256 images and evaluated at different resolutions. Left to right: 64×64, 128×128, 256×256, and 512×512. Top to bottom: cat, dog, and wild animal categories.

Figure 12. Additional results on FFHQ. The model is trained on randomly sampled 1/4 subsets of pixels from 256×256 images and evaluated at different resolutions. Left to right: 64×64, 128×128, 256×256, and 512×512.

Figure 13. Additional results on LSUN-Church. The model is trained on randomly sampled 1/4 subsets of pixels from 256×256 images and evaluated at different resolutions. Left to right: 64×64, 128×128, 256×256, and 512×512.