NRFlow: Towards Noise-Robust Generative Modeling via High-Order Flow Matching

Bo Chen* Chengyue Gong† Xiaoyu Li‡ Yingyu Liang§ Zhizhou Sha¶
Zhenmei Shi‖ Zhao Song** Mingda Wan†† Xugang Ye‡‡

Abstract

Flow-based generative models have shown promise in various machine learning applications, but they often face challenges in handling noise and ensuring robustness in trajectory estimation. In this work, we propose NRFlow, a novel extension to flow-based generative modeling that incorporates second-order dynamics through acceleration fields. We develop a comprehensive theoretical framework to analyze the regularization effects of high-order terms and derive noise robustness guarantees. Our method leverages a two-part loss function to simultaneously train first-order velocity fields and high-order acceleration fields, enhancing both smoothness and stability in learned transport trajectories. These results highlight the potential of high-order flow matching for robust generative modeling in complex and noisy environments.

* bc7b@mtmail.mtsu.edu. Middle Tennessee State University.
† cygong17@utexas.edu. The University of Texas at Austin.
‡ 7.xiaoyu.li@gmail.con. Independent Researcher.
§ yingyul@hku.hk. The University of Hong Kong. yliang@cs.wisc.edu. University of Wisconsin-Madison.
¶ shazz20@mails.tsinghua.edu.cn. Tsinghua University.
‖ zhmeishi@cs.wisc.edu. University of Wisconsin-Madison.
** magic.linuxkde@gmail.com. The Simons Institute for the Theory of Computing at UC Berkeley.
†† dylan.r.mathison@gmail.com. Anhui University.
‡‡ xugangye@ams.jhu.edu. Johns Hopkins University.

Contents

1 Introduction **3**

2 Related Work **4**

3 Preliminary **4**
- 3.1 Notations . . . 5
- 3.2 Assumptions . . . 5
- 3.3 Flow Matching and Rectified Flow . . . 6
- 3.4 Proposed Method . . . 8

4 Our Result **8**
- 4.1 Elliptic Regularity . . . 8
- 4.2 Regularization Effect . . . 9
- 4.3 Excess Risk . . . 9
- 4.4 Discrete Propagation under Noise . . . 12
- 4.5 Main Result . . . 13

5 Experiments **14**
- 5.1 Experiment Setup . . . 14
- 5.2 Results Analysis . . . 15

6 Conclusion **15**

A Preliminary **24**
- A.1 Notations . . . 24
- A.2 Flow Matching . . . 24
- A.3 Optimal Transport . . . 25

B More related work **26**

C Missing proof of Theorem 4.7 **27**

D Extension on third-order Flow Matching **28**
- D.1 Preliminary . . . 28
- D.2 Proposed Third-Order Algorithms . . . 29
- D.3 Elliptic Regularity . . . 30
- D.4 Regularization Effect . . . 30
- D.5 Excess Risk . . . 31
- D.6 Discrete Propagation . . . 33
- D.7 Main Result: Third-Order Noise Robustness . . . 34

E Extension on k-th order Flow Matching **35**
- E.1 Preliminary . . . 35
- E.2 Proposed k-th Order Algorithms . . . 36
- E.3 Elliptic Regularity . . . 37
- E.4 Regularization Effect . . . 37
- E.5 Excess Risk . . . 38
- E.6 Discrete Propagation . . . 39

E.7 Main result for k-th Order Noise Robustness 40

F Empirical Ablation Study **41**
F.1 Three Dataset 41
F.2 Only First Order Loss 42
F.3 Second Order NRFlow 42
F.4 Third Order NRFlow 43

1 Introduction

Flow-based generative modeling [LCBH+22, LGL22, AVE22, BASH+23, EKB+24] has recently gained substantial traction in machine learning due to its capacity to learn expressive, invertible transformations that map simple source distributions to more complex target distributions. In particular, flow matching techniques [LCBH+22, LGL22] have shown promising results in bridging the gap between traditional normalizing flows and score-based diffusion models. These methods typically construct a continuous time trajectory or "flow" that transports samples from a prior distribution, usually Gaussian distribution, to an unknown data distribution. By matching a parameterized velocity field to the ground-truth time derivatives along a path connecting these distributions, flow matching has demonstrated impressive empirical performance, as well as favorable theoretical properties.

Despite these advances, existing flow-based frameworks remain susceptible to perturbations such as noise contamination in the data or instability in the learned transport path [WET+24, HWA+24]. This vulnerability arises because standard (i.e., first-order) methods predominantly focus on velocity alignment, thereby neglecting higher-order dynamics and their influence on smoothness and robustness. Several works in diffusion-based modeling [Che23, HG24, LLLY24] have suggested that carefully accounting for noise and incorporating additional constraints can lead to more stable solutions. However, a principled and comprehensive approach to integrating higher-order information within flow matching has yet to be fully explored.

In this paper, we propose NRFlow, a novel extension to the traditional flow-based generative framework that leverages acceleration fields in addition to velocity fields. Our approach is motivated by the observation that second-order information can be interpreted as a form of regularization, acting to enforce higher-order smoothing constraints on the learned trajectories. Concretely, we show—both formally and informally—that these second-order terms can mitigate noisy or imperfect training data by providing stronger regularity conditions, which in turn bolster model robustness. The core idea is straightforward but powerful: we decompose the learning objective into two parts, one for velocity matching and one for acceleration matching, and jointly train these terms to ensure smooth, stable flows.

Our main theoretical contributions center on establishing a rigorous noise-robustness guarantee that quantifies how noise in the observed data propagates through the learned second-order flow. Specifically, we derive a two-part loss function whose first-order component learns a velocity field approximating $\dot{x}_t$, while the second-order component targets $\ddot{x}_t$. We then prove that if these losses remain small, the Sobolev H^2-norm of the estimation error is bounded, effectively demonstrating the regularization effect of the second-order term. Further, a discrete Gronwall-type analysis reveals that noise in the training distribution propagates sublinearly over time, thereby yielding improved stability compared with purely first-order flow matching.

In addition to the theoretical framework, we also propose a second-order inference algorithm that modifies the classic flow integration step by adding an acceleration update. Experimental results on a Gaussian mixture dataset highlight the effectiveness of our approach.

The summary of our contributions to the theoretical understanding of these architectures and their boundaries, showed as follows:

- We develop a unified second-order flow formulation offering a broad perspective on incorporating higher-order dynamics into generative models.
- We establish rigorous guarantees showing that NRFlow exhibits improved immunity to data noise, grounded in both an informal explanation of its regularization properties and a formal theorem bounding noise propagation.

2 Related Work

Flow Matching. Flow Matching (FM) [LHH+24] has recently gained prominence in generative modeling, particularly within the framework of Continuous Normalizing Flows (CNFs). FM offers a simulation-free approach to training CNFs by regressing vector fields along fixed conditional probability paths, thereby enhancing scalability and performance in generative tasks [LCBH+22]. Building upon this foundation, [TFM+23] developed Conditional Flow Matching (CFM), a family of simulation-free training objectives for CNFs. CFM facilitates conditional generative modeling and accelerates both training and inference processes. An exciting development in this area is the introduction of Rectified Flow, which refines flow-based methods by incorporating corrective adjustments to the learned vector fields, enabling more robust convergence and improved stability in generative modeling tasks. Rectified Flow not only enhances training efficiency but also synergizes effectively with other flow-matching methods, further extending the utility of FM in diverse applications. A notable variant within CFM, Optimal Transport Conditional Flow Matching (OT-CFM), approximates dynamic optimal transport in a simulation-free manner, leading to more efficient and stable training. Recent advancements in flow matching for generative modeling have introduced several innovative approaches. [HPPA24] proposed Wasserstein Flow Matching, extending traditional flow matching to families of distributions, enhancing its applicability in fields like computer graphics and genomics. Moreover, numerous recent works [XZC+22, DWB+23, PBHDE+23, WSD+23, WCZ+23, WXZ+24, CL24, KKN24, BRSR24] have significantly inspired and influenced our work.

Second Order Method. More recently, second-order methods have been applied to neural network optimization and used to solve a lot of problems. [M+10] introduced Hessian-free optimization, using conjugate gradients to approximately solve the Newton update. [VP12] used a Krylov subspace descent method to directly approximate the Newton update. For natural gradient methods, which perform the steepest descent in the space of network outputs rather than parameters, [Ama98] showed a connection to second-order optimization via the Fisher information matrix. [GM16] later extended K-FAC to convolutional neural networks. [BGM22] combined natural gradient with trust region methods to further improve stability and performance. Despite these advances, second-order neural network optimization remains an active area of research, such as [DSWY22, SWY23, GSWY23, GSY23b, GSY23a, BSY23, DMS23, GMS23, SSX23, QSS23, CLS+24, CLL+24b, LSS+24b, CLL+24a, LSSS24, LSS+24a, KLL+25]. Open problems include improving the scalability of Hessian approximations, handling non-convex optimization landscapes, and automating hyper-parameter selection.

Roadmap. In Section 3, we introduce essential computational techniques and key definitions of flow matching and our NRFlow. In Section 4, we provide a detailed regularity analysis and compute the upper bound of the excess risk. We also provide an inequality that quantifies the growth of estimation error under bounded noise. In Section 5, We design some preliminary experiments to demonstrate the validity of our theory and the results. We conclude in Section 6.

3 Preliminary

In this section, we provide the foundational concepts, notations, and assumptions required for the subsequent theoretical developments. In Section 3.1, we begin by listing the key notations employed throughout this work. In this Section 3.2, we state the principal assumptions under which our analysis is conducted. Next, we introduce the flow-matching framework, along with its second-order extension, and highlight several important definitions in Section 3.3. Finally, in

Section 3.4, we provide our second-order algorithms.

3.1 Notations

We use Pr[] to denote the probability. We use $\mathbb{E}[]$ to denote the expectation. We use Var[] to denote the variance. We use $\|x\|_p$ to denote the ℓ_p norm of a vector $x \in \mathbb{R}^n$, i.e. $\|x\|_1 := \sum_{i=1}^n |x_i|$, $\|x\|_2 := (\sum_{i=1}^n x_i^2)^{1/2}$, and $\|x\|_\infty := \max_{i\in[n]} |x_i|$. For variables a, b, We write $a \lesssim b$ to indicate that a is bounded above by b up to a multiplicative constant independent of the main parameters. We write $a \gtrsim b$ to indicate that a is bounded below by b up to a multiplicative constant independent of the main parameters. We denote $\dot{x}^{(k)}$ as the k-th order derivative field of x. We use $\|\cdot\|_{H^2(\Omega)}$ to denote the Sobolev norm in $W^{2,2}(\Omega)$, corresponding to $k = 2$ and $p = 2$.

3.2 Assumptions

We now outline the principal assumptions that underlie our theoretical analysis. These assumptions concern smoothness, Lipschitz continuity, bounded noise, function-class complexity, and time discretization. First, we show the assumption of smoothness and boundness.

Assumption 3.1 (Smoothness and boundedness)**.** *We assume the true trajectory x_t^{true} and its first and second derivatives are sufficiently smooth and bounded. Specifically, $x_t^{\text{true}} \in H^2(\Omega)$, and there exist constants $M_1, M_2 > 0$ such that*

$$\|\dot{x}_t^{\text{true}}\|_{H^2(\Omega)} \le M_1, \quad \|\ddot{x}_t^{\text{true}}\|_{H^2(\Omega)} \le M_2.$$

Assumption 3.2 (Lipschitz continuity)**.** *The learned fields $u_{1,\theta_1}(x,t)$ and $u_{2,\theta_2}(v,x,t)$ are L-Lipschitz continuous in spatial and temporal arguments. Formally, there exists $L > 0$ such that for all $x, y \in \mathbb{R}^d$ and $t, s \in [0,1]$:*

$$\|u_{1,\theta_1}(x,t) - u_{1,\theta_1}(y,t)\|_2 \le L\|x-y\|_2,$$
$$\|u_{2,\theta_2}(v,x,t) - u_{2,\theta_2}(v,y,t)\|_2 \le L\|x-y\|_2,$$

and

$$\|u_{2,\theta_2}(v,x,t) - u_{2,\theta_2}(v,y,t)\|_2 \le L\|x-y\|_2,$$
$$\|u_{2,\theta_2}(v,x,t) - u_{2,\theta_2}(v,x,s)\|_2 \le L\|t-s\|_2,$$

Similar conditions hold for time differences.

Assumption 3.3 (Bounded noise magnitude)**.** *There exists $\delta > 0$ such that $\|\eta_i\|_2 \le \delta$ or $\mathbb{E}[\|\eta_i\|_2^2] \le \delta^2$. This ensures that the noise does not grow without bounds.*

Assumption 3.4 (Rademacher complexity or VC dimension)**.** *There exist function classes $\mathcal{F}_1, \mathcal{F}_2$ such that*

$$u_{1,\theta_1}(\cdot,\cdot) \in \mathcal{F}_1, \quad u_{2,\theta_2}(\cdot,\cdot) \in \mathcal{F}_2.$$

The complexity of each class is measured by $\mathcal{C}(\mathcal{F}_1)$ and $\mathcal{C}(\mathcal{F}_2)$.

Assumption 3.5 (Bounded loss)**.** *There exists some constant $Q > 0$ such that for all $\theta \in \Theta$ and all $x \in \mathcal{X}$, the per-sample loss $l_\theta(x)$ satisfies $|l_\theta(x)| \le Q$.*

Assumption 3.6 (Time discretization)**.** *For the inference(deployment) stage, let $\Delta t = 1/L$ be the uniform step size, and define discrete times $t_l = l\Delta t$ for $l = 0, 1, \cdots, L$. The numerical scheme for forward integration is*

$$x_{l+1} = x_l + \Delta t \cdot u_{1,\theta_1}(x_l, t_l) + \frac{(\Delta t)^2}{2} u_{2,\theta_2}(u_{1,\theta_1}(x_l, t_l), x_l, t_l).$$

3.3 Flow Matching and Rectified Flow

Next, we describe the general framework of flow matching and its second-order rectification. These concepts form the basis for our proposed method, as they integrate first and second-order information for trajectory estimation.

Definition 3.7 (Easy error)**.** *Let*

$$c_1(t) := \dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}, \quad c_2(t) := \ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}.$$

Fact 3.8. *Let a field x_t be defined as*

$$x_t = \alpha_t x_0 + \beta_t x_1,$$

where α_t and β_t are functions of t, and x_0, x_1 are constants. Then, the first-order gradient $\dot{x}_t$ and the second-order gradient $\ddot{x}_t$ can be manually calculated as

$$\dot{x}_t = \dot{\alpha}_t x_0 + \dot{\beta}_t x_1,$$
$$\ddot{x}_t = \ddot{\alpha}_t x_0 + \ddot{\beta}_t x_1.$$

Definition 3.9 (A variant of flow matching in [LCBH+22])**.** *Given two distributions μ_0 and π_0 on $\mathbb{R}^d$, flow matching aims to learn a time-dependent velocity field*

$$v_\theta : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$$

such that for any trajectory x_t transporting $x_0 \sim \mu_0$ to $x_1 \sim \pi_0$, we have

$$\dot{x}_t \sim v_\theta(x_t,\, t).$$

Remark 3.10. *In practice, one often samples (x_0, x_1) from (μ_0, π_0) and parameterizes x_t (e.g. via interpolation) at intermediate times to build a training objective that matches the velocity field to the true time derivative $\dot{x}_t$.*

Definition 3.11 (Second-order flow matching and loss)**.** *We additionally learn an acceleration field*

$$u_{2,\theta_2}(v, x, t), \quad \text{where} \quad v = u_{1,\theta_1}(x, t),$$

to approximate $\ddot{x}_t$. Hence, the two-part (second-order) loss is:

$$L_{2\text{nd}}(\theta_1, \theta_2) = \underbrace{\mathbb{E}[\|\dot{x}_t^{\text{true}} - u_{1,\theta_1}(x_t, t)\|_2^2]}_{L_{2,1,\theta_1}} + \underbrace{\mathbb{E}[\|\ddot{x}_t^{\text{true}} - u_{2,\theta_2}(u_{1,\theta_1}(x_t, t), x_t, t)\|_2^2]}_{L_{2,2,\theta_2,\theta_1}}.$$

Here, $\dot{x}_t^{\text{true}}$ and $\ddot{x}_t^{\text{true}}$ are observed (or numerically approximated) true velocity and acceleration, while u_{1,θ_1} and u_{2,θ_2} are the networks to be trained.

Definition 3.12 (Trajectory and time parameterization). *Consider a continuous trajectory* $\{x_t\}_{t\in[0,1]} \in \mathbb{R}^d$ *connecting an initial distribution* μ_0 *to a target data distribution* π_0. *We assume* $x_0 \sim \mu_0$ *and* $x_1 \sim \pi_0$.

Definition 3.13 (First and second order flow). *A first-order rectified flow is characterized by a velocity field* $u_{1,\theta_1}(x,t)$ *approximating* $\dot{x}_t$. *A second-order rectified flow further involves an acceleration field* $u_{2,\theta_2}(v,x,t)$, *where* $v = u_{1,\theta_1}(x,t)$ *approximates* $\dot{x}_t$, *and* u_{2,θ_2} *approximates* $\ddot{x}_t$.

Definition 3.14 (Sobolev space). *For a domain* $\Omega \subset \mathbb{R}^d$, *the Sobolev space* $H^2(\Omega)$ *is defined as*

$$H^2(\Omega) = \{f \in L^2(\Omega) : D^\alpha f \in L^2(\Omega), \forall |\alpha| \leq 2\}.$$

We assume the trajectory $x_t(\omega)$ *or its corresponding fields lie in such spaces, ensuring sufficient smoothness.*

Definition 3.15 (Noisy data and noise proportion). *Let the training dataset* $X = \{x_i\}_{i=1}^N$ *be drawn from* π_0 *but corrupted by noise. We denote the noise proportion as* $\epsilon = N_{\text{noisy}}/N$. *A noisy sample can be modeled as*

$$x_i^{\text{noisy}} = x_i^{\text{clean}} + \eta_i,$$

where η_i *satisfies certain boundedness conditions.*

Definition 3.16 (Error). *As we assume in Assumption 3.6, we define the error in* H^2*-norm.*

$$e_k := \|x_k^{\text{est}} - x_k^{\text{true}}\|_{H^2(\Omega)}$$

Definition 3.17 (Second-order loss function). *The loss function for the second-order method contains two parts. We define the first part which is trying to using* $\dot{x}_t$ *in Fact 3.8,* x_t *and* t *to learn function* $u_{1,t}$, *thus the loss is*

$$L_{2,1,\theta_1} := \|\dot{x}_t - u_{1,\theta_1}(x_t,t)\|_2^2$$

Next, we define the second part, which is trying to use $\ddot{x}_t, u_{1,\theta_1}(x_t,t)$, x_t *and* t *to learn* u_{2,θ_2} *function, thus the loss is*

$$L_{2,2,\theta_2,\theta_1} := \|\ddot{x}_t - u_{2,\theta_2}(u_{1,\theta_1}(x_t,t),x_t,t)\|_2^2$$

Overall, the total loss is

$$L_{2,\theta} := L_{2,1,\theta_1} + L_{2,2,\theta_2,\theta_1}$$

Definition 3.18 (Empirical loss). *We define the empirical second-order loss as*

$$\widetilde{L}_{2,\theta} = \frac{1}{N}\sum_{i=1}^N l_\theta(x_i).$$

Definition 3.19 (Population loss). *We define the population second-order loss as*

$$L_{2,\theta} = \mathbb{E}[l_\theta(X)]$$

Algorithm 1 Our new second-order training process

procedure 2NDORDERFORWARD()
  for each iteration **do**
    Random sample x_0 and time t, with target x_1
    $x_t \leftarrow \alpha_t \cdot x_0 + \sqrt{1-\alpha_t^2} \cdot x_1$
    Compute gradient with respect to $L_{2,\theta}$ ▷ see Def. 3.17
  end for
  return u_1, u_2 ▷ Two network functions
end procedure

Algorithm 2 Our new second-order inference algorithm

procedure 2NDORDERINFERENCE(u_1, u_2)
  $x_0 \sim \mathcal{N}(0, 1)$
  Initial $x \leftarrow x_0$
  for t from 0 to 1 with step $\Delta t = 0.01$ **do**
    $x \leftarrow x + \Delta t \cdot u_1(x, t) + \frac{(\Delta t)^2}{2} \cdot u_2(u_1(x, t), x, t)$
  end for
  return x
end procedure

3.4 Proposed Method

In this section, we now summarize the second-order algorithms that arise from the definitions above. Due to the space limitation, we delay the original first-order algorithm and our new third algorithms in the appendix.

4 Our Result

In Section 4.1, we first present a classical elliptic regularity lemma. In Section 4.2, we then show how controlling the second-order loss ensures bounded estimation error in a stronger Sobolev norm, thereby revealing a key regularization effect. In Section 4.3, we derive an excess risk bound, demonstrating that our method generalizes well under finite-sample conditions. In Section 4.4, we further analyze a discrete propagation inequality under noise. In Section 4.5, we combine these insights in our main theorem, proving that the learned trajectory remains robust against noise and sampling limitations.

4.1 Elliptic Regularity

In this section, we introduce the first result, which is a classical result that characterizes the relationship between different Sobolev norms for sufficiently smooth functions.

Lemma 4.1 (Elliptic regularity in [Eva10]). *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with a sufficiently smooth boundary. Suppose $h : \Omega \to \mathbb{R}$ belongs to $L^2(\Omega)$, has weak derivatives up to second order in $L^2(\Omega)$ and satisfies appropriate boundary conditions. Then, there exists a constant $C_{\mathrm{reg}} > 0$, depending only on Ω and the boundary conditions, such that*

$$\|h\|_{H^2(\Omega)} \leq C_{\mathrm{reg}}(\|\nabla h\|_{L^2(\Omega)} + \|h\|_{L^2(\Omega)}).$$

The above result is fundamental in establishing norm equivalences in Sobolev spaces, which we will use to analyze the regularity of error terms in subsequent lemmas.

4.2 Regularization Effect

In this section, we now connect the second-order loss function with the Sobolev norm of the estimation error.

Lemma 4.2 (Regulazation effect)**.** *Let $\{(x_0, x_1, t)\}$ denote the sampling start point, endpoint, and time in the training set, and suppose the true trajectory $\ddot{x}_t \in H^2(\Omega)$, we consider*

$$L_{2,2,\theta_1,\theta_2} = \mathbb{E}[\|\ddot{x}_t^{\text{true}} - u_{2,\theta_2}(u_{1,\theta_1}(x_t^{\text{true}}, t), x_t^{\text{true}}, t)\|_2^2],$$

The second-order loss with respect to the true second derivative. Particularly, there exist $C_{\text{reg}} \in \mathbb{R}$ when $L_{2,2,\theta_1,\theta_2}$ is sufficiently small such that

$$\begin{aligned} &\|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{H^2(\Omega)} \\ &\leq C_{\text{reg}}(L_{2,2,\theta_1,\theta_2}^{1/2} + \|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{L^2(\Omega)}). \end{aligned}$$

Proof. First we let $h(\cdot) = \dot{x}_t^{\text{est}}(\cdot) - \dot{x}_t^{\text{true}}(\cdot)$, the problem depends on both t and x, we could it by $h(t, x)$. For clarity, we simply write $h(\cdot)$ and regard it as a function on Ω. Generally, one assumes $\dot{x}_t^{\text{est}}, \dot{x}_t^{\text{true}} \in H^2(\Omega)$ so that $h \in H^2(\Omega)$.

Applying Lemma 4.1 to $h(\cdot) = \dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}$, we have

$$\begin{aligned} &\|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{H^2(\Omega)} \\ &\leq C_{\text{reg}}(\|\nabla h\|_{L^2(\Omega)} + \|h\|_{L^2(\Omega)}). \end{aligned} \tag{1}$$

By the Definition of the loss function, a small $L_{2,2,\theta_1,\theta_2}$ implies

$$\|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{L^2(\Omega)} \lesssim L_{2,2,\theta_1,\theta_2}^{1/2} \tag{2}$$

Combining Eq.(1) and (2), we have

$$\begin{aligned} &\|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{H^2(\Omega)} \\ &\leq C_{\text{reg}}(L_{2,2,\theta_1,\theta_2}^{1/2} + \|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{L^2(\Omega)}). \end{aligned}$$

Thus, we complete the proof. □

The above lemma highlights the importance of small second-order loss: it guarantees that the estimation error in the stronger Sobolev norm $H^2(\Omega)$ remains controlled.

4.3 Excess Risk

In this section, we introduce the following result which bounds the difference between the empirical and population loss, demonstrating that our method generalizes well under finite-sample conditions.

Lemma 4.3 (Symmetrization bound)**.** *Let $\{x_i\}_{i=1}^N$ and $\{x_i'\}_{i=1}^N$ be i.i.d. samples. For $\mathcal{G} = \{\ell_\theta : \theta \in \Theta\}$, we have:*

$$\sup_{g \in \mathcal{G}} |\frac{1}{N} \sum_{i=1}^N (g(x_i) - g(x_i'))| \leq \frac{2}{N} \mathop{\mathbb{E}}_{\sigma}[\sup_{g \in \mathcal{G}} \sum_{i=1}^N \sigma_i g(x_i)],$$

where $\{\sigma_i\}_{i=1}^N$ are Rademacher random variables, $\sigma_i \in \{+1, -1\}$ with equal probability.

Proof. For each σ_i has a symmetric distribution, we have:

$$|\sum_{i=1}^{N}(g(x_i)-g(x_i'))| \leq \mathop{\mathbb{E}}_{\sigma}[|\sum_{i=1}^{N}\sigma_i(g(x_i)-g(x_i'))|$$

Taking the supremum over $g \in \mathcal{G}$ and noting that $\{x_i\}$ and $\{x_i'\}$ have the same distribution, we can split the expression inside the absolute value:

$$\begin{aligned}&\sup_{g\in\mathcal{G}}|\sum_{i=1}^{N}(g(x_i)-g(x_i'))|\\ \leq\ &\mathbb{E}_{\sigma}[\sup_{g\in\mathcal{G}}|\sum_{i=1}^{N}\sigma_i(g(x_i)-g(x_i'))|].\end{aligned}$$

By the triangle inequality, we get:

$$|\sum_{i=1}^{N}\sigma_i(g(x_i)-g(x_i'))| \leq |\sum_{i=1}^{N}\sigma_i g(x_i)| + |\sum_{i=1}^{N}\sigma_i g(x_i')|.$$

Hence,

$$\begin{aligned}&\sup_{g\in\mathcal{G}}|\sum_{i=1}^{N}(g(x_i)-g(x_i'))|\\ \leq\ &\mathbb{E}_{\sigma}[\sup_{g\in\mathcal{G}}|\sum_{i=1}^{N}\sigma_i\, g(x_i)| + \sup_{g\in\mathcal{G}}|\sum_{i=1}^{N}\sigma_i g(x_i')|].\end{aligned}$$

Because $\{x_i'\}$ is drawn from the same distribution as $\{x_i\}$, the two supremum terms have the same expected value. Therefore, we can combine them as follows:

$$\sup_{g\in\mathcal{G}}|\frac{1}{N}\sum_{i=1}^{N}(g(x_i)-g(x_i'))| \leq \frac{2}{N}\mathop{\mathbb{E}}_{\sigma}[\sup_{g\in\mathcal{G}}\sum_{i=1}^{N}\sigma_i g(x_i)],$$

Thu,s we complete the proof. □

Lemma 4.4 (Theorem 6.11 in [SSBD14])**.** *As we defined in Definition 3.14, 3.19 and 3.18, if Assumption 3.4 holds, for $g \in \mathcal{G}$ where $\mathcal{G} = \{\ell_\theta : \theta \in \Theta\}$, we have*

$$\sup_{g\in\mathcal{G}}|\frac{1}{N}\sum_{i=1}^{N}g(x_i') - \mathbb{E}[g(x)]| \leq O(\sqrt{\ln(1/\beta)/N})$$

Lemma 4.5 (Excess risk)**.** *As we defined in Definition 3.19, we have*

$$\begin{aligned}&\widetilde{L}_{2,\theta}\\ =\ &\frac{1}{N}\sum_{i=1}^{N}(\|\dot{x}_t^{\text{true},i} - u_{1,\theta_1}(\cdot)\|_2^2 + \|\ddot{x}_t^{\text{true},i} - u_{2,\theta_2}(\cdot)\|_2^2)\end{aligned} \tag{3}$$

and

$$L_{2,\theta} = \mathbb{E}[\|\dot{x}_t^{\text{true}} - u_{1,\theta_1}(\cdot)\|_2^2 + \|\ddot{x}_t^{\text{true}} - u_{2,\theta_2}(\cdot)\|_2^2] \tag{4}$$

Suppose $\mathcal{F}_1$ nad $\mathcal{F}_2$ have finite or at most polynomially growing complexities $\mathcal{C}(\mathcal{F}_1), \mathcal{C}(\mathcal{F}_2)$. Then for $\beta \in (0,1)$, with probability at least $1-\beta$, we have

$$|\widetilde{L}_{2,\theta} - L_{2,\theta}| \leq O((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \ln(1/\beta))/N)^{1/2}.$$

Proof. Let $\mathcal{G} = \{\ell_\theta : \theta \in \Theta\}$ represent the complexity of $\mathcal{G}$ the Rademacher/VC dimension, As we defined in Definition 3.11, 3.18 and 3.19, we calculate the empirical loss and population loss,

$$\begin{aligned}
\widetilde{L}_{2,\theta} &= \frac{1}{N}\sum_{i=1}^N l_\theta(x_i) \\
&= \frac{1}{N}\sum_{i=1}^N (\|\dot{x}_t^{\text{true},i} - u_{1,\theta_1}(\cdot)\|_2^2 + \|\ddot{x}_t^{\text{true},i} - u_{2,\theta_2}(\cdot)\|_2^2)
\end{aligned}$$

and

$$\begin{aligned}
L_{2,\theta} &= \mathbb{E}[l_\theta(X)] \\
&= \mathbb{E}[\|\dot{x}_t^{\text{true}} - u_{1,\theta_1}(\cdot)\|_2^2 + \|\ddot{x}_t^{\text{true}} - u_{2,\theta_2}(\cdot)\|_2^2]
\end{aligned}$$

let $\{x_i'\}_{i=1}^N$ be an i.i.d. sample from the same distribution as $\{x_i\}_{i=1}^N$, and let $\{\sigma_i\}_{i=1}^N$ be i.i.d. Rademacher random variables ($\sigma_i \in \{+1, -1\}$ with probability $1/2$ each). Then, for any $g \in \mathcal{G}$, we have

$$\begin{aligned}
&\sup_{g\in\mathcal{G}} |\frac{1}{N}\sum_{i=1}^N g(x_i) - \mathbb{E}[g(x)]| \\
\leq &\sup_{g\in\mathcal{G}} |\frac{1}{N}\sum_{i=1}^N (g(x_i)) - g(x_i')| \\
&+ \sup_{g\in\mathcal{G}} |\frac{1}{N}\sum_{i=1}^N g(x_i') - \mathbb{E}[g(x)]|
\end{aligned} \tag{5}$$

We can upper bound the first term in Eq. (5),

$$\begin{aligned}
&\sup_{g\in\mathcal{G}} |\frac{1}{N}\sum_{i=1}^N (g(x_i)) - g(x_i')| \\
\leq &\frac{2}{N}\mathbb{E}_\sigma[\sup_{g\in\mathcal{G}}\sum_{i=1}^N \sigma_i g(x_i)] \\
= &\, 2\widetilde{\mathcal{R}}_N(\mathcal{G}) \\
\leq &\, 2\cdot O(\sqrt{C(\mathcal{G})/N}) \\
\leq &\, O(\sqrt{(C(\mathcal{F}_1) + C(\mathcal{F}_2))/N})
\end{aligned} \tag{6}$$

where the first step follows from Lemma 4.3, the second step comes from we define $\widetilde{\mathcal{R}}_N(G) := \mathbb{E}_\sigma[\sup_{g\in\mathcal{G}} \frac{1}{N}\sum_{i=1}^N \sigma_i g(x_i)]$, the third step follows from Assumption 3.4, the forth step follows from the definition of $\mathcal{G}$.

We can upper bound the second term in Eq. (5) by using Lemma 4.4,

$$\sup_{g\in\mathcal{G}} |\frac{1}{N}\sum_{i=1}^N g(x_i') - \mathbb{E}[g(x)]| \leq O(\sqrt{\ln(1/\beta)/N}) \tag{7}$$

Loading Eq. (6) and Eq. (7), we can obtain

$$\begin{aligned}&\sup_{g\in\mathcal{G}}|\frac{1}{N}\sum_{i=1}^{N}g(x_i)-\mathbb{E}[g(x)]|\\&\leq O(\sqrt{(C(\mathcal{F}_1)+C(\mathcal{F}_2)+\ln(1/\beta))/N})\end{aligned}$$

Thus, we complete the proof. □

4.4 Discrete Propagation under Noise

In this section, we show the lemma about discrete propagation under noise, which quantifies how noise in the trajectory affects the error propagation in a discrete setting.

Lemma 4.6 (Discrete propagation under noise)**.** *Suppose η_i satisfies $\|\eta_i\| \leq \delta$, there exist $C_{\text{prop}} \in \mathbb{R}$ such that*

$$e_{l+1} \leq (1 + \Delta t \cdot C_{\text{prop}})e_l + C_{\text{prop}} \cdot \Delta t \cdot \delta \cdot \epsilon$$

unrolling for $l = 0, \cdots, L-1$, we have

$$e_L \leq e_0 \exp(C_{\text{prop}}) + \frac{\delta \cdot \epsilon}{C_{\text{prop}}}(\exp(C_{\text{prop}}) - 1)$$

Proof. By Assumptions 3.2 and 3.6, the discrete updates for both the estimated and true systems can be written as:

$$\begin{aligned}x_{l+1} = x_l &+ \Delta t \cdot u_{1,\theta_1}(x_l, t_l)\\&+ \frac{(\Delta t)^2}{2}u_{2,\theta_2}(u_{1,\theta_1}(x_l, t_l), x_l, t_l).\end{aligned}$$

Subtracting the true system update from the estimated one gives:

$$\begin{aligned}&x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\\&= (x_l^{\text{est}} - x_l^{\text{true}}) + \Delta t \cdot (u_{1,\theta_1}(x_l^{\text{est}}, t_l) - \dot{x}_l^{\text{true}})\\&\quad + \frac{(\Delta t)^2}{2} \cdot (u_{2,\theta_2}(\cdot) - \ddot{x}^{\text{true}}).\end{aligned}$$

Taking the H^2-norm, we have:

$$\begin{aligned}&\|x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\|_{H^2(\Omega)}\\&\leq \|x_l^{\text{est}} - x_l^{\text{true}}\|_{H^2(\Omega)}\\&\quad + \Delta t \cdot \|u_{1,\theta_1}(x_l^{\text{est}}, t_l) - \dot{x}_l^{\text{true}}\|_{H^2(\Omega)}\\&\quad + O((\Delta t)^2).\end{aligned}$$

Since $\dot{x}_l^{\text{true}} \sim u_{1,\theta_1}(x_l^{\text{true}}, t_l)$ and the deviation is controlled by $\|x_l^{\text{est}} - x_l^{\text{true}}\|$ and noise $\delta\epsilon$, we can write:

$$\begin{aligned}e_{l+1} &= \|x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\|_{H^2(\Omega)}\\&\leq (1 + \Delta t \cdot C_{\text{prop}})e_l + C_{\text{prop}}\Delta t \cdot \delta \cdot \epsilon,\end{aligned}$$

where C_{prop} depends on the Lipschitz constants of u_{1,θ_1} and u_{2,θ_2}. Repeatedly applying this inequality from $l = 0$ to $l = L - 1$, we have:

$$e_L = e_0 \prod_{j=0}^{L-1} (1 + \Delta t \cdot C_{\text{prop}}) + \sum_{l=0}^{L-1} (C_{\text{prop}} \Delta t \cdot \delta \cdot \epsilon \prod_{j=l+1}^{L-1} (1 + \Delta t \cdot C_{\text{prop}})).$$

Recognizing that:

$$\prod_{j=0}^{L-1} (1 + \Delta t \cdot C_{\text{prop}}) = \exp(C_{\text{prop}}),$$

we simplify the summation term using the geometric series formula:

$$\sum_{l=0}^{L-1} \prod_{j=l+1}^{L-1} (1 + \Delta t \cdot C_{\text{prop}}) = \frac{\exp(C_{\text{prop}}) - 1}{C_{\text{prop}}}.$$

Thus, we obtain:

$$e_L \leq e_0 \exp(C_{\text{prop}}) + \frac{\delta \cdot \epsilon}{C_{\text{prop}}} (\exp(C_{\text{prop}}) - 1).$$

This completes the proof. □

This result provides a discrete Gronwall-type inequality that quantifies the growth of estimation error under bounded noise.

4.5 Main Result

In this section, we now state and prove our main result with the auxiliary lemmas in place, which establishes the robustness of the learned trajectory against noise and finite-sample effects.

Theorem 4.7 (Noise robustness)**.** [1] *Suppose all Assumption 3.1, 3.2, 3.3 and 3.6 holds, Let* $\widetilde{\theta} = (\widetilde{\theta}_1, \widetilde{\theta}_2)$ *is the approximately optimal solution, then* $\beta \in (0, 0.1)$*, the final-time(*$t = 1$*) trajectory estimate satisfies*

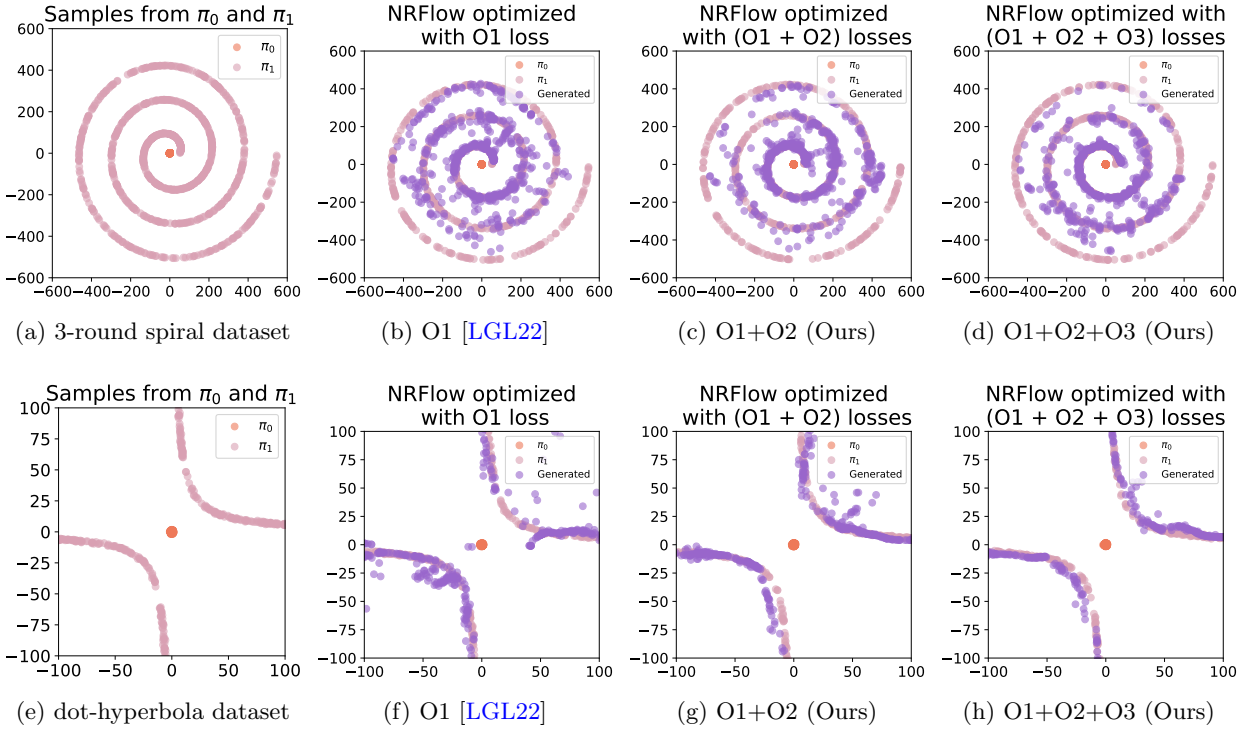
$$\begin{aligned} &\|x_{t=1}^{\text{est}} - x_{t=1}^{\text{true}}\|_{H^2(\Omega)} \\ &\leq C_1 \exp(C_2) \cdot (e_0 + \delta \cdot \epsilon) \\ &\quad + C_3 \cdot ((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \ln(1/\beta))/N)^{1/2}, \end{aligned}$$

where $e_0 = \|x_0^{\text{est}} - x_0^{\text{true}}\|$ *is the initial error.* C_1, C_2, C_3 *depends on Lipschitz constant* L*, dimension* d*, sobolev embedding constant,* Δt*,* $\exp(C_2)$ *represents the discrete gronwall factor for the time interval* $[0, 1]$*.*

[1] We state the proof of Theorem 4.7 in Section C in our Appendix.

5 Experiments

This section presents a series of experiments to evaluate the effectiveness of our NRFlow. Our results demonstrate that NRFlow significantly improves distribution generation, with the high-order loss playing a key role in enhancing model performance. In Section 5.1, we provide a detailed explanation for the setup of our experiments. In Section 5.2, we provide a comprehensive analysis of the effectiveness of our high-order supervision in our NRFlow model.



(a) 3-round spiral dataset (b) O1 [LGL22] (c) O1+O2 (Ours) (d) O1+O2+O3 (Ours)

(e) dot-hyperbola dataset (f) O1 [LGL22] (g) O1+O2 (Ours) (h) O1+O2+O3 (Ours)

Figure 1: **NRFlow on 3-round spiral dataset and dot-hyperbola dataset.** From left to right: the first column shows the 3-round spiral dataset and the dot-hyperbola dataset; the second column shows the results of NRFlow optimized with first-order loss (O1) [LGL22]; the third column shows the results of NRFlow optimized by first-order and second-order loss (O1+O2) (Ours); the fourth column are the results of NRFlow optimized by first-order, second-order and third-order loss (O1+O2+O3) (Ours). Our high-order NRFlows (third column and fourth column) show great capability in modeling complex distribution. Quantitative results are shown in Table 1.

5.1 Experiment Setup

We conduct comprehensive evaluations of NRFlow across diverse data distributions and multiple loss function combinations. Notably, the NRFlow implementation using only first-order loss (denoted as O1) corresponds exactly to the baseline Rectified Flow framework [LGL22]. Our proposed extensions consist of two configurations: second-order enhanced (O1+O2) and third-order augmented (O1+O2+O3) variants.

The evaluation employs two challenging synthetic datasets: a three-round spiral distribution and a dot-hyperbola distribution. Each dataset contains 100 sample points drawn from both source and target distributions. Our implementation utilizes a 2-layer fully connected network

Table 1: **Euclidean distance loss across three complex distribution datasets under the new trajectory setting.** Lower values indicate higher accuracy in distribution transfer. The optimal values are highlighted in **Bold**, and the second-best loss values (second-lowest) are represented by Underlined numbers for each dataset (row).

Loss terms	Five mode	3-round Spiral	Dot Hyperbola
O1 [LGL22]	1.755	17.338	18.096
O1 + O2 (Ours)	<u>0.956</u>	<u>15.514</u>	<u>3.823</u>
O1 + O2 + O3 (Ours)	**0.778**	**11.866**	**2.959**

with 100 hidden units, trained using the Adam optimizer with a learning rate of 0.005. For the three-round spiral dataset, we employ full-batch training (batch size 1000) over 1000 optimization steps. The dot-hyperbola configuration uses an increased batch size of 1600 to account for its greater geometric complexity. Numerical integration is performed using an adaptive ODE solver throughout all experiments.

5.2 Results Analysis

Our primary objective involves learning optimal transport trajectories between source distributions (depicted in orange in Figure 1) and target distributions (shown in pick). Empirical results demonstrate that the baseline Rectified Flow model (O1) exhibits significant limitations in target distribution modeling. As visualized in Figure 1 (second column), the first-order model generates substantial out-of-distribution artifacts for both synthetic datasets. This observation is quantitatively confirmed in Table 1, where O1 has the highest Euclidean distance metrics among all settings.

The introduction of second-order regularization (O1+O2) yields marked improvements. Final optimization with third-order constraints (O1+O2+O3) produces the most accurate distribution alignment, achieving near-perfect coverage of target domains. These results conclusively demonstrate that high-order supervision progressively enhances the model's ability to capture complex distributional geometries, with each additional regularization term contributing to statistically significant performance gains.

6 Conclusion

In summary, we introduced NRFlow, which augments traditional flow-based generative models by the second-order term. Our theoretical results demonstrate that these higher-order terms act as an effective regularizer, providing improved noise robustness and smoother trajectories under bounded perturbations. A discrete Gronwall analysis further shows that error propagation remains controlled, reinforcing the framework's stability. These findings highlight the promise of second-order methods for robust generative modeling.

References

[Ama98] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[AVE22] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[BASH+23] Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

[BCE+23] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[BGM22] Jimmy Ba, Roger Grosse, and James Martens. Distributed second-order optimization using kronecker-factored approximations. In *International conference on learning representations*, 2022.

[BHA+21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[BNX+23] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023.

[BRSR24] Vansh Bansal, Saptarshi Roy, Purnamrita Sarkar, and Alessandro Rinaldo. Straightness of rectified flow: A theoretical insight into wasserstein convergence. *arXiv preprint arXiv:2410.14949*, 2024.

[BSY23] Song Bian, Zhao Song, and Junze Yin. Federated empirical risk minimization via second-order method. *arXiv preprint arXiv:2305.17482*, 2023.

[Che23] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

[CHL+22] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[CHL+24] Yifang Chen, Jiayan Huo, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Fast gradient computation for rope attention in almost linear time. *arXiv preprint arXiv:2412.17316*, 2024.

[CL24] Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

[CLL+24a] Bo Chen, Xiaoyu Li, Yingyu Liang, Jiangxuan Long, Zhenmei Shi, and Zhao Song. Circuit complexity bounds for rope-based transformer architecture. *arXiv preprint arXiv:2411.07602*, 2024.

[CLL+24b] Bo Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Bypassing the exponential dependency: Looped transformers efficiently learn in-context by multi-step gradient descent. *arXiv preprint arXiv:2410.11268*, 2024.

[CLL+24c] Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. The computational limits of state-space models and mamba via the lens of circuit complexity. *arXiv preprint arXiv:2412.06148*, 2024.

[CLS+24] Bo Chen, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Hsr-enhanced sparse attention acceleration. *arXiv preprint arXiv:2410.10165*, 2024.

[CND+22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019.

[DMS23] Yichuan Deng, Sridhar Mahadevan, and Zhao Song. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. *arXiv preprint arXiv:2304.04397*, 2023.

[DSWY22] Yichuan Deng, Zhao Song, Yitan Wang, and Yuanyuan Yang. A nearly optimal size coreset algorithm with nearly linear time. *arXiv preprint arXiv:2210.08361*, 2022.

[DWB+23] Maximilian Dax, Jonas Wildberger, Simon Buchholz, Stephen R Green, Jakob H Macke, and Bernhard Scholkopf. Flow matching for scalable simulation-based inference. *arXiv preprint arXiv:2305.17161*, 2023.

[EKB+24] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[Eva10] Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 2010.

[FMZZ24] Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11979–11987, 2024.

[GFC21a] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

[GFC21b] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021.

[GHZ+23] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

[GM16] Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2016.

[GMS23] Yeqi Gao, Sridhar Mahadevan, and Zhao Song. An over-parameterized exponential regression. *arXiv preprint arXiv:2303.16504*, 2023.

[GSWY23] Yeqi Gao, Zhao Song, Weixin Wang, and Junze Yin. A fast optimization view: Reformulating single layer attention in llm based on tensor and svm trick, and solving it in matrix multiplication time. *arXiv preprint arXiv:2309.07418*, 2023.

[GSY23a] Yeqi Gao, Zhao Song, and Junze Yin. Gradientcoin: A peer-to-peer decentralized large language models. *arXiv preprint arXiv:2308.10502*, 2023.

[GSY23b] Yeqi Gao, Zhao Song, and Junze Yin. An iterative algorithm for rescaled hyperbolic functions regression. *arXiv preprint arXiv:2305.00660*, 2023.

[HG24] Tiankai Hang and Shuyang Gu. Improved noise schedule for diffusion training. *arXiv preprint arXiv:2407.03297*, 2024.

[HPPA24] Doron Haviv, Aram-Alexandre Pooladian, Dana Pe'er, and Brandon Amos. Wasserstein flow matching: Generative modeling over families of distributions. *arXiv preprint arXiv:2411.00698*, 2024.

[HWA+24] Vincent Hu, Di Wu, Yuki Asano, Pascal Mettes, Basura Fernando, Björn Ommer, and Cees Snoek. Flow matching for conditional text generation in a few sampling steps. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 380–392, 2024.

[HWSL24] Jerry Yao-Chieh Hu, Weimin Wu, Zhao Song, and Han Liu. On statistical rates and provably efficient criteria of latent diffusion transformers (dits). *arXiv preprint arXiv:2407.01079*, 2024.

[HysW+22] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[JZXG25] Yitong Jiang, Zhaoyang Zhang, Tianfan Xue, and Jinwei Gu. Autodir: Automatic all-in-one image restoration with latent diffusion. In *European Conference on Computer Vision*, pages 340–359. Springer, 2025.

[Kan58] Leonid Kantorovitch. On the translocation of masses. *Management science*, 5(1):1–4, 1958.

[KKN24] Leon Klein, Andreas Kramer, and Frank Noe. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36, 2024.

[KLL+24] Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Advancing the understanding of fixed point iterations in deep neural networks: A detailed analytical study. *arXiv preprint arXiv:2410.11279*, 2024.

[KLL+25] Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. On computational limits and provably efficient criteria of visual autoregressive models: A fine-grained complexity analysis. *arXiv preprint arXiv:2501.04377*, 2025.

[LARC21] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.

[LCBH+22] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[LGL22] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[LHH+24] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[LKW+24] Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[LL21] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2021.

[LLLY24] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5404–5411, 2024.

[LLS+24] Yingyu Liang, Jiangxuan Long, Zhenmei Shi, Zhao Song, and Yufa Zhou. Beyond linear approximations: A novel pruning approach for attention matrix, 2024.

[LLSZ24] Xiaoyu Li, Jiangxuan Long, Zhao Song, and Tianyi Zhou. Fast second-order method for neural network under small treewidth setting. In *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024.

[LSS+24a] Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Looped relu mlps may be all you need as practical programmable computers. *arXiv preprint arXiv:2410.09375*, 2024.

[LSS+24b] Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Multi-layer transformers gradient can be approximated in almost linear time. *arXiv preprint arXiv:2408.13233*, 2024.

[LSSS24] Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Differential privacy mechanisms in neural tangent kernel regression. *arXiv preprint arXiv:2407.13621*, 2024.

[LYHZ24] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[LZB+22] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[M+10] James Martens et al. Deep learning via hessian-free optimization. In *Icml*, volume 27, pages 735–742, 2010.

[McC97] Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128(1):153–179, 1997.

[MKBH22] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[Mon81] G. Monge. *Memoire sur la théorie des déblais et des remblais.* Imprimerie royale, 1781.

[NAA+21] Maxwell Nye, Anders Johan Andreassen, Gur AriGuy, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.

[Ope23] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[Ope24] OpenAI. Introducing ChatGPT, 2024.

[OWJ+22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022.

[PBHDE+23] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

[PX23] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[QSS23] Lianke Qin, Zhao Song, and Baocheng Sun. Is solving graph neural tangent kernel equivalent to training graph neural network? *arXiv preprint arXiv:2309.07452*, 2023.

[RBL+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[RCK+24] Litu Rout, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Beyond first-order tweedie: Solving inverse problems using latent diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9472–9481, 2024.

[SCL+23] Zhenmei Shi, Jiefeng Chen, Kunyang Li, Jayaram Raghuram, Xi Wu, Yingyu Liang, and Somesh Jha. The trade-off between universality and label efficiency of representations from contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[SCN+23] Neta Shaul, Ricky TQ Chen, Maximilian Nickel, Matthew Le, and Yaron Lipman. On kinetic optimal probability paths for generative models. In *International Conference on Machine Learning*, pages 30883–30907. PMLR, 2023.

[SE19] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[SSQ+22] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*. PMLR, 2022.

[SSX23] Anshumali Shrivastava, Zhao Song, and Zhaozhuo Xu. A theoretical analysis of nearest neighbor search on approximate near neighbor graph. *arXiv preprint arXiv:2303.06210*, 2023.

[SSZ+24a] Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, Yanyu Li, Yifan Gong, Kai Zhang, Hao Tan, Jason Kuen, Henghui Ding, et al. Lazydit: Lazy learning for the acceleration of diffusion transformers. *arXiv preprint arXiv:2412.12444*, 2024.

[SSZ+24b] Xuan Shen, Zhao Song, Yufa Zhou, Bo Chen, Jing Liu, Ruiyi Zhang, Ryan A Rossi, Hao Tan, Tong Yu, Xiang Chen, et al. Numerical pruning for efficient autoregressive models. *arXiv preprint arXiv:2412.12441*, 2024.

[SWY23] Zhao Song, Weixin Wang, and Junze Yin. A unified scheme of resnet and softmax. *arXiv preprint arXiv:2309.13482*, 2023.

[TFM+23] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

[TLI+23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Vil09] Cedric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[VONR+23] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*. PMLR, 2023.

[VP12] Oriol Vinyals and Daniel Povey. Krylov subspace descent for deep learning. In *Artificial intelligence and statistics*, pages 1261–1268. PMLR, 2012.

[VSP+17] 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*, 2017.

[WCZ+23] Yilin Wang, Zeyuan Chen, Liangjun Zhong, Zheng Ding, Zhizhou Sha, and Zhuowen Tu. Dolfin: Diffusion layout transformers without autoencoder. *arXiv preprint arXiv:2310.16305*, 2023.

[WET+24] Xiaofei Wang, Sefik Emre Eskimez, Manthan Thakker, Hemin Yang, Zirun Zhu, Min Tang, Yufei Xia, Jinzhu Li, Sheng Zhao, Jinyu Li, et al. An investigation of noise robustness for flow-matching-based zero-shot tts. *arXiv preprint arXiv:2406.05699*, 2024.

[WHL+23] Jerry Wei, Le Hou, Andrew Kyle Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc V Le. Symbol tuning improves in-context learning in language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[WSD+23] Zirui Wang, Zhizhou Sha, Zheng Ding, Yilin Wang, and Zhuowen Tu. Tokencompose: Grounding diffusion with token-level supervision. *arXiv preprint arXiv:2312.03626*, 2023.

[WXZ+24] Yilin Wang, Haiyang Xu, Xiang Zhang, Zeyuan Chen, Zhizhou Sha, Zirui Wang, and Zhuowen Tu. Omnicontrolnet: Dual-stage integration for conditional image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7436–7448, 2024.

[XSW+23] Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Yin Li, and Yingyu Liang. Improving foundation models for few-shot learning via multitask finetuning. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[XSW+24] Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. Towards few-shot adaptation of foundation models via multitask finetuning. In *The Twelfth International Conference on Learning Representations*, 2024.

[XZC+22] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8121–8130, 2022.

[ZHZ+23] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

[ZLX+23] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[ZWF+21] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*. PMLR, 2021.

NRFlow: Towards Noise-Robust Generative Modeling via High-Order Flow Matching (Supplementary Material)

Roadmap. In Section A, we introduce some notations and basic concepts. In Section B, we introduce more related work that inspires our work. We state the proof of Theorem 4.7 in Section C. In Section D, we extend our result to a third-order case. In Section E, we extend our result to k-th order. In Section F, we provide comprehensive experiments to evaluate our NRFlow under complex conditions.

A Preliminary

In Section A.1, we introduce some notations we use in the appendix. In Section A.2, we introduce some basic concepts about flow matching. In Section A.3, we introduce the background of optimal transport.

A.1 Notations

We use Pr[] to denote the probability. We use $\mathbb{E}[]$ to denote the expectation. We use Var[] to denote the variance. We use $\|x\|_p$ to denote the ℓ_p norm of a vector $x \in \mathbb{R}^n$, i.e. $\|x\|_1 := \sum_{i=1}^n |x_i|$, $\|x\|_2 := (\sum_{i=1}^n x_i^2)^{1/2}$, and $\|x\|_\infty := \max_{i\in[n]} |x_i|$. For variables a, b, We write $a \lesssim b$ to indicate that a is bounded above by b up to a multiplicative constant independent of the main parameters. We write $a \gtrsim b$ to indicate that a is bounded below by b up to a multiplicative constant independent of the main parameters. We denote $\dot{x}^{(k)}$ as the k-th order derivative field of x. We use Dist as the function represents the probability distribution of a given random variable or random vector, mapping it to its corresponding measure on the probability space.

A.2 Flow Matching

In this section, we restate and introduce some definitions of flow matching and the algorithm. We restate part of Definition 3.17 and introduce the loss function of flow matching.

Definition A.1 (Loss function)**.** *The loss function for the second order method contains two parts. We define the first part which is trying to using $\dot{x}_t$ in Fact 3.8, x_t and t to learn function $u_{1,t}$, thus the loss is*

$$L_{1\mathrm{st}} := \|\dot{x}_t - u_{1,\theta_1}(x_t, t)\|_2^2.$$

Here we restate Definition 3.9

Definition A.2 (a variant of flow matching in [LCBH+22])**.** *Given two distributions μ_0 and π_0 on $\mathbb{R}^d$, flow matching aims to learn a time-dependent velocity field*

$$v_\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$$

such that for any trajectory x_t transporting $x_0 \sim \mu_0$ to $x_1 \sim \pi_0$, we have

$$\dot{x}_t \sim v_\theta(x_t, t).$$

We present the training algorithm and inference algorithm of flow matching.

Algorithm 3 Training algorithm of flow matching

procedure 1stOrderForward()
  for each iteration **do**
    Random sample x_0 and time t, with target x_1
    $x_t \leftarrow \alpha_t \cdot x_0 + \sqrt{1-\alpha_t^2} \cdot x_1$
    Compute gradient with respect to $L_{1\text{st}}$ ▷ See Definition A.1
  end for
  return u_1 ▷ One network functions
end procedure

Algorithm 4 Inference algorithm of flow matching

procedure 1stOrderInference(u_1)
  $x_0 \sim \mathcal{N}(0, 1)$
  Initial $x \leftarrow x_0$
  for t from 0 to 1 with step $\Delta t = 0.01$ **do**
    $x \leftarrow x + \Delta t \cdot u_1(x, t)$
  end for
  return x
end procedure

A.3 Optimal Transport

In this section, we introduce some background of optimal transport.

The optimal transport (OT) problem, as originally framed by Monge [Mon81], seeks to minimize a cost functional:

$$\inf_{\mathcal{T}} \mathbb{E}[c(\mathcal{T}(x_0) - x_0)],$$
$$\text{s.t. } \text{Dist}(\mathcal{T}(x_0)) = \pi_0, \quad \text{Dist}(x_0) = \mu_0,$$

where the optimization is over deterministic mappings $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$ that define a coupling (x_0, x_1) with $x_1 = \mathcal{T}(x_0)$, minimizing the cost c [Vil09].

Kantorovich [Kan58] extended Monge's problem by introducing the Monge-Kantorovich (MK) formulation, which allows for both deterministic and stochastic couplings (x_0, x_1) with marginal distributions μ_0 and π_0. Notably, when μ_0 is absolutely continuous with respect to the Lebesgue measure, the optimal coupling remains deterministic, reducing the problem to the set of mappings $\mathcal{T}$. This equivalence facilitates a dynamic interpretation, where the aim is to identify a continuous-time trajectory $\{x_t\}_{t\in[0,1]}$ from a collection of smooth interpolants $\mathcal{X}$, such that $x_0 \sim \mu_0$ and $x_1 \sim \pi_0$. For a convex cost function c, Jensen's inequality implies:

$$\mathbb{E}[c(x_1 - x_0)] \geq \inf_{\{x_t\}_{t\in[0,1]}\in\mathcal{X}} \mathbb{E}\left[\int_0^1 c(\dot{x}_t)\mathrm{d}t\right].$$

The infimum is achieved when x_t follows the displacement interpolant, $x_t = tx_1 + (1-t)x_0$, representing a geodesic in the Wasserstein space [McC97].

When the process is governed by ordinary differential equations (ODEs) of the form $\mathrm{d}x_t = v_t(x_t)\mathrm{d}t$, the evolution of the Lebesgue density ϵ_t of x_t satisfies the continuity equation:

$$\frac{\partial \epsilon_t}{\partial t} + \nabla \cdot (v_t \epsilon_t) = 0.$$

The Monge problem can then be reformulated dynamically as:

$$\inf_{\{v_t\}_{t\in[0,1]},\{x_t\}_{t\in[0,1]}} \mathbb{E}\left[\int_0^1 c(v_t(x_t))\mathrm{d}t\right],$$
$$\text{s.t. } \frac{\partial \epsilon_t}{\partial t} + \nabla\cdot(v_t\epsilon_t) = 0,$$
$$\mu_0 = \frac{\mathrm{d}\mu_0}{\mathrm{d}\lambda}, \quad \pi_0 = \frac{\mathrm{d}\pi_0}{\mathrm{d}\lambda}.$$

Although this dynamic formulation provides deeper insights, solving it is computationally challenging. For cost functions like the ℓ_2 norm, this reduces to minimizing the kinetic energy of the flow, as shown by [SCN+23], where displacement interpolants are energy-optimal and correspond to straight-line flow paths.

B More related work

In this section, we discuss more related work which inspires our work.

Large Language Models. Neural networks built upon the Transformer architecture [VSP+17] have swiftly risen to dominate modern machine learning approaches in natural language processing. Extensive Transformer models, trained on wide-ranging and voluminous datasets while encompassing billions of parameters, are often termed large language models (LLM) or foundation models [BHA+21]. Representative instances include BERT [DCLT19], PaLM [CND+22], Llama [TLI+23], ChatGPT [Ope24], GPT4 [Ope23], among others. These LLMs have showcased striking general intelligence abilities [BCE+23] in various downstream tasks. Numerous adaptation methods have been developed to tailor LLMs for specific applications, such as adapters [HysW+22, ZHZ+23, GHZ+23, SCL+23], calibration schemes [ZWF+21, ZLX+23], multitask fine-tuning [GFC21a, XSW+23, VONR+23, XSW+24], prompt optimization [GFC21b, LARC21], scratchpad approaches [NAA+21], instruction tuning [LL21, CHL+22, MKBH22], symbol tuning [WHL+23], black-box tuning [SSQ+22], and reinforcement learning from human feedback (RLHF) [OWJ+22]. Additional lines of research endeavor to boost model efficiency without sacrificing performance across diverse domains, for example, in [LLS+24, LLSZ24, CLL+24c, CHL+24, KLL+24].

Diffusion Models. Diffusion Models have garnered significant attention for their capability to generate high-fidelity images by incrementally refining noisy samples, as exemplified by DiT [PX23] and U-ViT [BNX+23]. These approaches typically involve a forward process that systematically adds noise to an initial clean image and a corresponding reverse process that learns to remove noise step by step, thereby recovering the underlying data distribution in a probabilistic manner. Early works [SE19, SME20] established the theoretical foundations of this denoising strategy, introducing score-matching and continuous-time diffusion frameworks that significantly improved sample quality and diversity. Subsequent research has focused on more efficient training and sampling procedures [LZB+22, SSZ+24a, SSZ+24b], aiming to reduce computational overhead and converge faster without sacrificing image fidelity. Other lines of work leverage latent spaces to learn compressed representations, thereby streamlining both training and inference [RBL+22, HWSL24]. This latent learning approach integrates naturally with modern neural architectures and can be extended to various modalities beyond images, showcasing the versatility of diffusion processes in modeling complex data distributions. In parallel, recent researchers have also explored multi-scale noise scheduling and adaptive step-size strategies to enhance convergence stability and maintain high-resolution detail in generated content in [LKW+24, FMZZ24, RCK+24, JZXG25, LYHZ24]

C Missing proof of Theorem 4.7

Here, we state the proof of Theorem 4.7.

Proof. Let $\widetilde{\theta} = (\widetilde{\theta}_1, \widetilde{\theta}_2)$ denote the approximate optimal solution for the estimated loss in Eq. (3). By Lemma 4.5, we have

$$|\widetilde{L}_{2,\theta} - L_{2,\theta}| \leq O((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \ln(1/\beta))/N)^{1/2}.$$

Therefore, under the true distribution, $\dot{x}_t^{\text{est}}$ and $\ddot{x}_t^{\text{est}}$ approximate $\dot{x}_t^{\text{true}}$ and $\ddot{x}_t^{\text{true}}$ well in an L^2 sense.

As we defined $\ddot{x}_t^{\text{true}} \in H^2(\Omega)$,we then apply Lemma 4.2, which leverages the Assumption 3.1. If $L_{2,2,\theta_1,\theta_2}$ is small, there exist C_{reg} such that

$$\begin{aligned} &\|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{H^2(\Omega)} \\ &\leq C_{\text{reg}}(L_{2,2,\theta_1,\theta_2}^{1/2} + \|\dot{x}_t^{\text{est}} - \dot{x}_t^{\text{true}}\|_{L^2(\Omega)}). \end{aligned}$$

Since Lemma 4.5 already guarantees that $\dot{x}_t^{\text{est}}$ and $\ddot{x}_t^{\text{est}}$ are close to the true $\dot{x}_t^{\text{true}}$ and $\ddot{x}_t^{\text{true}}$ in L^2, we conclude that the learned fields are also close in the stronger $H^2(\Omega)$ norm. As we assumed in Assumption 3.3 and 3.6, For or uniform time steps $\Delta t = 1/L$, the update for the estimate is

$$\begin{aligned} x_{l+1}^{\text{est}} = x_l^{\text{est}} &+ \Delta t \cdot u_{1,\widetilde{\theta}_1}(x_l^{\text{est}}, t_l) \\ &+ \frac{(\Delta t)^2}{2} u_{2,\widetilde{\theta}_2}(u_{1,\widetilde{\theta}_1}(x_l^{\text{est}}, t_l), x_l^{\text{est}}, t_l). \end{aligned}$$

Similarly, the true trajectory x_{l+1}^{true} follows the same scheme but with the true velocity and acceleration (plus noise bounded by $\delta\epsilon$). Subtracting two updates and taking the H^2-norm, and invoking the Lipschitz condition in Assumption 3.2, yields Lemma 4.6, we have

$$\begin{aligned} e_{l+1} &= \|x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\|_{H^2(\Omega)} \\ &\leq (1 + \Delta t \cdot C_{\text{prop}}) e_l + C_{\text{prop}} \Delta t \cdot \delta \cdot \epsilon. \end{aligned}$$

Here C_{prop} depends on Lipschitz constants and bounds on $\dot{x}$ and $\ddot{x}$. Iterating the above and a discrete Gronwell argument shows

$$\begin{aligned} e_L &= \|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^2(\Omega)} \\ &\leq e_0 \exp(C_{\text{prop}}) + \frac{\delta \cdot \epsilon}{C_{\text{prop}}} (\exp(C_{\text{prop}}) - 1). \end{aligned}$$

Since $\exp(C_{\text{prop}})$ is just a constant factor, denote it by e^{C_2}. Combine all of these terms then yields the exact form of Theorem 4.7:

$$\begin{aligned} &\|x_{t=1}^{\text{est}} - x_{t=1}^{\text{true}}\|_{H^2(\Omega)} \\ &\leq C_1 \exp(C_2) \cdot (e_0 + \delta \cdot \epsilon) \\ &\quad + C_3 \cdot ((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \ln(1/\beta))/N)^{1/2}. \end{aligned}$$

Thus, we complete the proof. □

D Extension on third-order Flow Matching

In this section, we extend the second-order flow-matching framework in Section 3.3 to incorporate third-order information. We first introduce additional assumptions in Section D.1 to ensure that the third derivative of the true trajectory is sufficiently smooth and bounded. In Section D.2, we introduce our third-order training algorithm and the inference algorithm. In Section D.3, we introduce the elliptic regularity for third-order cases. In Section D.4, we present the result of the regularization effect result for the third-order loss function. In Section D.5, we show the excess risk of third-order cases. In Section D.6, we show the lemma about discrete propagation under noise quantifies how noise in the trajectory affects the error propagation in a third-order discrete setting. In Section D.7, we prove our third-order main result.

D.1 Preliminary

In this section, we introduce some additional definitions and assumptions specific to the third-order extension.

Assumption D.1 (smoothness in higher Sobolev spaces)**.** *We assume $x_t^{\text{true}} \in H^3(\Omega)$, its derivatives up tothe third order lie in $L^2(\Omega)$ and satisfy suitable boundary conditions.*

$$\|\dot{x}_t^{\text{true}}\|_{H^3(\Omega)} \le M_1, \quad \|\ddot{x}_t^{\text{true}}\|_{H^3(\Omega)} \le M_2, \quad \|\dddot{x}_t^{\text{true}}\|_{H^3(\Omega)} \le M_3.$$

In addition, we assume the third derivative $\dddot{x}_t^{\text{true}}$ is continuous over $[0,1]$ and satisfies

$$\|\dddot{x}_t^{\text{true}}\|_\infty \le M_3.$$

The assumption above is critical to ensure the trajectory has sufficient regularity for third-order analysis.

Remark D.2. *Assumption D.1 extends Assumption 3.1 by requiring a bounded third derivative and ensuring the entire trajectory has appropriate regularity in Sobolev space $H^3(\Omega)$. This added smoothness is essential for deriving higher-order error bounds.*

We now define the discrete-time update rule for third-order systems.

Assumption D.3 (time discretization for third-order update)**.** *Let $\Delta t = 1/L$ be the uniform step size, and define discrete times $t_l = l\Delta t$ for $l = 0, 1, \dots, L$. The third-order discrete update for the estimated system is:*

$$x_{l+1}^{\text{est}} = x_l^{\text{est}} + \Delta t u_{1,\theta_1}(x_l^{\text{est}}, t_l) + \frac{(\Delta t)^2}{2} u_{2,\theta_2}(u_{1,\theta_1}(x_l^{\text{est}}, t_l), x_l^{\text{est}}, t_l) + \frac{(\Delta t)^3}{6} u_{3,\theta_3}(u_{2,\theta_2}(\cdot), x_l^{\text{est}}, t_l).$$

The discrete update incorporates terms up to the third derivative, capturing the dynamics more accurately.

Assumption D.4. *The learned fields $u_{1,\theta_1}(x,t)$, $u_{2,\theta_2}(v,x,t)$ and $u_{3,\theta_3}(a,x,t)$ are L-Lipschitz continuous in spatial and temporal arguments. Formally, there exists $L > 0$ such that for all $x, y \in \mathbb{R}^d$ and $t, s \in [0,1]$:*

$$\begin{aligned}
\|u_{1,\theta_1}(x,t) - u_{1,\theta_1}(y,t)\|_2 &\le L\|x-y\|_2, \\
\|u_{2,\theta_2}(v,x,t) - u_{2,\theta_2}(v,y,t)\|_2 &\le L\|x-y\|_2, \\
\|u_{3,\theta_3}(a,x,t) - u_{3,\theta_3}(a,y,t)\|_2 &\le L\|x-y\|_2,
\end{aligned}$$

This is the natural extension of the second-order scheme in Assumption 3.6.

This assumption is necessary to control the propagation of errors through the system.

Definition D.5 (third-order rectified flow)**.** *A* third-order rectified flow *is determined by three learned fields:*

$$\begin{aligned} &u_{1,\theta_1}(x,t) \\ &u_{2,\theta_2}(v,x,t) \text{ where } \quad v = u_{1,\theta_1}(x,t), \\ &u_{3,\theta_3}(a,x,t) \text{ where } \quad a = u_{2,\theta_2}(v,x,t). \end{aligned}$$

These fields aim to approximate $\dot{x}_t^{\text{true}}$*,* $\ddot{x}_t^{\text{true}}$*, and* $\dddot{x}_t^{\text{true}}$*, respectively.*

We now introduce the third-order analog of the velocity and acceleration fields. In addition to the velocity u_{1,θ_1} and acceleration u_{2,θ_2} fields, we define a field

$$u_{3,\theta_3}(a,x,t),$$

where $a = u_{2,\theta_2}(v,x,t)$ and $v = u_{1,\theta_1}(x,t)$. This field aims to approximate the third derivative $\dddot{x}_t^{\text{true}}$.

Here, we introduce the definition of the field of third-order flow.

Definition D.6 (third-order flow field)**.** *A third-order rectified flow is characterized by a velocity field* $u_{1,\theta_1}(x,t)$*, an acceleration field* $u_{2,\theta_2}(v,x,t)$*, and a field*

$$u_{3,\theta_3}(a,x,t),$$

where

$$v = u_{1,\theta_1}(x,t), \quad a = u_{2,\theta_2}(u_{1,\theta_1}(x,t),x,t).$$

The function u_{3,θ_3} *aims to approximate the* $\dddot{x}_t^{\text{true}}$*.*

And we present the loss function of third-order flow as follows.

Definition D.7 (third-order loss function)**.** *Let* $\dot{x}_t^{\text{true}}$*,* $\ddot{x}_t^{\text{true}}$*, and* $\dddot{x}_t^{\text{true}}$ *be the true velocity, acceleration, and of the trajectory* x_t^{true}*. We define the third-order loss as*

$$L_{3\text{rd}}(\theta_1,\theta_2,\theta_3) = \underbrace{\mathbb{E}[\|\dot{x}_t^{\text{true}} - u_{1,\theta_1}(x_t,t)\|_2^2]}_{L_{3,1,\theta_1}} + \underbrace{\mathbb{E}[\|\ddot{x}_t^{\text{true}} - u_{2,\theta_2}(u_{1,\theta_1}(x_t,t),x_t,t)\|_2^2]}_{L_{3,2,\theta_2,\theta_1}} + \underbrace{\mathbb{E}[\|\dddot{x}_t^{\text{true}} - u_{3,\theta_3}(u_{2,\theta_2}(\cdot),x_t,t)\|_2^2]}_{L_{3,3,\theta_3,\theta_2,\theta_1}}$$

where each expectation is taken over the possibly noisy samples of the continuous trajectory x_t^{true}*.*

Here's the empirical third-order loss.

Definition D.8 (empirical third-order loss)**.** *Given a training dataset* $\{(x_0^i,x_1^i)\}_{i=1}^N$ *and time samples* $\{t_i\}$*, we define the empirical third-order loss:*

$$\widetilde{L}_{3\text{rd}} = \frac{1}{N}\sum_{i=1}^{N}[\|\dot{x}_t^{\text{true},i} - u_{1,\theta_1}(x_t^i,t_i)\|^2 + \|\ddot{x}_t^{\text{true},i} - u_{2,\theta_2}(u_{1,\theta_1}(x_t^i,t_i),x_t^i,t_i)\|^2 + \|\dddot{x}_t^{\text{true},i} - u_{3,\theta_3}(u_{2,\theta_2}(\cdot),x_t^i,t_i)\|^2].$$

D.2 Proposed Third-Order Algorithms

We present the natural extension of the second-order methods in Section 3.4 to incorporate the jerk term. Here are our third-order training algorithm and inference algorithm.

Algorithm 5 Our third-order training algorithm

procedure 3RDORDERFORWARD()
  for each iteration **do**
    Random sample x_0 and time t, with target x_1
    $x_t \leftarrow \alpha_t \cdot x_0 + \sqrt{1-\alpha_t^2} \cdot x_1$
    Compute gradient with respect to $L_{3\text{rd}}$ ▷ See Definition D.7
  end for
  return u_1, u_2, u_3 ▷ Three network functions
end procedure

Algorithm 6 Our third-order inference algorithm

procedure 3RDORDERINFERENCE(u_1, u_2, u_3)
  $x_0 \sim \mathcal{N}(0, 1)$
  Initial $x \leftarrow x_0$
  for t from 0 to 1 with step $\Delta t = 0.01$ **do**
    $x \leftarrow x + \Delta t \cdot u_1(x,t) + \frac{(\Delta t)^2}{2} \cdot u_2(u_1(x,t),x,t) + \frac{(\Delta t)^3}{6} \cdot u_3(u_2(u_1(x,t),x,t),x_t,t)$
  end for
  return x
end procedure

D.3 Elliptic Regularity

We now provide key lemmas and the main theorem establishing noise-robustness for third-order flow matching. In this section, we first introduce the elliptic regularity.

Lemma D.9 (Elliptic Regularity in [Eva10]). *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain witha smooth boundary. Suppose a function $h : \Omega \to \mathbb{R}$ has weak derivatives up to order 3 in $L^2(\Omega)$ and satisfies relevant boundary conditions. Then there exists a constant $C_{\text{reg},3} > 0$ (depending on Ω) such that*

$$\|h\|_{H^3(\Omega)} \leq C_{\text{reg},3}(\|\nabla^2 h\|_{L^2(\Omega)} + \|\nabla h\|_{L^2(\Omega)} + \|h\|_{L^2(\Omega)}).$$

D.4 Regularization Effect

In this section, we show the result of the regularization effect for the third-loss order.

Lemma D.10 (Regularization Effect for third-order Loss). *As we defined in Definition D.8, then we define*

$$L_{3\text{rd}} := \mathbb{E}[\| \dddot{x}_t^{\text{true}} - u_{3,\theta_3}(u_{2,\theta_2}(\cdot), x_t^{\text{true}}, t)\|^2].$$

If $\dddot{x}_t^{\text{true}} \in H^3(\Omega)$ and $L_{3\text{rd}}$ is sufficiently small, then there exists a constant $C_{\text{reg},3}$ such that

$$\|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{H^3(\Omega)} \leq C_{\text{reg},3}(L_{3\text{rd}}^{1/2} + \|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{L^2(\Omega)}).$$

Proof. Applying Lemma D.9 to $h(\cdot) = \ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}$, we have

$$\begin{aligned} &\|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{H^3(\Omega)} \\ &\leq C_{\text{reg},3}(\|\nabla^2 h\|_{L^2(\Omega)} + \|\nabla h\|_{L^2(\Omega)} + \|h\|_{L^2(\Omega)}). \end{aligned} \tag{8}$$

By the Definition of the loss function, a small $L_{2,2,\theta_1,\theta_2}$ implies

$$\|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{L^2(\Omega)} \lesssim L_{3\text{rd}}^{1/2} \tag{9}$$

Combining Eq.(8) and (9), we have

$$\begin{aligned} &\|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{H^3(\Omega)} \\ \leq\ & C_{\text{reg},3}(L_{3\text{rd}}^{1/2} + \|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{L^2(\Omega)}). \end{aligned}$$

Thus, we complete the proof. □

D.5 Excess Risk

In this section, we first introduce some necessary tools that need to be used in Lemma D.10. Then, we show our result of excess risk for third-order flow. First, we restate the symmetrization bound again.

Lemma D.11 (Symmetrization Bound, formal version of Lemma 4.3)**.** *Let $\{x_i\}_{i=1}^N$ and $\{x_i'\}_{i=1}^N$ be i.i.d. samples. For $\mathcal{G} = \{\ell_\theta : \theta \in \Theta\}$, we have:*

$$\sup_{g\in\mathcal{G}} \|\frac{1}{N}\sum_{i=1}^N (g(x_i) - g(x_i'))\| \leq \frac{2}{N}\mathop{\mathbb{E}}_{\sigma}[\sup_{g\in\mathcal{G}} \sum_{i=1}^N \sigma_i g(x_i)],$$

where $\{\sigma_i\}_{i=1}^N$ are Rademacher random variables, $\sigma_i \in \{+1, -1\}$ with equal probability.

Proof. For each σ_i has a symmetric distribution, we have:

$$\|\sum_{i=1}^N (g(x_i) - g(x_i'))\| \leq \mathop{\mathbb{E}}_{\sigma}[\|\sum_{i=1}^N \sigma_i(g(x_i) - g(x_i'))\|$$

Taking the supremum over $g \in \mathcal{G}$ and noting that $\{x_i\}$ and $\{x_i'\}$ have the same distribution, we can split the expression inside the absolute value:

$$\begin{aligned} &\sup_{g\in\mathcal{G}} \|\sum_{i=1}^N (g(x_i) - g(x_i'))\| \\ \leq\ & \mathbb{E}_\sigma[\sup_{g\in\mathcal{G}} \|\sum_{i=1}^N \sigma_i(g(x_i) - g(x_i'))\|]. \end{aligned}$$

By the triangle inequality, we get:

$$\|\sum_{i=1}^N \sigma_i(g(x_i) - g(x_i'))\| \leq \|\sum_{i=1}^N \sigma_i g(x_i)\| + \|\sum_{i=1}^N \sigma_i g(x_i')\|.$$

Hence,

$$\sup_{g\in\mathcal{G}} \|\sum_{i=1}^N (g(x_i) - g(x_i'))\|$$

$$\leq \mathbb{E}_\sigma[\sup_{g\in\mathcal{G}} \| \sum_{i=1}^N \sigma_i\, g(x_i)\| + \sup_{g\in\mathcal{G}} \| \sum_{i=1}^N \sigma_i g(x_i')].$$

Because $\{x_i'\}$ is drawn from the same distribution as $\{x_i\}$, the two supremum terms have the same expected value. Therefore, we can combine them as follows:

$$\sup_{g\in\mathcal{G}} \| \frac{1}{N}\sum_{i=1}^N (g(x_i) - g(x_i'))\| \leq \frac{2}{N}\mathop{\mathbb{E}}_{\sigma}[\sup_{g\in\mathcal{G}} \sum_{i=1}^N \sigma_i g(x_i)],$$

Thus, we complete the proof. □

Here we restate Lemma 4.4.

Lemma D.12 (formal version of Lemma 4.4)**.** *As we defined in Definition 3.14, D.7 and D.8, if Assumption 3.4 holds, for $g \in \mathcal{G}$ where $\mathcal{G} = \{\ell_\theta : \theta \in \Theta\}$, we have*

$$\sup_{g\in\mathcal{G}} \| \frac{1}{N}\sum_{i=1}^N g(x_i') - \mathbb{E}[g(x)]\| \leq O(\sqrt{\ln(1/\beta)/N})$$

We next present the result of excess risk for third-order flow.

Lemma D.13 (excess risk)**.** *As we defined in Definition D.7 and D.8, we have*

$$\widetilde{L}_{\mathrm{3rd}} = \frac{1}{N}\sum_{i=1}^N [\|\dot{x}_t^{\mathrm{true},i} - u_{1,\theta_1}(x_t^i, t_i)\|^2 + \|\ddot{x}_t^{\mathrm{true},i} - u_{2,\theta_2}(u_{1,\theta_1}(x_t^i,t_i), x_t^i, t_i)\|^2 + \| \dddot{x}_t^{\mathrm{true},i} - u_{3,\theta_3}(u_{2,\theta_2}(\cdot), x_t^i, t_i)\|^2]$$

and

$$L_{\mathrm{3rd}}(\theta_1,\theta_2,\theta_3) = \underbrace{\mathbb{E}[\|\dot{x}_t^{\mathrm{true}} - u_{1,\theta_1}(x_t,t)\|^2]}_{L_{3,1,\theta_1}} + \underbrace{\mathbb{E}[\|\ddot{x}_t^{\mathrm{true}} - u_{2,\theta_2}(u_{1,\theta_1}(x_t,t),x_t,t)\|^2]}_{L_{3,2,\theta_2,\theta_1}} + \underbrace{\mathbb{E}[\| \dddot{x}_t^{\mathrm{true}} - u_{3,\theta_3}(u_{2,\theta_2}(\cdot),x_t,t)\|^2]}_{L_{3,3,\theta_3,\theta_2,\theta_1}}$$

Suppose $\mathcal{F}_1$ and $\mathcal{F}_2$ have finite or at most polynomially growing complexities $\mathcal{C}(\mathcal{F}_1), \mathcal{C}(\mathcal{F}_2)$. Then for $\beta \in (0,1)$, with probability at least $1-\beta$, we have

$$|\widetilde{L}_{3rd} - L_{3rd}| \leq O((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \ln(1/\beta))/N)^{1/2}.$$

Proof. Let $\mathcal{G} = \{\ell_\theta : \theta \in \Theta\}$ represent the complexity of $\mathcal{G}$ the Rademacher/VC dimension, As we defined in Definition D.5, D.7 and D.8 we calculate the empirical loss and population loss,

$$\widetilde{L}_{\mathrm{3rd}} = \frac{1}{N}\sum_{i=1}^N [\|\dot{x}_t^{\mathrm{true},i} - u_{1,\theta_1}(x_t^i, t_i)\|^2 + \|\ddot{x}_t^{\mathrm{true},i} - u_{2,\theta_2}(u_{1,\theta_1}(x_t^i,t_i), x_t^i, t_i)\|^2 + \| \dddot{x}_t^{\mathrm{true},i} - u_{3,\theta_3}(u_{2,\theta_2}(\cdot), x_t^i, t_i)\|^2]$$

and

$$L_{\mathrm{3rd}}(\theta_1,\theta_2,\theta_3) = \underbrace{\mathbb{E}[\|\dot{x}_t^{\mathrm{true}} - u_{1,\theta_1}(x_t,t)\|^2]}_{L_{3,1,\theta_1}} + \underbrace{\mathbb{E}[\|\ddot{x}_t^{\mathrm{true}} - u_{2,\theta_2}(u_{1,\theta_1}(x_t,t),x_t,t)\|^2]}_{L_{3,2,\theta_2,\theta_1}} + \underbrace{\mathbb{E}[\| \dddot{x}_t^{\mathrm{true}} - u_{3,\theta_3}(u_{2,\theta_2}(\cdot),x_t,t)\|^2]}_{L_{3,3,\theta_3,\theta_2,\theta_1}}$$

let $\{x_i'\}_{i=1}^N$ be an i.i.d. sample from the same distribution as $\{x_i\}_{i=1}^N$, and let $\{\sigma_i\}_{i=1}^N$ be i.i.d. Rademacher random variables ($\sigma_i \in \{+1,-1\}$ with probability 1/2 each). Then, for any $g \in \mathcal{G}$, we have

$$\sup_{g\in\mathcal{G}} \| \frac{1}{N}\sum_{i=1}^N g(x_i) - \mathbb{E}[g(x)]\| \leq \sup_{g\in\mathcal{G}} \| \frac{1}{N}\sum_{i=1}^N (g(x_i)) - g(x_i')\| + \sup_{g\in\mathcal{G}} \| \frac{1}{N}\sum_{i=1}^N g(x_i') - \mathbb{E}[g(x)]\| \quad (10)$$

We can upper bound the first term in Eq. (10),

$$\begin{aligned}
&\sup_{g\in\mathcal{G}} \|\frac{1}{N}\sum_{i=1}^{N}(g(x_i)) - g(x_i')\| \\
\leq\ & \frac{2}{N}\mathbb{E}_{\sigma}[\sup_{g\in\mathcal{G}}\sum_{i=1}^{N}\sigma_i g(x_i)] \\
=\ & 2\widetilde{\mathcal{R}}_N(\mathcal{G}) \\
\leq\ & 2\cdot O(\sqrt{C(\mathcal{G})/N}) \\
\leq\ & O(\sqrt{(C(\mathcal{F}_1)+C(\mathcal{F}_2))/N})
\end{aligned} \tag{11}$$

where the first step follows from Lemma D.11, the second step comes from we define $\widetilde{\mathcal{R}}_N(G) := \mathbb{E}_\sigma[\sup_{g\in\mathcal{G}}\frac{1}{N}\sum_{i=1}^{N}\sigma_i g(x_i)]$, the third step follows from Assumption 3.4, the forth step follows from the definition of $\mathcal{G}$.

We can upper bound the second term in Eq. (10) by using Lemma D.12,

$$\sup_{g\in\mathcal{G}} \|\frac{1}{N}\sum_{i=1}^{N} g(x_i') - \mathbb{E}[g(x)]\| \leq O(\sqrt{\ln(1/\beta)/N}) \tag{12}$$

Loading Eq. (11) and Eq. (12), we can obtain

$$\sup_{g\in\mathcal{G}} \|\frac{1}{N}\sum_{i=1}^{N} g(x_i) - \mathbb{E}[g(x)]\| \leq O(\sqrt{(C(\mathcal{F}_1)+C(\mathcal{F}_2)+\ln(1/\beta))/N})$$

□

D.6 Discrete Propagation

In this section, we show the lemma about discrete propagation under noise quantifies how noise in the trajectory affects the error propagation in a discrete setting for third-order flow.

Lemma D.14 (Discrete Propagation with Jerk)**.** *Under Assumptions D.1, D.3 and D.4 let*

$$e_l = \|x_l^{\text{est}} - x_l^{\text{true}}\|_{H^3(\Omega)}.$$

Then there is a constant $C_{\text{prop},3} > 0$ such that

$$e_{l+1} \leq (1 + \Delta t C_{\text{prop},3})e_l + C_{\text{prop},3}\delta\epsilon\Delta t.$$

Unrolling from $l = 0$ to $l = L-1$ with $\Delta t = 1/L$ yields

$$e_L \leq e_0 \exp(C_{\text{prop},3}) + \frac{\delta\epsilon}{C_{\text{prop},3}}(\exp(C_{\text{prop},3}) - 1).$$

Proof. By Assumptions D.3 and D.4, we have

$$\begin{aligned}
x_{l+1}^{\text{est}} &= x_l^{\text{est}} + \Delta t u_{1,\theta_1}(x_l^{\text{est}}, t_l) + \frac{(\Delta t)^2}{2}u_{2,\theta_2}(u_{1,\theta_1}(\cdot), x_l^{\text{est}}, t_l) + \frac{(\Delta t)^3}{6}u_{3,\theta_3}(u_{2,\theta_2}(\cdot), x_l^{\text{est}}, t_l), \\
x_{l+1}^{\text{true}} &= x_l^{\text{true}} + \Delta t \dot{x}_l^{\text{true}} + \frac{(\Delta t)^2}{2}\ddot{x}_l^{\text{true}} + \frac{(\Delta t)^3}{6}\dddot{x}_l^{\text{true}}.
\end{aligned}$$

Subtracting the true update from the estimated one and taking the H^3-norm, the difference involves Lipschitz constants, the prior error e_l, and a noise term bounded by $\delta\epsilon$. One obtains

$$\|x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\|_{H^3(\Omega)} \le (1 + \Delta t \cdot C_{\text{prop},3})\|x_l^{\text{est}} - x_l^{\text{true}}\|_{H^3(\Omega)} + C_{\text{prop},3}\delta\epsilon\Delta t.$$

where $C_{\text{prop},3}$ depends on Lipschitz constants of u_{1,θ_1}, u_{2,θ_2}, u_{3,θ_3}, and the boundedness of $\ddot{x}_t^{\text{true}}$, $\dddot{x}_t^{\text{true}}$.

Repeating this inequality from $l = 0$ to $l = L - 1$ and noting $\Delta t = 1/L$, by a discrete Gronwall argument we have

$$\begin{aligned} e_L &= \|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^3(\Omega)} \\ &\le e_0 \exp(C_{\text{prop},3}) + \sum_{l=0}^{L-1}(C_{\text{prop},3}\delta\epsilon\Delta t \prod_{j=l+1}^{L-1}(1 + \Delta t C_{\text{prop},3})), \\ &\le e_0 \exp(C_{\text{prop},3}) + \frac{\delta\epsilon}{C_{\text{prop},3}}(\exp(C_{\text{prop},3}) - 1). \end{aligned}$$

This completes the proof. □

D.7 Main Result: Third-Order Noise Robustness

Combining the above, we obtain the final noise-robustness result for third-order flow matching in this section.

Theorem D.15 (third-order noise robustness)**.** *Suppose Assumptions D.1, D.3, 3.2 and 3.3 hold. Let $\widetilde{\theta} = (\widetilde{\theta}_1, \widetilde{\theta}_2, \widetilde{\theta}_3)$ be an approximately optimal solution minimizing the empirical loss $\widetilde{L}_{3\text{rd},\theta_1,\theta_2,\theta_3}$. Then, with probability at least $1 - \beta$, for uniform time steps $t_l = l\Delta t$ with $\Delta t = 1/L$, we have*

$$\|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^3(\Omega)} \le C_1' \exp(C_2')(e_0 + \delta\epsilon) + C_3'((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \mathcal{C}(\mathcal{F}_3) + \ln(1/\beta))/N)^{1/2},$$

where $e_0 = \|x_0^{\text{est}} - x_0^{\text{true}}\|_{H^3(\Omega)}$ denotes the initial error, and C_1', C_2', C_3' depend on Lipschitz constants, the dimension d, and Sobolev embedding constants in $H^3(\Omega)$.

Proof. Let $\widetilde{\theta} = (\widetilde{\theta}_1, \widetilde{\theta}_2)$ denote the approximate optimal solution for the estimated loss, use Lemma D.13, we have

$$|\widetilde{L}_{3\text{rd}} - L_{3\text{rd}}| \le O((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \mathcal{C}(\mathcal{F}_3)\ln(1/\beta))/N)^{1/2}.$$

Therefore, under the true distribution, $\dot{x}_t^{\text{est}}$ and $\ddot{x}_t^{\text{est}}$ approximate $\dot{x}_t^{\text{true}}$ and $\ddot{x}_t^{\text{true}}$ well in an L^2 sense.

As we defined $\dddot{x}_t^{\text{true}} \in H^3(\Omega)$,we then apply Lemma D.10, which leverages the Assumption D.1. If $L_{3\text{rd}}$ is small, there exist $C_{\text{reg},3}$ such that

$$\|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{H^3(\Omega)} \le C_{\text{reg},3}(L_{3\text{rd}}^{1/2} + \|\ddot{x}_t^{\text{est}} - \ddot{x}_t^{\text{true}}\|_{L^2(\Omega)}).$$

We conclude that the learned fields are also close to the stronger $H^2(\Omega)$ norm. As we assumed in Assumption 3.3 and D.3, For or uniform time steps $\Delta t = 1/L$, the update for the estimate is

$$\begin{aligned} x_{l+1}^{\text{est}} &= x_l^{\text{est}} + \Delta t u_{1,\widetilde{\theta}_1}(x_l^{\text{est}}, t_l) + \frac{(\Delta t)^2}{2} u_{2,\widetilde{\theta}_2}(u_{1,\widetilde{\theta}_1}(x_l^{\text{est}}, t_l), x_l^{\text{est}}, t_l) + \frac{(\Delta t)^3}{6} u_{3,\widetilde{\theta}_3}(u_{2,\widetilde{\theta}_2}(\cdot), x_l^{\text{est}}, t_l), \\ x_{l+1}^{\text{true}} &= x_l^{\text{true}} + \Delta t \dot{x}_l^{\text{true}} + \frac{(\Delta t)^2}{2}\ddot{x}_l^{\text{true}} + \frac{(\Delta t)^3}{6}\dddot{x}_l^{\text{true}}. \end{aligned}$$

Subtracting two updates and taking the H^3-norm, and invoking the Lipschitz condition in Assumption D.4, yields Lemma D.14, we have

$$e_{l+1} = \|x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\|_{H^3(\Omega)} \leq (1 + \Delta t \cdot C_{\text{prop},3})e_l + C_{\text{prop},3}\Delta t \cdot \delta \cdot \epsilon.$$

Here $C_{\text{prop},3}$ depends on Lipschitz constants and bounds on $\dot{x}$ and $\ddot{x}$, $\dddot{x}$. Iterating the above and a discrete Gronwell argument shows

$$e_L = \|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^3(\Omega)} \leq e_0 \exp(C_{\text{prop},3}) + \frac{\delta \cdot \epsilon}{C_{\text{prop},3}}(\exp(C_{\text{prop},3}) - 1).$$

Since $\exp(C_{\text{prop},3})$ is just a constant factor, denote it by e^{C_2}. Combine all of these terms then yields the exact form of Theorem D.15:

$$\|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^3(\Omega)} \leq C_1' \exp(C_2')(e_0 + \delta\epsilon) + C_3'((\mathcal{C}(\mathcal{F}_1) + \mathcal{C}(\mathcal{F}_2) + \mathcal{C}(\mathcal{F}_3) + \ln(1/\beta))/N)^{1/2},$$

Thus, we complete the proof. □

E Extension on k-th order Flow Matching

In this section, we extend the second-order flow-matching framework in Section 3.3 to incorporate third-order information. We first introduce additional assumptions in Section E.1 to ensure that the third derivative of the true trajectory is sufficiently smooth and bounded. In Section E.2, we introduce our third-order training algorithm and the inference algorithm. In Section E.3, we introduce the elliptic regularity for the third-order case. In Section E.4, we present the result of the regularization effect result for the third-order loss function. In Section E.5, we show the excess risk of the third-order case. In Section E.6, we show the lemma about discrete propagation under noise quantifies how noise in the trajectory affects the error propagation in a third-order discrete setting. In Section E.7, we prove our k-th order main result.

E.1 Preliminary

In this section, we introduce some additional definitions and assumptions specific to the k-th order extension.

Assumption E.1 (Smoothness in higher Sobolev spaces)**.** *We assume the true trajectory $x_t^{\text{true}} \in H^k(\Omega)$ and that its derivatives up to the k-th order are sufficiently smooth and bounded. Formally, there exist constants $\{M_j\}_{j=1}^k > 0$ such that*

$$\|\dot{x}_t^{(j),\text{true}}\|_{H^k(\Omega)} \leq M_j, \quad \text{for} j = 1, \ldots, k,$$

where $\dot{x}_t^{(j),\text{true}}$ denotes the j-th order time derivative of x_t^{true}. We also require these derivatives to be continuous on $[0, 1]$ in the time variable.

Then, we introduce our assumption for time discretization under k-th order update.

Assumption E.2 (Time discretization for k-th order update)**.** *Let $\Delta t = 1/L$ be the uniform step size, and define discrete times $t_l = l\Delta t$ for $l = 0, 1, \ldots, L$. We consider the following k-th order discrete update for the estimated system:*

$$x_{l+1}^{\text{est}} = x_l^{\text{est}} + \sum_{j=1}^{k} \frac{(\Delta t)^j}{j!} u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\ldots u_{1,\theta_1}(x_l^{\text{est}}, t_l)\ldots), x_l^{\text{est}}, t_l), \tag{13}$$

where each u_{j,θ_j} is a learned field approximating the j-th order derivative $\dot{x}_t^{(j),\text{true}}$.

The k-th order Lipschitz continuity is also necessary, and we present it here.

Assumption E.3 (k-th order Lipschitz continuity)**.** *We assume the learned fields $(u_{j,\theta_j})_{j=1}^{k}$ are each L-Lipschitz continuous in their spatial and temporal arguments. Formally, there exists $L > 0$ such that for any $x, y \in \mathbb{R}^d$ and $t, s \in [0, 1]$*

$$\|u_{j,\theta_j}(\dots, x, t) - u_{j,\theta_j}(\dots, y, t)\|_2 \le L\|x - y\|_2$$
$$\|u_{j,\theta_j}(\dots, x, t) - u_{j,\theta_j}(\dots, x, s)\|_2 \le L\|t - s\|_2.$$

Next we introduce the definition of k-th order loss function.

Definition E.4 (k-th order flow)**.** *A k-order flow involves a sequence of learned fields u_{j,θ_j} for $j = 1, \dots, k$, each targeting the approximation of $(\dot{x}_t^{(j)})^{\text{True}}$.*

Here is the k-th order loss function.

Definition E.5 (k-th order loss function)**.** *The k-order loss function evaluates the accuracy of approximations for each derivative:*

$$L_{\text{k-order}}(\theta_1, \dots, \theta_k) = \sum_{j=1}^{k} \mathbb{E}[\|(\dot{x}_t^{(j)})^{\text{True}} - u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots), x_t, t)\|^2].$$

And we introduce empirical k-th order loss here.

Definition E.6 (Empirical k-th order loss)**.** *Given a training dataset $\{(x_0^i, x_1^i)\}_{i=1}^{N}$ with times $\{t_i\}$ and (approximate) ground-truth derivatives up to the k-th order, the empirical k-th order loss is*

$$\widetilde{L}_{\text{k-order}}(\theta_1, \dots, \theta_k) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} \|(\dot{x}_{t_i}^{(j)})^{\text{true},i} - u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots), x_{t_i}^i, t_i)\|^2.$$

E.2 Proposed k-th Order Algorithms

In this section, we show our k-th order training algorithm and inference algorithm. First, we show the k-th order training algorithm.

Algorithm 7 Our k-th order training algorithm

procedure K-THORDERFORWARD()
  for each iteration **do**
    Random sample x_0 and time t, with target x_1
    $x_t \leftarrow \alpha_t \cdot x_0 + \sqrt{1 - \alpha_t^2} \cdot x_1$
    Compute gradient with respect to $L_{\text{k-order}}$ ▷ See Definition E.5
  end for
  return $u_1, u_2, \cdots, u_k$ ▷ k network functions
end procedure

Algorithm 8 Our k-th order inference algorithm

procedure K-THORDERINFERENCE($u_1, \dots, u_k$)
  $x_0 \sim \mathcal{N}(0, 1)$
  Initialize $x \leftarrow x_0$
  for t from 0 to 1 with step $\Delta t = 0.01$ **do**
    $x \leftarrow x + \sum_{j=1}^{k} \frac{(\Delta t)^j}{j!} \cdot u_j(u_{j-1}(\dots u_1(x,t)\dots), x, t)$
  end for
  return x
end procedure

E.3 Elliptic Regularity

In this section, we introduce the first result, which is a classical result that characterizes the relationship between different Sobolev norms for sufficiently smooth functions for k-th order flow.

Lemma E.7 (Elliptic Regularity in [Eva10])**.** *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with a sufficiently smooth boundary. Suppose $h : \Omega \to \mathbb{R}$ has weak derivatives up to order k in $L^2(\Omega)$. Then there is a constant $C_{\mathrm{reg},k} > 0$ depending on Ω such that*

$$\|h\|_{H^k(\Omega)} \leq C_{\mathrm{reg},k}(\sum_{m=0}^{k-1} \|\nabla^m h\|_{L^2(\Omega)}).$$

E.4 Regularization Effect

In this section, we now connect the second-order loss function with the Sobolev norm of the estimation error under the k-th order loss function.

Lemma E.8 (regularization effect for k-th order loss)**.** *As we defined in Definition E.6, then we define*

$$L_{\mathrm{k-order}}(\theta_1, \dots, \theta_k) = \sum_{j=1}^{k} \mathbb{E}[\|(\dot{x}_t^{(j)})^{\mathrm{True}} - u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots), x_t, t)\|^2].$$

If $(\dot{x}_t^{\mathrm{true}})^{(k)} \in H^3(\Omega)$ and $L_{\mathrm{k-order}}$ is sufficiently small, then there exists a constant C_{regk} such that

$$\|(\dot{x}_t^{\mathrm{est}})^{(k-1)} - (\dot{x}_t^{\mathrm{true}})^{(k-1)}\|_{H^k(\Omega)} \leq C_{\mathrm{regk}}(L_{\mathrm{k-order}}^{1/2} + \|(\dot{x}_t^{\mathrm{est}})^{(k-1)} - (\dot{x}_t^{\mathrm{true}})^{(k-1)}\|_{L^2(\Omega)}).$$

Proof. Applying Lemma E.7 to $h(\cdot) = \ddot{x}_t^{\mathrm{est}} - (\dot{x}_t^{\mathrm{true}})^{(k-1)}$, we have

$$\begin{aligned} &\|(\dot{x}_t^{\mathrm{est}})^{(k-1)} - (\dot{x}_t^{\mathrm{true}})^{(k-1)}\|_{H^k(\Omega)} \\ \leq\ & C_{\mathrm{regk}}(\sum_{m=0}^{k-1} \|\nabla^m h\|_{L^2(\Omega)}). \end{aligned} \tag{14}$$

By the Definition of the loss function, a small $L_{2,2,\theta_1,\theta_2}$ implies

$$\|(\dot{x}_t^{\mathrm{est}})^{(k-1)} - (\dot{x}_t^{\mathrm{true}})^{(k-1)}\|_{L^2(\Omega)} \lesssim L_{\mathrm{k-order}}^{1/2} \tag{15}$$

Combining Eq.(14) and (15), we have

$$\begin{aligned}&\|(\dot{x}_t^{\text{est}})^{(k-1)} - (\dot{x}_t^{\text{true}})^{(k-1)}\|_{H^k(\Omega)}\\ \leq\ & C_{\text{regk}}(L_{\text{k-order}}^{1/2} + \|(\dot{x}_t^{\text{est}})^{(k-1)} - (\dot{x}_t^{\text{true}})^{(k-1)}\|_{L^2(\Omega)}).\end{aligned}$$

Thus, we complete the proof. □

E.5 Excess Risk

In this section, we present a result that quantifies the gap between the empirical and population loss, highlighting the strong generalization capabilities of our method even with a finite sample size.

Lemma E.9 (Excess risk for k-th order)**.** *As in Definition E.5 and E.6, let*

$$\widetilde{L}_{\text{k-order}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}\|(\dot{x}_{t_i}^{(j)})^{\text{true},i} - u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots), x_{t_i}^i, t_i)\|^2$$

and

$$L_{\text{k-order}}(\theta_1,\dots,\theta_k) = \sum_{j=1}^{k}\mathbb{E}[\|\dot{x}_t^{(j),\text{true}} - u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots), x_t, t)\|^2].$$

Suppose each function class $\mathcal{F}_j$ has finite or at most polynomially growing complexity $\mathcal{C}(\mathcal{F}_j)$. Then for $\beta \in (0,1)$, with probability at least $1-\beta$, we have

$$|\widetilde{L}_{\text{k-order}} - L_{\text{k-order}}| \leq O(\sum_{i=1}^{k} C(\mathcal{F}_i) + \ln(1/\beta))/N)^{1/2}.$$

Proof. Let $\mathcal{G} = \{\ell_\theta : \theta \in \Theta\}$ represent the complexity of $\mathcal{G}$ the Rademacher/VC dimension, as we defined in Definition D.5, D.7 and D.8 we calculate the empirical loss and population loss,

$$\widetilde{L}_{\text{k-order}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}\|(\dot{x}_{t_i}^{(j)})^{\text{true},i} - u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots), x_{t_i}^i, t_i)\|^2$$

and

$$L_{\text{k-order}}(\theta_1,\dots,\theta_k) = \sum_{j=1}^{k}\mathbb{E}[\|\dot{x}_t^{(j),\text{true}} - u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots), x_t, t)\|^2].$$

let $\{x_i'\}_{i=1}^N$ be an i.i.d. sample from the same distribution as $\{x_i\}_{i=1}^N$, and let $\{\sigma_i\}_{i=1}^N$ be i.i.d. Rademacher random variables ($\sigma_i \in \{+1,-1\}$ with probability $1/2$ each). Then, for any $g \in \mathcal{G}$, we have

$$\sup_{g\in\mathcal{G}}\|\frac{1}{N}\sum_{i=1}^{N} g(x_i) - \mathbb{E}[g(x)]\| \leq \sup_{g\in\mathcal{G}}\|\frac{1}{N}\sum_{i=1}^{N}(g(x_i)) - g(x_i')\| + \sup_{g\in\mathcal{G}}\|\frac{1}{N}\sum_{i=1}^{N} g(x_i') - \mathbb{E}[g(x)]\| \quad (16)$$

We can upper bound the first term in Eq. (16),

$$\sup_{g\in\mathcal{G}}\|\frac{1}{N}\sum_{i=1}^{N}(g(x_i)) - g(x_i')\|$$

$$
\begin{aligned}
&\leq \frac{2}{N} \mathop{\mathbb{E}}_{\sigma}[\sup_{g \in \mathcal{G}} \sum_{i=1}^{N} \sigma_i g(x_i)] \\
&= 2\widetilde{\mathcal{R}}_N(\mathcal{G}) \\
&\leq 2 \cdot O(\sqrt{C(\mathcal{G})/N}) \\
&\leq O((\sum_{i=1}^{k} C(\mathcal{F}_i)/N)^{1/2}) \qquad (17)
\end{aligned}
$$

where the first step follows from Lemma D.11, the second step comes from we define $\widetilde{\mathcal{R}}_N(G) := \mathbb{E}_\sigma[\sup_{g \in \mathcal{G}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i g(x_i)]$, the third step follows from Assumption 3.4, the forth step follows from the definition of $\mathcal{G}$ in k-th order case.

We can upper bound the second term in Eq. (16) by using Lemma D.12,

$$
\sup_{g \in \mathcal{G}} \| \frac{1}{N} \sum_{i=1}^{N} g(x_i') - \mathbb{E}[g(x)] \| \leq O(\sqrt{\ln(1/\beta)/N}) \qquad (18)
$$

Loading Eq. (17) and Eq. (18), we can obtain

$$
\sup_{g \in \mathcal{G}} \| \frac{1}{N} \sum_{i=1}^{N} g(x_i) - \mathbb{E}[g(x)] \| \leq O(((\sum_{i=1}^{k} C(\mathcal{F}_i) + \ln(1/\beta))/N)^{1/2})
$$

□

E.6 Discrete Propagation

In this section, we show the lemma about discrete propagation under noise quantifies how noise in the trajectory affects the error propagation in a discrete setting for k-th order flow.

Lemma E.10 (Discrete propagation under noise for k-th order)**.** *Under Assumptions E.2 and E.3, let*

$$
e_l = \| x_l^{\text{est}} - x_l^{\text{true}} \|_{H^k(\Omega)}.
$$

Then there is a constant $C_{\text{prop},k} > 0$ such that

$$
e_{l+1} \leq (1 + \Delta t C_{\text{prop},k}) e_l + C_{\text{prop},k} \delta \epsilon \Delta t.
$$

Iterating from $l = 0$ to $l = L - 1$ (where $\Delta t = 1/L$) gives

$$
e_L \leq e_0 \exp(C_{\text{prop},k}) + \frac{\delta \epsilon}{C_{\text{prop},k}} (\exp(C_{\text{prop},k}) - 1).
$$

Proof. By Assumptions E.2 and E.3, we have

$$
\begin{aligned}
x_{l+1}^{\text{est}} = x_l^{\text{est}} &+ \Delta t u_{1,\theta_1}(x_l^{\text{est}}, t_l) + \frac{(\Delta t)^2}{2} u_{2,\theta_2}(u_{1,\theta_1}(\cdot), x_l^{\text{est}}, t_l) + \frac{(\Delta t)^3}{6} u_{3,\theta_3}(u_{2,\theta_2}(\cdot), x_l^{\text{est}}, t_l) \\
&+ \sum_{j=4}^{k} \frac{(\Delta t)^j}{j!} u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots u_{1,\theta_1}(x_l^{\text{est}}, t_l) \dots), x_l^{\text{est}}, t_l),
\end{aligned}
$$

$$x_{l+1}^{\text{true}} = x_l^{\text{true}} + \Delta t \dot{x}_l^{\text{true}} + \frac{(\Delta t)^2}{2}\ddot{x}_l^{\text{true}} + \frac{(\Delta t)^3}{6}\dddot{x}_l^{\text{true}} + \sum_{j=4}^{k}\frac{(\Delta t)^j}{j!}(\dot{x}_t^{\text{true}})^{(j)}.$$

Subtracting the true update from the estimated one and taking the H^k-norm, the difference involves Lipschitz constants, the prior error e_l, and a noise term bounded by $\delta\epsilon$. One obtains

$$\|x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\|_{H^k(\Omega)} \le (1 + \Delta t \cdot C_{\text{prop,k}})\|x_l^{\text{est}} - x_l^{\text{true}}\|_{H^k(\Omega)} + C_{\text{prop,k}}\delta\epsilon\Delta t.$$

where $C_{\text{prop,k}}$ depends on Lipschitz constants of u_{1,θ_1}, u_{2,θ_2}, $\dots u_{k,\theta_k}$, and the boundedness of $(\dot{x}_t^{\text{true}})^{(k-1)}, (\dot{x}_t^{\text{true}})^k$.

Repeating this inequality from $l = 0$ to $l = L-1$ and noting $\Delta t = 1/L$, by a discrete Gronwall argument we have

$$\begin{aligned} e_L &= \|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^k(\Omega)} \\ &\le e_0 \exp(C_{\text{prop,k}}) + \sum_{l=0}^{L-1}(C_{\text{prop,k}}\delta\epsilon\Delta t \prod_{j=l+1}^{L-1}(1 + \Delta t C_{\text{prop,k}})), \\ &\le s e_0 \exp(C_{\text{prop,k}}) + \frac{\delta\epsilon}{C_{\text{prop,k}}}(\exp(C_{\text{prop,k}}) - 1). \end{aligned}$$

This completes the proof. □

E.7 Main result for k-th Order Noise Robustness

In this section, we formally state and prove our main result, leveraging the auxiliary lemmas to demonstrate the robustness of the learned trajectory against noise and the effects of finite sample sizes for k-th order flow.

Theorem E.11 (Noise robustness for k-th order flow matching)**.** *Suppose Assumptions E.1, E.2, E.3, and 3.3 hold. Let*

$$\widetilde{\theta} = (\widetilde{\theta}_1, \dots, \widetilde{\theta}_k)$$

be an approximately optimal solution minimizing the empirical k-th order loss in Definition E.6. Then, with probability at least $1-\beta$, the final-time estimate x_L^{est} satisfies:

$$\|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^k(\Omega)} \le C_1'' \exp(C_2'')(e_0 + \delta\epsilon) + C_3''((\sum_{i=1}^{k}\mathcal{C}(\mathcal{F}_i) + \ln(1/\beta))/N)^{1/2},$$

where $e_0 = \|x_0^{\text{est}} - x_0^{\text{true}}\|_{H^k(\Omega)}$ and C_1'', C_2'', C_3'' depend on the Lipschitz constants, Sobolev embedding constants, and dimension d. The term $e^{C_2''}$ arises from the discrete Gronwall factor over $[0,1]$.

Proof. Let $\widetilde{\theta} = (\widetilde{\theta}_1, \dots, \widetilde{\theta}_k)$ denote the approximate optimal solution for the estimated loss, use Lemma E.9, we have

$$|\widetilde{L}_{\text{k-order}} - L_{\text{k-order}}| \le O(\sum_{i=1}^{k} C(\mathcal{F}_i) + \ln(1/\beta))/N)^{1/2}.$$

Therefore, under the true distribution, $\dot{x}_t^{\text{est}}$ and $\ddot{x}_t^{\text{est}}$ approximate $\dot{x}_t^{\text{true}}$ and $\ddot{x}_t^{\text{true}}$ well in an L^2 sense, the higher order field also hold.

As we defined $(\dot{x}_t^{\text{true}})^{(k)} \in H^k(\Omega)$,we then apply Lemma E.8, which leverages the Assumption E.1. If $L_{\text{k-order}}$ is small, there exist C_{regk} such that

$$\|(\dot{x}_t^{\text{est}})^{(k-1)} - (\dot{x}_t^{\text{true}})^{(k-1)}\|_{H^k(\Omega)} \le C_{\text{regk}}(L_{\text{k-order}}^{1/2} + \|(\dot{x}_t^{\text{est}})^{(k-1)} - (\dot{x}_t^{\text{true}})^{(k-1)}\|_{L^2(\Omega)}).$$

We conclude that the learned fields are also close to the stronger $H^2(\Omega)$ norm. As we assumed in Assumption 3.3 and E.2, For or uniform time steps $\Delta t = 1/L$, the update for the estimate is

$$\begin{aligned} x_{l+1}^{\text{est}} &= x_l^{\text{est}} + \Delta t u_{1,\theta_1}(x_l^{\text{est}}, t_l) + \frac{(\Delta t)^2}{2} u_{2,\theta_2}(u_{1,\theta_1}(\cdot), x_l^{\text{est}}, t_l) + \frac{(\Delta t)^3}{6} u_{3,\theta_3}(u_{2,\theta_2}(\cdot), x_l^{\text{est}}, t_l) \\ &\quad + \sum_{j=4}^{k} \frac{(\Delta t)^j}{j!} u_{j,\theta_j}(u_{j-1,\theta_{j-1}}(\dots u_{1,\theta_1}(x_l^{\text{est}}, t_l)\dots), x_l^{\text{est}}, t_l), \\ x_{l+1}^{\text{true}} &= x_l^{\text{true}} + \Delta t \dot{x}_l^{\text{true}} + \frac{(\Delta t)^2}{2} \ddot{x}_l^{\text{true}} + \frac{(\Delta t)^3}{6} \dddot{x}_l^{\text{true}} + \sum_{j=4}^{k} \frac{(\Delta t)^j}{j!} (\dot{x}_t^{\text{true}})^{(j)}. \end{aligned}$$

Subtracting two updates and taking the H^k-norm, and invoking the Lipschitz condition in Assumption E.3, yields Lemma D.14, we have

$$e_{l+1} = \|x_{l+1}^{\text{est}} - x_{l+1}^{\text{true}}\|_{H^k(\Omega)} \le (1 + \Delta t \cdot C_{\text{prop,k}}) e_l + C_{\text{prop,k}} \Delta t \cdot \delta \cdot \epsilon.$$

Here $C_{\text{prop,k}}$ depends on Lipschitz constants and bounds on $\dot{x}$, $\ddot{x}, \dots, \dot{x}_t^{(k)}$. Iterating the above and a discrete Gronwell argument shows

$$e_L = \|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^k(\Omega)} \le e_0 \exp(C_{\text{prop,k}}) + \frac{\delta \cdot \epsilon}{C_{\text{prop,k}}} (\exp(C_{\text{prop,k}}) - 1).$$

Since $\exp(C_{\text{prop}})$ is just a constant factor, denote it by e^{C_2}. Combine all of these terms then yields the exact form of Theorem E.11:

$$\|x_L^{\text{est}} - x_L^{\text{true}}\|_{H^k(\Omega)} \le C_1'' \exp(C_2'')(e_0 + \delta\epsilon) + C_3''((\sum_{i=1}^{k} \mathcal{C}(\mathcal{F}_i) + \ln(1/\beta))/N)^{1/2},$$

Thus, we complete the proof. □

F Empirical Ablation Study

In Section F.1, we introduce the three datasets used in our experiments: the five-mode, 3-round spiral, and dot-hyperbola datasets. In Section F.2, we present results for the first-order loss applied to the datasets, and in Section F.3, we examine the effect of the second-order loss. Section F.4 extends this analysis by including the third-order loss.

F.1 Three Dataset

We employ three datasets for our experiments: the five-mode dataset, the 3-round spiral dataset, and the dot-hyperbola Gaussian mixture distribution dataset, all with a variance of 0.3 for each Gaussian component. In the five-mode dataset, five source modes (**orange**) are positioned at a distance of $D_0 = 6$ from the origin, and five target modes (**pink**) are positioned at $D_0 = 13$, each mode containing 200 sampled points. For the 3-round spiral dataset, 600 points are drawn from Gaussian distributions, each with a variance of 0.3, for both the source and target distributions. Similarly, the dot-hyperbola dataset consists of 900 points sampled from Gaussian distributions with a variance of 0.3 for both the source and target.

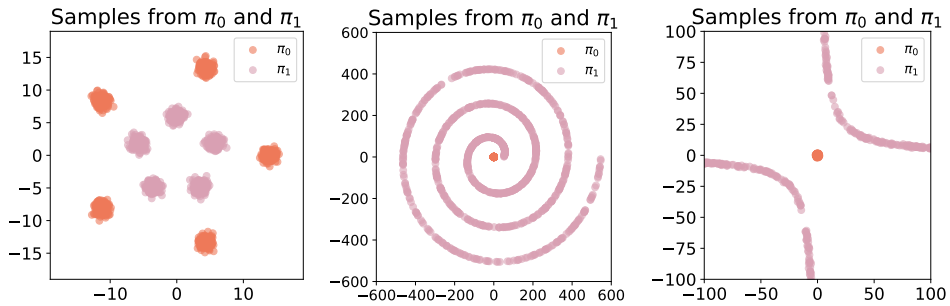


Figure 2: Gaussian mixture distributions visualized: five-mode dataset (**Left**), 3-round spiral dataset (**Middle**), and dot-hyperbola dataset (**Right**). The primary objective is for NRFlow to learn the transport trajectory from the source distribution π_0 (**orange**) to the target distribution π_1 (**pink**).

F.2 Only First Order Loss

The models are optimized by minimizing the sum of squared error (SSE). Both the source and target distributions are Gaussian. The target transport trajectory is modeled using the VP ODE framework from [LGL22], expressed as $x_t = \alpha_t x_0 + \beta_t x_1$. The parameters α_t and β_t are defined as $\alpha_t = \exp(-\frac{1}{4}a(1-t)^2 - \frac{1}{2}b(1-t))$ and $\beta_t = \sqrt{1-\alpha_t^2}$, with hyperparameters $a = 19.9$ and $b = 0.1$. In each of the five-mode, 3-round spiral, and dot-hyperbola datasets, 100 points are sampled from both the source and target distributions for each mode. The five-mode dataset training involves an ODE solver and Adam optimizer, using a 2-layer MLP with 100 hidden dimensions, a batch size of 800, a learning rate of 0.005, and 2000 training steps. For the 3-round spiral dataset, the training setup is similar, except with a batch size of 1000 and 1000 training steps. For the dot-hyperbola dataset, the batch size is increased to 1600 while maintaining the same learning rate and optimizer settings and 1000 training steps.

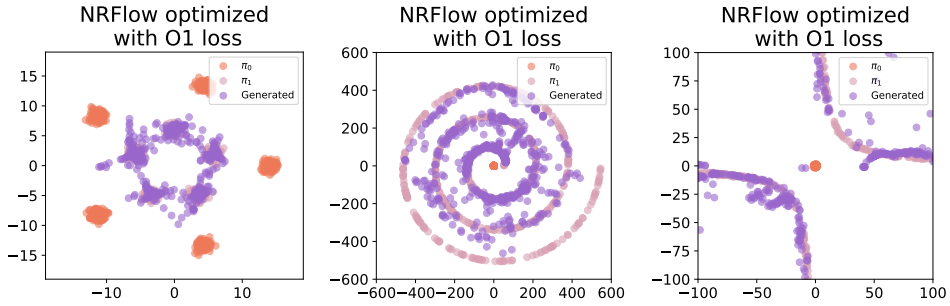


Figure 3: NRFlow generated distributions optimized by the first-order loss only: five-mode dataset (**Left**), 3-round spiral dataset (**Middle**), and dot-hyperbola dataset (**Right**). The source distribution π_0 (**orange**), the target distribution π_1 (**pink**), and the generated distribution (**purple**) are shown.

F.3 Second Order NRFlow

The models are optimized by minimizing the sum of squared error (SSE). Both the source and target distributions are Gaussian. The target transport trajectory is modeled using the VP ODE framework from [LGL22], expressed as $x_t = \alpha_t x_0 + \beta_t x_1$. The parameters α_t and β_t are defined as

$\alpha_t = \exp(-\frac{1}{4}a(1-t)^2 - \frac{1}{2}b(1-t))$ and $\beta_t = \sqrt{1-\alpha_t^2}$, with hyperparameters $a = 19.9$ and $b = 0.1$. In each of the five-mode, 3-round spiral, and dot-hyperbola datasets, 100 points are sampled from both the source and target distributions for each mode. The five-mode dataset training involves an ODE solver and Adam optimizer, using a 2-layer MLP with 100 hidden dimensions, a batch size of 800, a learning rate of 0.005, and 2000 training steps. For the 3-round spiral dataset, the training setup is similar, except with a batch size of 1000 and 1000 training steps. For the dot-hyperbola dataset, the batch size is increased to 1600 while maintaining the same learning rate and optimizer settings and 1000 training steps.

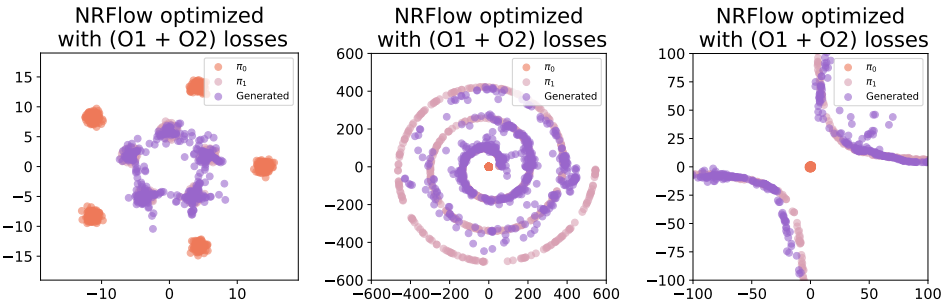


Figure 4: NRFlow generated distributions optimized by the first order and second order losses: five-mode dataset (**Left**), 3-round spiral dataset (**Middle**), and dot-hyperbola dataset (**Right**). The source distribution π_0 (**orange**), the target distribution π_1 (**pink**), and the generated distribution (**purple**) are shown.

F.4 Third Order NRFlow

The models are optimized by minimizing the sum of squared error (SSE). Both the source and target distributions are Gaussian. The target transport trajectory is modeled using the VP ODE framework from [LGL22], expressed as $x_t = \alpha_t x_0 + \beta_t x_1$. The parameters α_t and β_t are defined as $\alpha_t = \exp(-\frac{1}{4}a(1-t)^2 - \frac{1}{2}b(1-t))$ and $\beta_t = \sqrt{1-\alpha_t^2}$, with hyperparameters $a = 19.9$ and $b = 0.1$. In each of the five-mode, 3-round spiral, and dot-hyperbola datasets, 100 points are sampled from both the source and target distributions for each mode. The five-mode dataset training involves an ODE solver and Adam optimizer, using a 2-layer MLP with 100 hidden dimensions, a batch size of 800, a learning rate of 0.005, and 2000 training steps. For the 3-round spiral dataset, the training setup is similar, except with a batch size of 1000 and 1000 training steps. For the dot-hyperbola dataset, the batch size is increased to 1600 while maintaining the same learning rate and optimizer settings and 1000 training steps.

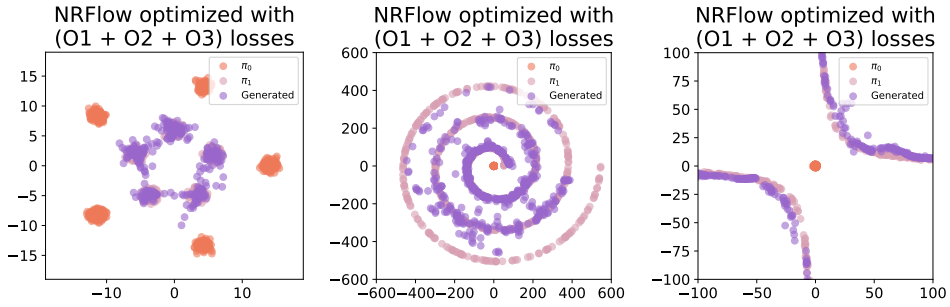


Figure 5: NRFlow generated distributions optimized by the first-order, second-order, and third-order losses: five-mode dataset (**Left**), 3-round spiral dataset (**Middle**), and dot-hyperbola dataset (**Right**). The source distribution π_0 (**orange**), the target distribution π_1 (**pink**), and the generated distribution (**purple**) are shown.