TARGET-GUIDED FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow Matching has become an important generative modeling framework, which is trained by regressing the vector fields of fixed conditional probability paths. We prove that the probability path is jointly determined by the source distribution and the velocity field. This result motivates designing a source distribution that captures the statistical structure of the target distribution to improve the generative performance of Flow Matching. To this end, we propose Target-Guided Flow Matching (TGFlow), which utilizes the marginal second-order statistics (i.e., the variance of each dimension) of the target data to construct an initial source distribution. TGFlow preserves the analytical forms of the conditional velocity field and optimization objective of Flow Matching, requires no additional training costs, and shortens the average length of actual generation trajectories, bringing the generated distribution closer to the target distribution. Experiments on 2D synthetic datasets and high-dimensional image datasets such as CIFAR-10 and ImageNet demonstrate that TGFlow improves generation quality while reducing NFE. We further verify that TGFlow supports various samplers and can be extended to a broader range of source distributions.

1 INTRODUCTION

Flow Matching (Lipman et al., 2023) has emerged as an efficient generative modelling framework following continuous normalizing flows (CNFs) (Chen et al., 2018) and diffusion models (Song et al., 2020). The core idea is to learn a time-dependent velocity field that causes the samples to gradually evolve along a continuous probability flow from a simple source distribution to the target data distribution. Flow Matching is capable of directly learning deterministic continuous dynamical systems, and therefore offers advantages such as training stability and high numerical solution efficiency. Currently, it has been widely applied in various fields, including image generation (Liu et al., 2022; Esser et al., 2024; Liang et al., 2026; Geng et al., 2026), 3D molecular generation (Jing et al., 2024; Geffner et al., 2026), multimodal modeling (Chefer et al., 2026; Tewel et al., 2026), and scientific computing (Zhu & Lin, 2024; Song & Yu, 2026; Pourya et al., 2026), and has gradually emerged as a major research direction in generative modeling.



Figure 1: Generated samples by a model trained on ImageNet-128 using Target-Guided Flow Matching.

From probabilistic perspectives, Flow Matching obeys the law of probability conservation. We prove that the probability paths in Flow Matching are jointly determined by the velocity field and the source distribution, highlighting two complementary avenues for improving generative performance. One line of research focuses on improving velocity field learning through formulations in non-Euclidean spaces (Chen & Lipman, 2024; Collas et al., 2026; Zaghen et al., 2026), prior-constrained velocity fields (Kornilov et al., 2024), and learnable velocity fields (Shaul et al., 2025). A complementary line of research focuses on designing source distributions that better capture the statistical structure of the target distribution. Such alignment can reduce the required transport distance and facilitate velocity field learning, thereby enhancing the model's ability to fit the target distribution. In this line of research, existing methods primarily target conditional generation tasks, constructing source

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

distributions using predefined parameters, conditional information, or geometric constraints. Specifically, Chen et al. (2026) employs a shifted and scaled Gaussian distribution, Kim et al. (2026); Lee et al. (2026); Wang et al. (2026b) construct source distributions based on conditional information or geometric constraints, and Luo et al. (2026) uses a Gaussian mixture distribution. However, in the more general setting, these methods have limited ability to construct source distributions that accurately capture the statistical structure of the target distribution.

To move beyond these limitations, we seek a source distribution design that directly captures statistical information from the target distribution without requiring conditioning information or predefined parameters. Marginal second-order statistics offer a tractable starting point: they quantify variation along individual dimensions and can be estimated directly from target samples. This raises a natural question: how can we systematically incorporate the target distribution's marginal second-order statistics into the learning of probability paths in Flow Matching? Addressing this question requires more than modifying the source distribution alone. In Flow Matching, the mutual relationships among the modified source distribution, probability paths, and conditional velocity fields remain insufficiently understood. A principled solution therefore requires clarifying these relationships and incorporating the modified source distribution consistently into the Flow Matching framework.

In this work, we propose TGFlow, whose core idea is to reconstruct the source distribution using the marginal second-order statistics of the target data. We prove that TGFlow preserves the analytical forms of the conditional velocity field and loss function in Flow Matching (Details in Section 2.2), enabling the method to be realized without incurring additional training overhead. Meanwhile, we prove that a better source distribution reduces the KL divergence between the generated distribution and the target distribution. (Details in Section 3.2). Experiments on 2D synthetic datasets show that TGFlow reduces the average length of generation trajectories, improves alignment between the generated and target distributions, and accommodates a broader range of source distributions. On high-dimensional datasets, including CIFAR-10 and ImageNet, TGFlow outperforms Flow Matching baselines, delivering higher generation quality and lower negative log-likelihood with fewer function evaluations (NFE). TGFlow achieves performance improvements across various samplers and can be extended to a broader range of source distributions. Figure 1 presents unconditional 128×128 ImageNet samples generated by TGFlow.

2 SOURCE-VELOCITY COUPLING PERSPECTIVE OF FLOW MATCHING

This section introduces Flow Matching from both dynamical and probabilistic perspectives. From the dynamical perspective, Flow Matching learns a time-dependent velocity field to transport samples from the source distribution to the target distribution along a probability path. We then interpret this process as probability-mass-conserving transport and proves that the probability path is jointly determined by the source distribution and the velocity field.

2.1 FLOW MATCHING

Flow Matching (FM) (Lipman et al., 2023) aims to continuously transform a simple source distribution into a complex target data distribution. Unlike diffusion models, which generate data through an inverse diffusion process, Flow Matching directly learns a deterministic time-varying vector field and generates data by solving ordinary differential equations (ODEs):

$$\dot{\boldsymbol{x}}_t = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t). \tag{1}$$

Specifically, $x_0 \sim p_0(x)$ denotes an initial sample drawn from the source distribution, $x_1 \sim p_1(x)$ denotes a sample drawn from the target data distribution, and $v_\theta(x, t)$ is the time dependent vector field to be learned,the time interval $t \in [0, 1]$. To solve for the velocity field described above, Flow Matching first constructs a continuous probability path between the source distribution and the target distribution: $p_t(x)$ it satisfies:

$$p_0(x) \to p_t(x) \to p_1(x). \tag{2}$$

To construct learnable conditional paths, Conditional Flow Matching (CFM) further defines a conditional transmission trajectory for each pair of samples (x_0, x_1). In the most commonly used optimal transport (OT) path, $\mathcal{T}_t(\cdot)$ takes the form of linear interpolation:

$$\mathcal{T}_t(x) = (1 - t)x + tx_1. \tag{3}$$

It can be seen that linear interpolation is merely a specific implementation of the transport flow map $\mathcal{T}_t(\cdot)$, and different probability paths correspond to different forms of the transport flow map.

Given a Transport Flow Map, the corresponding Conditional Vector Field is defined as:

$$u_t(x) = \mathbb{E}\left[x_1 - x_0 \,\middle|\, x_t = x\right]. \tag{4}$$

This vector field describes the expected transport direction of a particle located at position x at time t. Finally, Flow Matching uses supervised learning to make the velocity field predicted by the network approximate the true conditional vector field, and its optimization objective is formulated as:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{x_0\sim p_0,\, x_1\sim p_1,\, t\sim\text{Uniform}[0,1]}\left[\left\|v_\theta(x_t,t) - \mathbb{E}\left[x_1 - x_0 \,\middle|\, x_t\right]\right\|^2\right]. \tag{5}$$

2.2 Probability mass conservation perspective of flow matching

The previous section introduced Flow Matching from the perspective of continuous dynamics. This section further examines FM from the broader perspective of probability theory to illustrate the importance of the source distribution in Flow Matching. In Flow Matching, particles evolve along trajectories induced by a time dependent velocity field, from an Eulerian perspective, the corresponding distributional dynamics can be interpreted as the conservative transport of probability mass governed by the continuity equation (Chen et al., 2018). Therefore, the process must obey the law of probability mass conservation, and its evolution is governed by the continuity equation:

$$\frac{\partial p(x,t)}{\partial t} + \nabla\cdot\big(p(x,t)v(x,t)\big) = 0. \tag{6}$$

To facilitate analysis, we have further defined the probability flow:

$$\boldsymbol{J}(x,t) = p(x,t)v(x,t). \tag{7}$$

Then the continuity equation can be further written as:

$$\frac{\partial p(x,t)}{\partial t} + \nabla\cdot\boldsymbol{J}(x,t) = 0. \tag{8}$$

Equation 6 establishes that the total probability mass is conserved when the net flux at infinity vanishes. (Proof in Appendix C.1). Accordingly, Flow Matching neither creates nor annihilates probability mass, rather, it learns a velocity field that transports the mass prescribed by the source distribution toward the target distribution. This conservation perspective underscores the importance of the source distribution, as it determines the initial spatial configuration of the mass to be transported.

Theorem 1 (Probability Transport Determinacy Theorem)**.** *Let $v(t,x)$ be continuous in t, globally Lipschitz and continuously differentiable with respect to x, and satisfy at most linear growth condition. Let $\mathcal{T}_t$ denote the unique invertible flow map induced by v. Define $\mathcal{F}_t(p_0,v) = (\mathcal{T}_t)_{\#}p_0$. Given $p(x,0) = p_0(x)$ and suppose equation 6 holds, then $p(\cdot,t) = \mathcal{F}_t(p_0,v)$. Moreover, for all $t\in[0,1]$, $p_0^{(1)} \neq p_0^{(2)} \Longrightarrow \mathcal{F}_t(p_0^{(1)},v) \neq \mathcal{F}_t(p_0^{(2)},v)$.*

See Appendix A.1 for the proof. The theorem establishes that the probability path is jointly determined by the source distribution and the velocity field, even when the velocity field remains fixed, altering the source distribution changes the probability path it induces. Accordingly, we leverage statistical information from the target distribution to construct the source distribution, thereby facilitating velocity field learning and probability transport.

3 Target-guided flow matching

In this subsection, we propose Target-Guided Flow Matching (TGFlow). In this framework, the standard source distribution $\mathcal{N}(0,I)$ used in Flow Matching is replaced with a source distribution. As illustrated in Figure 2, the target-guided source distribution is constructed using the marginal second order statistics of the target data, thereby achieving closer alignment with the dimension wise statistical structure of the target distribution.

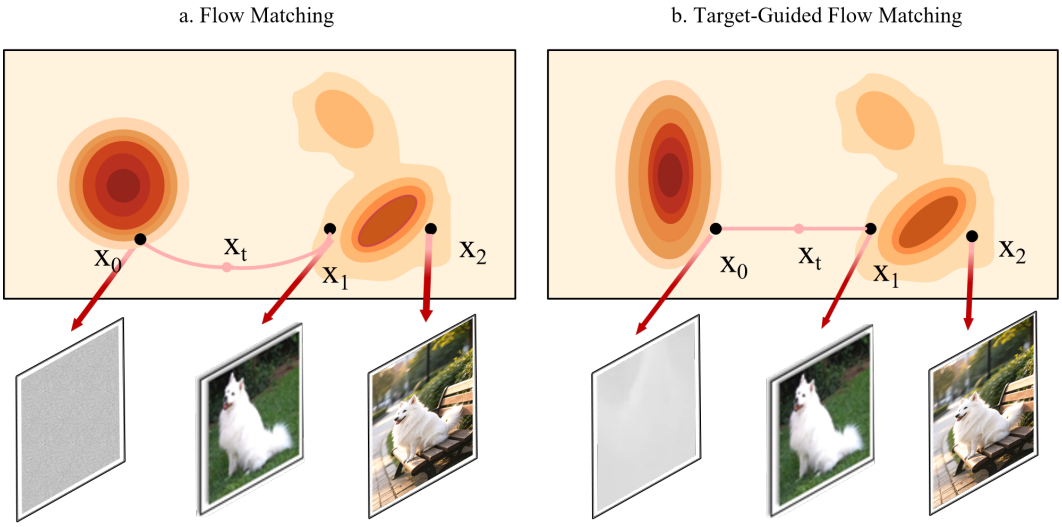

Figure 2: Illustration comparing Flow Matching and TGFlow. (a) Illustrates the Flow Matching process, (b) illustrates the TGFlow process. Both processes rely on a continuous random process $\{x_t\}_{0\leq t\leq 1}$ to achieve a step-by-step transformation from the initial source sample x_0 to the target sample x_1.

3.1 Target Variance-Aware Source Distribution

In image generation, the Flow Matching method uses the unit covariance matrix $I \in \mathbb{R}^{d\times d}$ to construct the initial source distribution $\mathcal{N}(0, I)$. Here d denotes the total dimension of flattened image data. From the perspective of the population-level distribution, $\mathcal{N}(0, I)$ assigns identical unit marginal variance to all image dimensions. As illustrated in Figure 2, x_1 and x_2 denote two images with distinct backgrounds belonging to the same semantic category. Although they share similar foreground semantics, obvious discrepancies persist in backgrounds, object boundaries and local textured regions. This example qualitatively demonstrates that different image dimensions may possess varying degrees of cross-sample variations. Specifically, pixels associated with object boundaries and textures generally yield large cross-sample variations, whereas relatively stable background regions exhibit minor variations (Wang et al., 2026a). Consequently, mismatched marginal variances between the FM source and target distributions force the learned velocity field to differentially stretch and compress the source across dimensions to progressively match the target's variance structure. Moreover, drawing on the path kinetic perspective of Flow Matching (Li et al., 2026), such statistical mismatch elevates the integrated kinetic energy of probability paths. Meanwhile, its theoretical results further demonstrate that elevated kinetic path energy gives rise to larger velocity magnitudes and increased computational overhead for ODE based sampling. Based on this, we redefine the standard normal source distribution in FM from $p(x) = \mathcal{N}(x \mid 0, I)$ to $p(x) = \mathcal{N}(x \mid 0, \Lambda)$, where $\Lambda \in \mathbb{R}^{d\times d}$ is a diagonal matrix given by $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$. We define λ_i as:

$$\lambda_i = \frac{\mathrm{Var}_{x\in\mathcal{X}}(x_i)}{\max_j \mathrm{Var}_{x\in\mathcal{X}}(x_j)}. \tag{9}$$

Here, $\mathrm{Var}_{x\in\mathcal{X}}(x_i)$ refers to the unbiased sample variance of the i-th dimension across the entire dataset $\mathcal{X}$. To obtain this statistical quantity, we traverse all training images, flatten each image into a one-dimensional vector, and calculate the unbiased variance corresponding to the i-th pixel position over the full training set. The detailed formulation is defined as follows:

$$\mathrm{Var}_{x\in\mathcal{X}}(x_i) = \frac{1}{N-1}\sum_{n=1}^{N}\left(x_i^{(n)} - \bar{x}_i\right)^2, \tag{10}$$

where $x_i^{(n)}$ is the i-th-dimensional entry of the n-th flattened training sample, and $\bar{x}_i$ is the sample mean of the i-th pixel dimension across all training data. This formulation quantifies the dataset-level

dispersion of the i-th pixel dimension: it accumulates squared deviations from the dimension-wise mean and employs Bessel's correction with $N-1$ to yield an unbiased sample variance. In equation 9, $\max_j \mathrm{Var}_{x\in\mathcal{X}}(x_j)$ denotes the maximum variance across all pixel dimensions of the flattened training images. For each dimension j, we compute its unbiased sample variance over training samples $x \in \mathcal{X}$, and take the maximum of these per-dimension variances as the global normalization constant.

3.2 Conditional probability paths and velocity fields for target-guided flow matching

In (Lipman et al., 2023), it is usually not possible to directly compute the vector field v_t required to generate p_t. Therefore, p_t and v_t must be constructed using the probability paths and conditional vector fields defined for each sample. For this model, it is not possible to directly solve for a closed-form solution of the optimal vector field. Instead, a training objective suitable for numerical optimization must be constructed using a loss function. We parameterize the marginal probability path $p_t(x) = \int p_t(x \mid x_1)q(x_1)\mathrm{d}x_1$ as a mixture of per sample conditional probability paths. Following the Gaussian conditional path assumption in Lipman et al. (2023), the conditional probability path adopted in our method is defined as follows:

$$p_t(x|x_1) = \mathcal{N}\big(x\big|\mu_t(x_1), \sigma_t(x_1)^2\Lambda\big). \tag{11}$$

Similar to the Flow Matching, $\boldsymbol{u} : [0,1]\times\mathbb{R}^d \to \mathbb{R}^d$ represents the mean function of a Gaussian distribution that varies over time, while $\sigma : [0,1]\times\mathbb{R} \to \mathbb{R}_{>0}$ is used to define the scalar standard deviation that varies over time. The difference is that we impose the following boundary conditions: at $t = 0$, the mean satisfies $\mu_0(x_1) = 0$, the standard deviation satisfies $\sigma_0(x_1) = 1$, and the covariance matrix is set to Λ. This ensures that all conditional probability paths converge at $t = 0$ to the same Gaussian noise distribution $p(x) = \mathcal{N}(x \mid 0, \Lambda)$. At the same time, at $t = 1$, we set the mean to $\mu_1(x_1) = x_1$, and the standard deviation to $\sigma_1(x_1) = \sigma_{\min}$. Therefore, the conditional distribution $p_1(x \mid x_1)$ becomes a highly concentrated Gaussian distribution centered at the true sample x_1. However, the above construction does not uniquely determine the vector field governing the probability transport. In fact, if v_t is a valid velocity field and f_t satisfies:

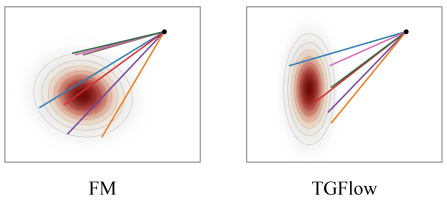

Figure 3: Illustration of trajectories for FM and TGFlow in 2D synthetic dataset example. Each colored line denotes the temporal generation trajectory of one sample.

$$\nabla\cdot\big(p_t f_t\big) = 0, \tag{12}$$

then $v_t + f_t$ induces the same probability evolution according to equation 6. Thus, infinitely many vector fields can generate the same conditional probability path. For example, a p_t weighted divergence free component, such as a rotational field for a rotation invariant distribution, can be superimposed on v_t without altering the transported density. The corresponding proof is provided in the Appendix C.2. This nonuniqueness motivates us to select a simple and analytically tractable vector field. We therefore construct the transport process through an affine transformation of Gaussian noise. Specifically, conditioned on the target sample x_1, we define:

$$\mathcal{T}_t(x) = \sigma_t(x_1)x + \mu_t(x_1). \tag{13}$$

When x follows a Gaussian distribution, the noise distribution $p_0(x \mid x_1)$ is pushed-forward through the flow map to yield $p_t(x \mid x_1)$. More precisely, in TGFlow, the source noise distribution $p(x_0) = \mathcal{N}(x \mid 0, \Lambda)$ is pushed-forward via the affine conditional flow map $\mathcal{T}_t$, giving rise to equation 11. Detailed derivations are deferred to Appendix C.3. Furthermore, according to the theory established in Lipman et al. (2023), although a prescribed probability path can generally be generated by multiple vector fields, a well-defined invertible flow map uniquely determines its associated vector field. Consequently, the unique conditional vector field that characterizes $\mathcal{T}_t$ is given by:

$$u_t(x \mid x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}\big(x - \mu_t(x_1)\big) + \mu_t'(x_1). \tag{14}$$

To obtain a more stable probability flow with lower transport costs, TGFlow method employs the OT transport described in Lipman et al. (2023), whose mean and standard deviation functions are

defined as follows: $\mu_t(x) = tx_1$, $\sigma_t(x) = 1 - (1 - \sigma_{\min})t$. Based on equation 14, the conditional probability path can be further expressed as:

$$u_t(x \mid x_1) = \frac{x_1 - (1 - \sigma)x}{1 - (1 - \sigma)t}. \tag{15}$$

Meanwhile, TGFlow retains the loss function formulation of Flow Matching as shown in equation 5.

Theorem 2 (Analytical equivalence theorem for Flow Matching)**.** *Let $\Sigma = \Sigma^\top \succ 0$, $t \in [0, 1]$, and $\sigma_t(x_1) > 0$. For*

$$p_0(x) = \mathcal{N}(x \mid 0, \Sigma), \qquad p_t(x \mid x_1) = \mathcal{N}\big(x \mid \mu_t(x_1), \sigma_t^2(x_1)\Sigma\big), \tag{16}$$

define $\mathcal{T}_t(x \mid x_1)$ satisfies the equation 13. Consequently, $u_t(x \mid x_1)$ and $\mathcal{L}(\theta)$ induced by $\mathcal{T}_t$ are analytically equivalent to their FM counterparts.

See Appendix A.2 for the proof of Theorem 2. Since the matrix Λ in TGFlow is a special case of Σ, its conditional velocity field and loss function share the same analytical forms as those of Flow Matching. Figure 3 shows the sampling paths of FM and TGFlow under optimal transport conditions. We quantify the transport trajectories of both methods using toy experiments in Section 4.2. The results demonstrate that TGFlow yields markedly shorter trajectories.

For TGFlow and FM, a smaller KL divergence indicates that their source distributions provide a better information-theoretic fit to the true target distribution. Therefore, Theorem 3 establishes that, under the stated conditions, TGFlow achieves a smaller KL divergence between the generated distribution and the true distribution than FM. The proof is provided in Appendix A.3.

Theorem 3 (Distributional approximation guarantee for TGFlow)**.** *Let $q = \mathcal{N}(0, D), D = D^\top \succ 0, \sigma_{\min} \in (0, 1)$, $\Lambda = D/\lambda_{\max}(D)$. The generative distribution of TGFlow is*

$$\widetilde{P}_\Lambda(x) = \int \mathcal{N}\big(x \mid x_1, \sigma_{\min}^2 \Lambda\big) q(x_1)\, \mathrm{d}x_1, \tag{17}$$

and the generative distribution of isotropic FM is

$$\widetilde{P}_I(x) = \int \mathcal{N}\big(x \mid x_1, \sigma_{\min}^2 I\big) q(x_1)\, \mathrm{d}x_1. \tag{18}$$

Then $\mathrm{KL}\left(\widetilde{P}_\Lambda \parallel q\right) \le \mathrm{KL}\left(\widetilde{P}_I \parallel q\right)$. Equality holds if and only if there exists a positive constant c such that $D = cI$.

3.3 Generalizing target-guided flow matching to multiple classes of source distributions

Above, we take the Gaussian distribution as an example to illustrate how to construct a new source distribution using the marginal second-order statistical information of target data. However, this construction method does not rely on the specific assumption that the source distribution follows a Gaussian distribution. Its core idea is to adjust the existing source distribution based on the statistical properties of the target distribution. Therefore, our method can be naturally extended to more general source distributions. Meanwhile, we conduct mapping experiments from moons→8gaussians on 2D synthetic datasets to verify the applicability of TGFlow to general non-Gaussian source distributions (Section 4.2).

For general source distributions, we use the OT-CFM method proposed by Tong et al. (2023) to construct conditional probability paths. In contrast to independent sampling, OT-CFM avoids independent sampling from the marginal distributions $q_0(x_0)$ and $q_1(x_1)$, and instead constructs a joint distribution that satisfies the marginal constraints:

$$\pi(x_0, x_1). \tag{19}$$

And obtain the optimal transport plan by minimizing the transport cost between the source and destination samples:

$$\pi = \underset{\pi \in \Pi(q_0, q_1)}{\arg\min} \int \|x_1 - x_0\|^2 \, d\pi(x_0, x_1), \tag{20}$$

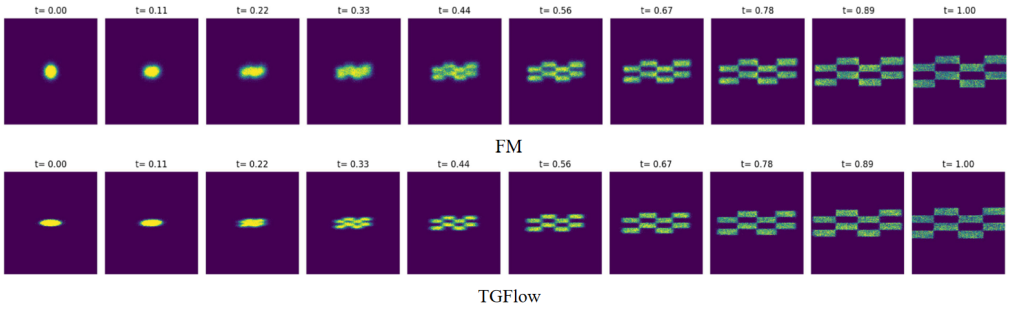

Figure 4: Trajectories of CNFS trained with different objectives on a 2D checkerboard data. The top shows the FM method, and the bottom shows the TGFlow method.

where $\pi(x_0) = q_0(x_0)$ and $\pi(x_1) = q_1(x_1)$. $\pi(q_0, q_1)$ denotes the set of all joint probability distributions satisfying the marginal constraints.

For OT-CFM, the conditional probability path is also equation 11, but $\mu_t(x_1) = tx_1 + (1-t)x_0$, and $\sigma_t(x_1) = \sigma$. Interestingly, x_0 here represents a sample drawn from any source distribution. Based on equation 13, the formula for $\mathcal{T}_t(x)$ is:

$$\mathcal{T}_t(x) = \sigma x + tx_1 + (1-t)x_0. \tag{21}$$

The corresponding condition vector field $u_t(x \mid x_1)$ can also be expressed as: $x_1 - x_0$. Furthermore, since the conditional probability paths constructed by CFM depend only on the given source sample x_0 and target sample x_1, rather than on the specific form of the source sample's distribution, there is no need to derive a new optimization objective function or modify the original loss function, training can still be performed using the loss function given in equation 5.

4 Experiments

This section systematically validates the superiority of TGFlow over standard FM and existing methods in generative performance from five dimensions: intuitive visual verification, generative comparison on 2D synthetic datasets, unconditional high-dimensional real image generation, diversity of generated images, and compatibility with various samplers. First, to intuitively verify whether TGFlow can produce more accurate and reasonable evolution trajectories under OT paths, We visually compare the evolution trajectories of TGFlow and FM on 2D checkerboard data. Second, to verify the effectiveness and generality of TGFlow on multiple typical 2D synthetic datasets, we construct four pairs of 2D synthetic dataset mappings and compare our results with Tong et al. (2024), We construct mappings for four pairs of 2D synthetic datasets and compare our experimental results with those from Tong et al. (2024). Meanwhile, we construct a custom multimodal 2D synthetic dataset to verify the trajectory distances of TGFlow. Then, to investigate whether TGFlow retains competitive advantages for high-dimensional real image generation, we adopt the hyperparameters reported in Lipman et al. (2023) to evaluate our TGFlow method on the CIFAR-10 dataset and ImageNet with resolutions of 32, 64, and 128, and compare our experimental results with those reported in Lipman et al. (2023). Additionally, we use the hyperparameters given in Tong et al. (2023) and compare our results on the CIFAR-10 dataset with those reported in Tong et al. (2023). Next, to verify that TGFlow improves generative performance without sacrificing sample diversity, we compare it with the standard Flow Matching method on the ImageNet-128 dataset. Finally, We also evaluate whether TGFlow is compatible with different samplers.

4.1 2D synthetic example

To visually validate the effectiveness of TGFlow, we conducted comparative experiments with the Flow Matching method on a 2D checkerboard dataset. This 2D synthetic dataset allows for direct visualization of the stochastic transport process, enabling us to analyze the Continuous Normalizing Flows trajectories generated by different methods. (See details in Appendix E.1)

Table 1: Comparison of W_2 distance and NPE on 2D synthetic datasets.

Metric →	W_2 (↓)				Normalized Path Energy (↓)			
Algorithm ↓ | Dataset →	$\mathcal{N}$→8gaussians	moons→8gaussians	$\mathcal{N}$→moons	$\mathcal{N}$→scurve	$\mathcal{N}$→8gaussians	moons→8gaussians	$\mathcal{N}$→moons	$\mathcal{N}$→scurve
FM (Lipman et al., 2023)	$0.343_{\pm 0.058}$	—	$0.209_{\pm 0.055}$	$0.198_{\pm 0.037}$	$0.190_{\pm 0.054}$	—	$0.762_{\pm 0.099}$	$0.743_{\pm 0.116}$
I-CFM (Tong et al., 2023)	$0.373_{\pm 0.103}$	$1.557_{\pm 0.407}$	$0.178_{\pm 0.014}$	$0.242_{\pm 0.141}$	$0.202_{\pm 0.055}$	$2.680_{\pm 0.292}$	$0.891_{\pm 0.120}$	$0.856_{\pm 0.031}$
RF (Liu, 2022)	$0.421_{\pm 0.071}$	$1.525_{\pm 0.330}$	$0.283_{\pm 0.045}$	$0.345_{\pm 0.079}$	$0.044_{\pm 0.031}$	$0.203_{\pm 0.090}$	$0.130_{\pm 0.078}$	$0.099_{\pm 0.066}$
SB-CFM (Tong et al., 2023)	$2.314_{\pm 2.112}$	—	$0.434_{\pm 0.594}$	$0.341_{\pm 0.468}$	$1.000_{\pm 0.000}$	—	$0.995_{\pm 0.000}$	$0.745_{\pm 0.039}$
OT-CFM (Tong et al., 2023)	$0.303_{\pm 0.043}$	$0.601_{\pm 0.027}$	$0.130_{\pm 0.016}$	$0.144_{\pm 0.028}$	$0.031_{\pm 0.027}$	$0.015_{\pm 0.010}$	$0.083_{\pm 0.009}$	$0.027_{\pm 0.012}$
DSB (De Bortoli et al., 2021)	$0.411_{\pm 0.084}$	$0.987_{\pm 0.324}$	$0.190_{\pm 0.049}$	$0.272_{\pm 0.065}$	—	—	—	—
DSBM-IMF (Shi et al., 2023)	$0.338_{\pm 0.091}$	$0.838_{\pm 0.098}$	$0.144_{\pm 0.024}$	$0.145_{\pm 0.037}$	$0.029_{\pm 0.017}$	$0.345_{\pm 0.049}$	$0.230_{\pm 0.028}$	$0.286_{\pm 0.033}$
DSBM-IPF (Shi et al., 2023)	$0.315_{\pm 0.079}$	$0.812_{\pm 0.092}$	$0.140_{\pm 0.006}$	$0.140_{\pm 0.024}$	$0.022_{\pm 0.020}$	$0.244_{\pm 0.027}$	$0.383_{\pm 0.034}$	$0.297_{\pm 0.036}$
$[SF]^2$M-I (Tong et al., 2024)	$0.393_{\pm 0.054}$	$1.482_{\pm 0.151}$	$0.185_{\pm 0.028}$	$0.201_{\pm 0.062}$	$0.160_{\pm 0.019}$	$2.577_{\pm 0.323}$	$0.855_{\pm 0.130}$	$0.845_{\pm 0.106}$
$[SF]^2$M-Exact (Tong et al., 2024)	$0.275_{\pm 0.058}$	$0.726_{\pm 0.137}$	$0.124_{\pm 0.023}$	$0.128_{\pm 0.005}$	$0.016_{\pm 0.012}$	$0.045_{\pm 0.031}$	$0.053_{\pm 0.038}$	$0.034_{\pm 0.024}$
TGFlow(Ours)	$\mathbf{0.261}_{\pm 0.052}$	$\mathbf{0.452}_{\pm 0.021}$	$\mathbf{0.117}_{\pm 0.020}$	$\mathbf{0.121}_{\pm 0.014}$	$\mathbf{0.012}_{\pm 0.008}$	$\mathbf{0.009}_{\pm 0.007}$	$\mathbf{0.037}_{\pm 0.024}$	$\mathbf{0.022}_{\pm 0.010}$

As shown in Figure 4, FM starts with a standard Gaussian distribution and must learn a more complex probability transport process to redistribute the initial probability mass onto the checkerboard target distribution. In contrast, the TGFlow method uses the marginal second-order statistical information of the target distribution to adjust the source distribution, making the initial probability mass of the source distribution statistically closer to that of the target distribution. As a result, it is able to learn smoother transition trajectories while simultaneously generating clearer checkerboard patterns.

4.2 2D SYNTHETIC DATASETS OPTIMAL TRANSPORT EXPERIMENT

We train ordinary differential equation mappings between four pairs of two-dimensional datasets using the TGFlow method: $\mathcal{N}$→8gaussians , $\mathcal{N}$→moons, $\mathcal{N}$→scurve and moons→8gaussians. We compare our experimental results with those reported in (Tong et al., 2024). Experimental details are provided in Appendix E.2.

As shown in Table 1, TGFlow performs better on all four groups of tasks. Interestingly, TGFlow achieves more favorable performance under the moons→8gaussians setting, which further indicates that the advantages of TGFlow are amplified in multi-modal and clustered distribution scenarios. For the TGFlow method, in the tasks $\mathcal{N}$→8gaussians , $\mathcal{N}$→moons, $\mathcal{N}$→scurve, the source distribution is the Gaussian. As such, we adopt the same OT path formulation as the FM method (equation 3). Meanwhile, the moons→8gaussians experiment also verifies that TGFlow supports arbitrary source distributions instead of being restricted to the Gaussian. Nevertheless, for an arbitrary source distribution, TGFlow establishes source-target sample matching through the optimal transport coupling defined in equation 19, and transports the coupled samples along the conditional OT path defined in equation 21.We also designed a custom multimodal 2D distribution to simulate high-dimensional image data and evaluate the trajectory distances of TGFlow and FM. (Details in the Appendix E.2.1).

4.3 TRAINING ON HIGH-DIMENSIONAL DATA INCLUDING IMAGENET AND CIFAR-10

To verify the effectiveness of the proposed method, we equip TGFlow with the optimal transport described in Section 3.2 for path construction in our experiments. The experimental results are shown in Table 2 and Table 3 in the Appendix E. We compare TGFlow with the experimental results reported in Lipman et al. (2023), using exactly the same parameters for training. For specific implementation details see Appendix E.3. Experimental results demonstrate that compared with the baseline methods, our method reduces average FID by 50.0% on the CIFAR-10 and ImageNet datasets with fewer sampling steps, further demonstrating its capability to generate higher-quality data. Figure 5(a) shows the FID curves for the TGFlow and FM methods during training on the ImageNet 32×32 dataset. We can observe that TGFlow reduces the FID value more rapidly and effectively than FM. In addition, Figures 7 - 10 in the appendix present selected samples from our method on the CIFAR-10 dataset and ImageNet (at resolutions of 32, 64, and 128).

Since the training hyperparameters adopted in (Tong et al., 2023) differ from those in (Lipman et al., 2023), we further conduct comparative experiments using the parameter settings from (Tong et al., 2023). As shown in Table 4 in the Appendix E, compared to the hyperparameter setting from (Lipman et al., 2023), this setting yields lower FID values for both FM and TGFlow. Meanwhile, TGFlow still maintains state-of-the-art performance compared with all other baselines. For specific details about the experiment, please refer to the Appendix E.4.

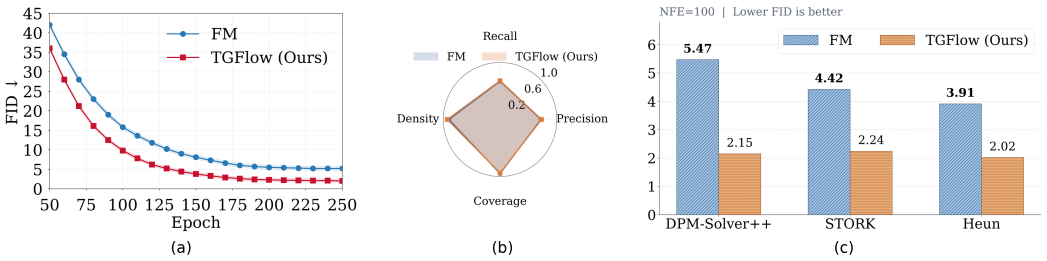

Figure 5: (a) Shows the FID curves of FM and TGFlow across training epochs. (b) Compares FM and TGFlow in terms of precision, recall, density, and coverage. (c) Compares the FID values of FM and TGFlow under different samplers at NFE = 100.

4.4 Comparision on generation diversity

To evaluate whether TGFlow can preserve the diversity of generated samples, we compare TGFlow with the baseline FM method on the unconditional generation task of the ImageNet 128 dataset. To ensure fair comparison, both methods adopt identical hyperparameter settings. Detailed experimental settings are provided in the Appendix E.5. As shown in Figure 5(b), TGFlow exhibits sample diversity similar to that of FM. This indicates that TGFlow improves generation quality without sacrificing sample diversity.

4.5 Comparison across different samplers

As shown in Figure 5(c), with the NFE fixed at 100, We compare the performance of TGFlow and FM under different samplers on the CIFAR-10. (Details in the Appendix E.6). The experimental results demonstrate that TGFlow can adapt well to different types of samplers, when using the same sampler, TGFlow consistently achieves better generation quality than FM. These consistent performance improvements suggest that the gains brought by TGFlow are not limited to any specific sampler.

Table 2: Likelihood (NLL), sample quality (FID), and evaluation time (NFE) for different methods.

	CIFAR-10			ImageNet 32×32			ImageNet 64×64		
Model	NLL↓	FID↓	NFE↓	NLL↓	FID↓	NFE↓	NLL↓	FID↓	NFE↓
DDPM	3.12	7.48	274	3.54	6.99	262	3.32	17.36	264
Score Matching	3.16	19.94	242	3.56	5.68	178	3.40	19.74	441
ScoreFlow	3.09	20.78	428	3.55	14.14	195	3.36	24.95	601
FM/ Diffusion	3.10	8.06	183	3.54	6.37	193	3.33	16.88	187
FM/ OT	2.99	6.35	142	3.53	5.02	122	3.31	14.45	138
TGFlow(Ours)	**2.76**	**2.31**	**110**	**3.32**	**2.03**	**100**	**3.04**	**9.75**	**98**

5 Conclusion

We introduce TGFlow, which revisits the influence of the source distribution on generative modeling through the lens of probability transport. By leveraging the statistical characteristics of the target data, TGFlow constructs a Gaussian source distribution that is more closely aligned with the target distribution, while further validating its potential applicability to non-Gaussian source distributions and its compatibility with different samplers. Extensive experiments show that TGFlow reduces transport costs and improves both sampling efficiency and generation quality without compromising the simplicity and flexibility of Flow Matching. This work may inspire a new research direction centered on designing source distributions better aligned with the target distribution.

AI USE STATEMENT

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of the LLM were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure that the results presented in this paper are reproducible. All code is anonymized and submitted as a .zip file in the supplementary materials to facilitate replication and verification. The experimental setup, including training procedures, model configurations, and hardware specifications, is described in detail in the Appendix E. In addition, the types of public datasets used in this work, such as ImageNet and CIFAR-10, are publicly available, ensuring consistent and reproducible evaluation results.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Mohammadreza Armandpour, Ali Sadeghian, Chunyuan Li, and Mingyuan Zhou. Partition-guided gans. *arXiv preprint arXiv:2104.00816*, 2021.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *ICML*, 2026.

Chen Chen, Pengsheng Guo, Liangchen Song, Jiasen Lu, Rui Qian, Tsu-Jui Fu, Xinze Wang, Wei Liu, Yinfei Yang, and Alex Schwing. Car-flow: Condition-aware reparameterization aligns source and target for better flow matching. *Advances in Neural Information Processing Systems*, 38: 94919–94945, 2026.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, volume 2024, pp. 47922–47945, 2024.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Antoine Collas, Ce Ju, Nicolas Salvy, and Bertrand Thirion. Riemannian flow matching for brain connectivity matrices via pullback geometry. *Advances in Neural Information Processing Systems*, 38:59589–59639, 2026.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Valentin De Bortoli, Iryna Korshunova, Andriy Mnih, and Arnaud Doucet. Schrodinger bridge flow for unpaired data translation. *Advances in Neural Information Processing Systems*, 37:103384–103441, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 67350–67395, 2026.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026.

Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. Mgan: Training generative adversarial nets with multiple generators. *ICLR*, 2018.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

Junwan Kim, Jiho Park, Seonghu Jeon, and Seungryong Kim. Better source, better flow: Learning condition-dependent source distribution for flow matching. *arXiv preprint arXiv:2602.05951*, 2026.

Nikita Kornilov, Petr Mokrov, Alexander Gasnikov, and Alexander Korotin. Optimal flow matching: Learning straight trajectories in just one step. *Advances in Neural Information Processing Systems*, 37:104180–104204, 2024.

Junho Lee, Kwanseok Kim, and Joonseok Lee. Geometry-aware image flow matching. *ICML*, 2026.

Ziyun Li, Huancheng Hu, Soon Hoe Lim, Xuyu Li, Fei Gao, Enmao Diao, Zezhen Ding, Michalis Vazirgiannis, and Henrik Bostrom. A kinetic energy perspective of flow matching. *arXiv preprint arXiv:2602.07928*, 2026.

Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. Leapalign: Post-training flow matching models at any generation step by building two-step trajectories. *CVPR*, 2026.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. *ICLR*, 2023.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

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.

Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *International conference on machine learning*, pp. 4183–4192. PMLR, 2019.

Gaoxiang Luo, Frank Cole, Sihang Zhang, Yuxiang Wan, Yulong Lu, and Ju Sun. Flow matching for multimodal distributions. *CVPR*, 2026.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo i Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. *ICML*, 2023.

Mehrsa Pourya, Bassam El Rawas, and Michael Unser. Flower: A flow-matching solver for inverse problems. In *International Conference on Learning Representations*, volume 2026, pp. 91318–91343, 2026.

Alexander Sage, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Logo synthesis and manipulation with clustered generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5879–5888, 2018.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *International Conference on Learning Representations*, volume 2025, pp. 16397–16429, 2025.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in neural information processing systems*, 36:62183–62223, 2023.

Xiaozhuang Song and Tianshu Yu. Physically-informed flow matching with graph neural networks for complex fluid dynamics. *AAAI 2026*, 40(2):1024–1032, 2026.

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

Sophia Tang, Yinuo Zhang, Alexander Tong, and Pranam Chatterjee. Branched schrödinger bridge matching. *ArXiv*, pp. arXiv–2506, 2026.

Yoad Tewel, Yuval Atzmon, Gal Chechik, and Lior Wolf. Unpaired visual editing with self-consistent flow matching. *ICML*, 2026.

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.

Alexander Tong, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. *International Conference on Artificial Intelligence and Statistics, Vol 238*, 238, 2024.

Binxu Wang and John J Vastola. The unreasonable effectiveness of gaussian score approximation for diffusion models and its applications. *arXiv preprint arXiv:2412.09726*, 2024.

Binxu Wang, Jacob Zavatone-Veth, and Cengiz Pehlevan. A random matrix theory perspective on the consistency of diffusion models. *ICML*, 2026a.

Zifan Wang, Alice Harting, Matthieu Barreau, Michael Zavlanos, and Karl H Johansson. Source-guided flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 106484–106514, 2026b.

Olga Zaghen, Floor Eijkelboom, Alison Pouplin, Cong Liu, Max Welling, Jan-Willem van de Meent, and Erik Bekkers. Riemannian variational flow matching for material and protein design. In *International Conference on Learning Representations*, volume 2026, pp. 34243–34286, 2026.

Qunxi Zhu and Wei Lin. Switched flow matching: Eliminating singularities via switching odes. *ICML*, 2024.

A THEOREM ROOFS

A.1 PROOF OF THEOREM 1

Under the global Lipschitz and linear-growth assumptions on v, the characteristic ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}X_{s,t}(x) = v\big(t, X_{s,t}(x)\big), \qquad X_{s,s}(x) = x, \tag{22}$$

admits a unique global solution on $[0,1]$. By uniqueness of the characteristic ODE, the flow maps satisfy the composition property

$$X_{s,t} \circ X_{r,s} = X_{r,t}, \qquad r,s,t \in [0,1]. \tag{23}$$

Setting $(r,s,t) = (0,t,0)$ and $(r,s,t) = (t,0,t)$ in equation 23, respectively, gives

$$X_{t,0} \circ X_{0,t} = X_{0,0} = \mathrm{I}_{\mathbb{R}^d}, \qquad X_{0,t} \circ X_{t,0} = X_{t,t} = \mathrm{I}_{\mathbb{R}^d}\,. \tag{24}$$

Therefore, $X_t := X_{0,t}$ is invertible and $X_t^{-1} = X_{t,0}$. For any test function $\varphi \in C_c^\infty(\mathbb{R}^d)$, the definition $p_t = (X_t)_\# p_0$ gives

$$\int_{\mathbb{R}^d} \varphi(y)\, p_t(\mathrm{d}y) = \int_{\mathbb{R}^d} \varphi\big(X_t(x)\big)\, p_0(\mathrm{d}x). \tag{25}$$

Differentiating equation 25 with respect to t and using equation 22, we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^d} \varphi(y)\, p_t(\mathrm{d}y) &= \int_{\mathbb{R}^d} \nabla\varphi\big(X_t(x)\big) \cdot v\big(t, X_t(x)\big)\, p_0(\mathrm{d}x) \\ &= \int_{\mathbb{R}^d} \nabla\varphi(y) \cdot v(t,y)\, p_t(\mathrm{d}y).\end{aligned} \tag{26}$$

Equation equation 26 is the weak form of equation 6. Moreover, since $X_0 = \mathrm{I}$,

$$p_{t=0} = (X_0)_\# p_0 = p_0. \tag{27}$$

Under the assumed regularity conditions on v, the measure-valued solution of the continuity equation is unique. Hence,

$$p(\cdot,t) = (X_t)_\# p_0 = \mathcal{F}_t(p_0; v), \qquad t \in [0,1]. \tag{28}$$

This proves the first claim. For the second claim, fix any $t \in [0,1]$ and suppose that

$$\mathcal{F}_t(p_0^{(1)}; v) = \mathcal{F}_t(p_0^{(2)}; v). \tag{29}$$

By the definition of $\mathcal{F}_t$, equation equation 29 is equivalent to

$$(X_t)_\# p_0^{(1)} = (X_t)_\# p_0^{(2)}. \tag{30}$$

Applying $(X_t^{-1})_\#$ to both sides and using equation 24 and the composition rule for pushforward measures gives

$$\begin{aligned}p_0^{(1)} &= (X_t^{-1})_\#\big((X_t)_\# p_0^{(1)}\big) \\ &= (X_t^{-1})_\#\big((X_t)_\# p_0^{(2)}\big) = p_0^{(2)}.\end{aligned} \tag{31}$$

Thus,

$$\mathcal{F}_t(p_0^{(1)}; v) = \mathcal{F}_t(p_0^{(2)}; v) \quad \Longrightarrow \quad p_0^{(1)} = p_0^{(2)}. \tag{32}$$

Taking the contrapositive of equation 32 yields

$$p_0^{(1)} \neq p_0^{(2)} \quad \Longrightarrow \quad \mathcal{F}_t(p_0^{(1)}; v) \neq \mathcal{F}_t(p_0^{(2)}; v). \tag{33}$$

Since $t \in [0,1]$ is arbitrary, the result holds for every $t \in [0,1]$.

A.2 PROOF OF THEOREM 2

Let $\Sigma \in \mathbb{R}^{d\times d}$ be a fixed symmetric positive definite matrix, i.e., $\Sigma = \Sigma^\top$ and $\Sigma \succ 0$. Consider the source distribution

$$p_0(x_0) = \mathcal{N}(x_0 \mid 0, \Sigma), \tag{34}$$

and the conditional probability path

$$p_t(x \mid x_1) = \mathcal{N}\left(x \mid \mu_t(x_1), \sigma_t^2(x_1)\Sigma\right), \tag{35}$$

where $\sigma_t(x_1) > 0$ and $\mu_t(x_1)$ are differentiable with respect to t. The matrix Σ is assumed to be independent of both t and x_1.

For a sample $x_0 \sim p_0$, define the conditional flow map as

$$\mathcal{T}_t(x_0 \mid x_1) = \sigma_t(x_1)x_0 + \mu_t(x_1). \tag{36}$$

where $x_0 \sim \mathcal{N}(0, \Sigma)$. Since $\sigma_t(x_1)$ and $\mu_t(x_1)$ are deterministic for fixed t and x_1, the conditional mean of x is

$$\begin{aligned}\mathbb{E}[x \mid x_1] &= \mathbb{E}\left[\sigma_t(x_1)x_0 + \mu_t(x_1) \mid x_1\right] \\ &= \sigma_t(x_1)\mathbb{E}[x_0] + \mu_t(x_1) \\ &= \mu_t(x_1).\end{aligned} \tag{37}$$

Moreover, the conditional covariance of x is

$$\begin{aligned}\mathrm{Cov}(x \mid x_1) &= \mathrm{Cov}\left(\sigma_t(x_1)x_0 + \mu_t(x_1) \mid x_1\right) \\ &= \sigma_t^2(x_1)\,\mathrm{Cov}(x_0) \\ &= \sigma_t^2(x_1)\Sigma.\end{aligned} \tag{38}$$

Because x is an affine transformation of the Gaussian random variable x_0, it is also Gaussian. Combining equation 37 and equation 38, we obtain

$$\mathcal{T}_t(x_0 \mid x_1) \sim \mathcal{N}\left(\mu_t(x_1), \sigma_t^2(x_1)\Sigma\right). \tag{39}$$

Therefore,

$$\left(\mathcal{T}_t(\cdot \mid x_1)\right)_{\#} p_0 = p_t(\cdot \mid x_1) = \mathcal{N}\left(\cdot \mid \mu_t(x_1), \sigma_t^2(x_1)\Sigma\right). \tag{40}$$

Hence, the same affine conditional flow map in equation 36 transports the source distribution $\mathcal{N}(0, \Sigma)$ to the conditional distribution in equation 35. In particular, the analytical form of the conditional flow map is unchanged by replacing I with Σ.

Since $\sigma_t(x_1) > 0$, the flow map is invertible. Its inverse is

$$\mathcal{T}_t^{-1}(x \mid x_1) = \frac{x - \mu_t(x_1)}{\sigma_t(x_1)}. \tag{41}$$

Notably, equation 41 does not depend on Σ. Keeping x_0 and x_1 fixed, the time derivative of the flow map is

$$\partial_t \mathcal{T}_t(x_0 \mid x_1) = \sigma_t'(x_1)x_0 + \mu_t'(x_1). \tag{42}$$

By the definition of the induced conditional velocity field,

$$u_t(x \mid x_1) = \partial_t \mathcal{T}_t\left(\mathcal{T}_t^{-1}(x \mid x_1) \mid x_1\right). \tag{43}$$

Substituting equation 41 and equation 42 into equation 43 yields

$$\begin{aligned}u_t(x \mid x_1) &= \sigma_t'(x_1)\frac{x - \mu_t(x_1)}{\sigma_t(x_1)} + \mu_t'(x_1) \\ &= \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}\left(x - \mu_t(x_1)\right) + \mu_t'(x_1).\end{aligned} \tag{44}$$

Thus, the induced conditional velocity field is independent of Σ and has exactly the same analytical form as that obtained with the identity covariance matrix.

The TGFlow objective under the generalized source distribution is

$$\mathcal{L}_{\mathrm{TGFlow}}^{\Sigma}(\theta) = \mathbb{E}_{t, x_1,\, x_0\sim\mathcal{N}(0,\Sigma)}\left[\left\|v_\theta\left(\mathcal{T}_t(x_0 \mid x_1), t \mid x_1\right) - \partial_t \mathcal{T}_t(x_0 \mid x_1)\right\|_2^2\right]. \tag{45}$$

Using equation 43, we have

$$\partial_t \mathcal{T}_t(x_0 \mid x_1) = u_t\left(\mathcal{T}_t(x_0 \mid x_1) \mid x_1\right). \tag{46}$$

Substituting equation 46 into equation 45 gives

$$\mathcal{L}^{\Sigma}_{\mathrm{TGFlow}}(\theta) = \mathbb{E}_{t,x_1,\, x_0\sim\mathcal{N}(0,\Sigma)}\left[\left\|v_\theta\left(\mathcal{T}_t(x_0 \mid x_1), t \mid x_1\right) - u_t\left(\mathcal{T}_t(x_0 \mid x_1) \mid x_1\right)\right\|_2^2\right]. \tag{47}$$

Because $x = \mathcal{T}_t(x_0 \mid x_1) \sim p_t(\cdot \mid x_1)$, equation 47 can be equivalently written as

$$\mathcal{L}^{\Sigma}_{\mathrm{TGFlow}}(\theta) = \mathbb{E}_{t,x_1,\, x\sim p_t(\cdot|x_1)}\left[\left\|v_\theta(x, t \mid x_1) - u_t(x \mid x_1)\right\|_2^2\right]. \tag{48}$$

Therefore, replacing I in both the source distribution and the conditional probability path with the same fixed symmetric positive definite matrix Σ preserves the analytical forms of the conditional flow map, the induced conditional velocity field, and the TGFlow training objective.

A.3 Proof of theorem 3

Let d be the dimension of the sample space, and write $s := \sigma_{\min}^2 > 0$, $m := \lambda_{\max}(D) > 0$. Let $\lambda_1, \ldots, \lambda_d$ be the eigenvalues of D. Since D is symmetric positive definite, there exists an orthogonal matrix U such that

$$D = U\,\mathrm{diag}(\lambda_1, \ldots, \lambda_d)U^\top, \quad 0 < \lambda_i \le m. \tag{49}$$

Set $\alpha := 1 + s/m$. By the convolution property of Gaussian distributions,

$$\widetilde{P}_\Lambda = \mathcal{N}(0, D + s\Lambda) = \mathcal{N}(0, \alpha D), \qquad \widetilde{P}_I = \mathcal{N}(0, D + sI). \tag{50}$$

Consider the orthogonal change of coordinates $y = U^\top x$. Since U is orthogonal, this change of variables has Jacobian determinant of absolute value one and does not alter the KL divergence. In these coordinates, all three distributions factor into products of one-dimensional Gaussian distributions. In particular, the variance of the i-th coordinate is respectively λ_i, $\alpha\lambda_i$, and $\lambda_i + s$ under q, $\widetilde{P}_\Lambda$, and $\widetilde{P}_I$.

For $v > 0$, denote the density of $\mathcal{N}(0, v)$ by

$$\varphi_v(y) := \frac{1}{\sqrt{2\pi v}} \exp\left(-\frac{y^2}{2v}\right). \tag{51}$$

The log-density ratio for the i-th coordinate of $\widetilde{P}_\Lambda$ and q is

$$\log \frac{\varphi_{\alpha\lambda_i}(y)}{\varphi_{\lambda_i}(y)} = -\frac{1}{2}\log\alpha + \frac{y^2}{2}\left(\frac{1}{\lambda_i} - \frac{1}{\alpha\lambda_i}\right). \tag{52}$$

Taking expectation with respect to $y \sim \mathcal{N}(0, \alpha\lambda_i)$, for which $\mathbb{E}[y^2] = \alpha\lambda_i$, gives

$$\int_{\mathbb{R}} \varphi_{\alpha\lambda_i}(y) \log \frac{\varphi_{\alpha\lambda_i}(y)}{\varphi_{\lambda_i}(y)}\,\mathrm{d}y = \frac{1}{2}\big(\alpha - 1 - \log\alpha\big). \tag{53}$$

Summing over the d coordinates, we obtain

$$\mathrm{KL}(\widetilde{P}_\Lambda \parallel q) = \frac{d}{2}\left[\frac{s}{m} - \log\left(1 + \frac{s}{m}\right)\right]. \tag{54}$$

Similarly, the log-density ratio for the i-th coordinate of $\widetilde{P}_I$ and q is

$$\log \frac{\varphi_{\lambda_i+s}(y)}{\varphi_{\lambda_i}(y)} = -\frac{1}{2}\log\left(1 + \frac{s}{\lambda_i}\right) + \frac{y^2}{2}\left(\frac{1}{\lambda_i} - \frac{1}{\lambda_i + s}\right). \tag{55}$$

Taking expectation with respect to $y \sim \mathcal{N}(0, \lambda_i + s)$ yields

$$\int_{\mathbb{R}} \varphi_{\lambda_i+s}(y) \log \frac{\varphi_{\lambda_i+s}(y)}{\varphi_{\lambda_i}(y)}\,\mathrm{d}y = \frac{1}{2}\left[\frac{s}{\lambda_i} - \log\left(1 + \frac{s}{\lambda_i}\right)\right]. \tag{56}$$

Consequently,

$$\mathrm{KL}(\widetilde{P}_I \parallel q) = \frac{1}{2}\sum_{i=1}^{d}\left[\frac{s}{\lambda_i} - \log\left(1+\frac{s}{\lambda_i}\right)\right]. \tag{57}$$

Define $f(t) := t - \log(1+t)$ for $t > 0$. Since

$$f'(t) = \frac{t}{1+t} > 0, \tag{58}$$

f is strictly increasing. Because $\lambda_i \le m$ and $s > 0$,

$$f\left(\frac{s}{\lambda_i}\right) \ge f\left(\frac{s}{m}\right) \quad \text{for every } i. \tag{59}$$

It follows that

$$\mathrm{KL}(\widetilde{P}_I \parallel q) = \frac{1}{2}\sum_{i=1}^{d} f\left(\frac{s}{\lambda_i}\right) \ge \frac{d}{2} f\left(\frac{s}{m}\right) = \mathrm{KL}(\widetilde{P}_\Lambda \parallel q). \tag{60}$$

Equality holds if and only if $f(s/\lambda_i) = f(s/m)$ for every i. Since f is strictly increasing and $s > 0$, this is equivalent to $\lambda_i = m$ for every i, which in turn is equivalent to $D = mI_d$. Thus equality implies that $D = cI_d$ for some $c > 0$.

B Relate work

Flow Matching (Lipman et al., 2023) are trained by regressing toward the vector fields of fixed conditional probability paths. Related studies extend this framework from two perspectives: improving the velocity field or modifying the source distribution. For instance, Shi et al. (2023); Tong et al. (2024); Tang et al. (2026); De Bortoli et al. (2024) improve the training workflows for velocity fields in Schrödinger-bridge-based approaches. Gat et al. (2024) extend the modeling objective from the ODE velocity field to the drift–diffusion vector field in the SDE framework. Liu et al. (2022); Tong et al. (2023); Pooladian et al. (2023) optimize the velocity field by modifying the source-target sample coupling.

Among methods that modify the source distribution, prior work has predominantly focused on conditional generation, typically defining source distributions through predefined parameterizations, conditioning information, or geometric constraints. Specifically, Chen et al. (2026) uses a shifted and rescaled Gaussian distribution, while Kim et al. (2026); Lee et al. (2026); Wang et al. (2026b) construct source distributions by leveraging conditional information or imposing geometric constraints. In contrast, Luo et al. (2026) adopts a Gaussian mixture distribution. However, when applied to more general settings, these approaches remain limited in their ability to construct source distributions that faithfully characterize the statistical structure of the target distribution. Wang & Vastola (2024) constructs a score field from training data to approximate the neural score of diffusion models in the high-noise regime, accelerating sampling by skipping initial sampling steps. While this work demonstrates the value of target-data information for initializing generation within the diffusion framework, how to use such information to design better source distributions for Flow Matching remains underexplored. To address this gap, we propose TGFlow, which directly incorporates the target data's marginal second-order statistics into source distribution design within the Flow Matching framework. Our theoretical analysis demonstrates that the probability path in Flow Matching is jointly determined by the velocity field and the source distribution, while TGFlow preserves the analytical forms of the conditional velocity field and the loss function in Flow Matching.

C Additional proof details

C.1 Proof of total probability mass conservation

For any region $\Omega \subset \mathbb{R}^d$ in the state space, the probability mass within that region is defined as:

$$M(\Omega, t) = \int_\Omega p(x,t)\,\mathrm{d}x \tag{61}$$

Since the probability density function $p(x,t)$ is continuously differentiable with respect to time t and satisfies the commutativity condition between the integral and the time derivative, we have:

$$\frac{\mathrm{d}}{\mathrm{d}t}M(\Omega,t)=\int_{\Omega}\frac{\partial p(x,t)}{\partial t}\mathrm{d}x \tag{62}$$

The continuity equation is expressed as:

$$\frac{\partial p(x,t)}{\partial t}=-\nabla\cdot\boldsymbol{J}(x,t) \tag{63}$$

Substituting the above equation into the integral yields:

$$\frac{\mathrm{d}}{\mathrm{d}t}M(\Omega,t)=-\int_{\Omega}\nabla\cdot\boldsymbol{J}(x,t)\,\mathrm{d}x \tag{64}$$

Applying the Gauss divergence theorem to the vector field $J(x,t)$, the volume integral can be converted into a surface integral over the boundary of this region:

$$\int_{\Omega}\nabla\cdot\boldsymbol{J}(x,t)\,\mathrm{d}x=\oint_{\partial\Omega}\boldsymbol{J}(x,t)\cdot\boldsymbol{n}\,\mathrm{d}s \tag{65}$$

Substituting into Equation (5) yields the formula for the probability-mass conservation in this region:

$$\frac{\mathrm{d}}{\mathrm{d}t}M(\Omega,t)=-\oint_{\partial\Omega}\boldsymbol{J}(x,t)\cdot\boldsymbol{n}\,\mathrm{d}s \tag{66}$$

At the same time, we derive the global probability conservation property. Let the domain be the entire space $\Omega=\mathbb{R}^d$. Combining this with the decay condition at infinity $\lim_{\|x\|\to\infty}p(x,t)=0$, we obtain that the probability flux at the boundary at infinity is zero:

$$\oint_{\partial\mathbb{R}^d}\boldsymbol{J}(x,t)\cdot\boldsymbol{n}\,\mathrm{d}s=0 \tag{67}$$

Therefore, we can conclude that the global probability is conserved:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^d}p(x,t)\,\mathrm{d}x=0 \tag{68}$$

C.2 Non-uniqueness of transport fields

We show that a probability path does not uniquely determine its transport velocity field. In particular, any velocity component whose probability-weighted divergence vanishes can be added to the original velocity field without changing the induced probability evolution. Assume that the probability path $\{p_t\}_{t\in[0,1]}$ satisfies the continuity equation associated with the velocity field v_t:

$$\partial_t p_t(x)+\nabla\cdot\big(p_t(x)v_t(x)\big)=0. \tag{69}$$

Let $f_t:\mathbb{R}^d\to\mathbb{R}^d$ be an additional velocity field satisfying the p_t-weighted divergence-free condition

$$\nabla\cdot\big(p_t(x)f_t(x)\big)=0,\qquad (t,x)\in[0,1]\times\mathbb{R}^d. \tag{70}$$

Define the modified velocity field by

$$\widetilde{v}_t(x):=v_t(x)+f_t(x). \tag{71}$$

Using the linearity of the divergence operator, we have

$$\begin{aligned}\partial_t p_t+\nabla\cdot\big(p_t\widetilde{v}_t\big)&=\partial_t p_t+\nabla\cdot\big(p_t(v_t+f_t)\big)\\&=\partial_t p_t+\nabla\cdot(p_t v_t)+\nabla\cdot(p_t f_t)=0,\end{aligned} \tag{72}$$

where the first two terms vanish by equation 69, and the last term vanishes by equation 70. Therefore, the same probability path also satisfies

$$\partial_t p_t + \nabla \cdot \big(p_t \widetilde{v}_t\big) = 0. \tag{73}$$

Hence, v_t and $\widetilde{v}_t = v_t + f_t$ induce the same probability evolution. The same conclusion can be expressed in weak form. For any test function $\varphi \in C_c^\infty(\mathbb{R}^d)$, we obtain from equation 69

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^d} \varphi(x) p_t(x)\,\mathrm{d}x &= \int_{\mathbb{R}^d} \varphi(x)\partial_t p_t(x)\,\mathrm{d}x \\ &= -\int_{\mathbb{R}^d} \varphi(x)\nabla\cdot\big(p_t(x)v_t(x)\big)\,\mathrm{d}x \\ &= \int_{\mathbb{R}^d} \nabla\varphi(x)\cdot v_t(x)p_t(x)\,\mathrm{d}x,\end{aligned} \tag{74}$$

where the last equality follows from integration by parts and the compact support of φ. Similarly, equation 70 implies

$$\int_{\mathbb{R}^d} \nabla\varphi(x)\cdot f_t(x)p_t(x)\,\mathrm{d}x = -\int_{\mathbb{R}^d} \varphi(x)\nabla\cdot\big(p_t(x)f_t(x)\big)\,\mathrm{d}x = 0. \tag{75}$$

Adding equation 75 to equation 74 yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^d} \varphi(x) p_t(x)\,\mathrm{d}x &= \int_{\mathbb{R}^d} \nabla\varphi(x)\cdot\big(v_t(x)+f_t(x)\big)p_t(x)\,\mathrm{d}x \\ &= \int_{\mathbb{R}^d} \nabla\varphi(x)\cdot\widetilde{v}_t(x)p_t(x)\,\mathrm{d}x.\end{aligned} \tag{76}$$

Thus, the modified field $\widetilde{v}_t$ gives exactly the same weak evolution of p_t as the original field v_t. We next construct a class of nontrivial fields satisfying equation 70. Assume $d \geq 2$, and let $A \in \mathbb{R}^{d\times d}$ be a nonzero skew-symmetric matrix:

$$A^\top = -A. \tag{77}$$

Let $\psi_t \in C_c^\infty(\mathbb{R}^d)$ be supported in a region where $p_t(x) > 0$, and define

$$f_t(x) = \frac{1}{p_t(x)} A\nabla\psi_t(x). \tag{78}$$

Then the probability-weighted field satisfies

$$\begin{aligned}\nabla\cdot\big(p_t(x)f_t(x)\big) &= \nabla\cdot\big(A\nabla\psi_t(x)\big) \\ &= \sum_{i=1}^d\sum_{j=1}^d A_{ij}\partial_{ij}\psi_t(x).\end{aligned} \tag{79}$$

Since the Hessian of ψ_t is symmetric and A is skew-symmetric, we have

$$\sum_{i=1}^d\sum_{j=1}^d A_{ij}\partial_{ij}\psi_t = \frac{1}{2}\sum_{i=1}^d\sum_{j=1}^d (A_{ij}+A_{ji})\,\partial_{ij}\psi_t = 0. \tag{80}$$

Therefore,

$$\nabla\cdot(p_t f_t) = 0. \tag{81}$$

Because ψ_t can be selected from an infinite-dimensional space of smooth compactly supported functions, this construction produces infinitely many distinct p_t-weighted divergence-free fields. Hence, the family

$$\{v_t + f_t :\ \nabla\cdot(p_t f_t) = 0\} \tag{82}$$

contains infinitely many velocity fields that generate the same probability path $\{p_t\}_{t\in[0,1]}$. As a concrete example, suppose that p_t is rotation invariant. Then there exists a scalar function ρ_t such that

$$p_t(x) = \rho_t(\|x\|). \tag{83}$$

Let

$$f_t(x) = A_t x, \qquad A_t^\top = -A_t. \tag{84}$$

For $x \neq 0$, the gradient of the rotation-invariant density is radial:

$$\nabla p_t(x) = \rho_t'(\|x\|)\frac{x}{\|x\|}. \tag{85}$$

Moreover, the skew-symmetry of A_t implies

$$\begin{aligned} x^\top A_t x &= \left(x^\top A_t x\right)^\top \\ &= x^\top A_t^\top x \\ &= -x^\top A_t x, \end{aligned} \tag{86}$$

and therefore

$$x^\top A_t x = 0. \tag{87}$$

Consequently,

$$\nabla p_t(x) \cdot f_t(x) = \rho_t'(\|x\|)\frac{x^\top A_t x}{\|x\|} = 0. \tag{88}$$

At the same time, the divergence of the rotational field is

$$\nabla \cdot f_t(x) = \nabla \cdot (A_t x) = \mathrm{tr}(A_t) = 0, \tag{89}$$

where the last equality follows from the fact that every skew-symmetric matrix has zero diagonal trace. Thus,

$$\nabla \cdot (p_t f_t) = \nabla p_t \cdot f_t + p_t \nabla \cdot f_t = 0. \tag{90}$$

Therefore, any rotational component of the form $f_t(x) = A_t x$ can be added to the original velocity field without changing the transported density:

$$\partial_t p_t + \nabla \cdot (p_t v_t) = 0 \quad \Longrightarrow \quad \partial_t p_t + \nabla \cdot \big(p_t(v_t + A_t x)\big) = 0. \tag{91}$$

Hence, a given probability path generally does not uniquely determine its transport velocity field. In particular, infinitely many distinct velocity fields can generate the same probability evolution.

C.3 Push-forward of target-guided Gaussian source

Fix a target sample $x_1 \in \mathbb{R}^d$ and a time $t \in [0, 1]$. To simplify the notation, let

$$\mu_t := \mu_t(x_1), \qquad \sigma_t := \sigma_t(x_1). \tag{92}$$

Assume that $\Lambda \in \mathbb{R}^{d\times d}$ is symmetric positive definite and that $\sigma_t > 0$. The source density is given by

$$p_0(x_0) = \mathcal{N}(x_0 \mid 0, \Lambda) = \frac{1}{(2\pi)^{d/2}|\Lambda|^{1/2}} \exp\left(-\frac{1}{2} x_0^\top \Lambda^{-1} x_0\right). \tag{93}$$

For the fixed condition x_1, consider the affine conditional flow map

$$\mathcal{T}_t(x_0 \mid x_1) = \sigma_t x_0 + \mu_t. \tag{94}$$

Let $x = \mathcal{T}_t(x_0 \mid x_1)$. Since $\sigma_t > 0$, the map $\mathcal{T}_t(\cdot \mid x_1)$ is invertible. Solving the preceding equation for x_0 gives

$$x_0 = \mathcal{T}_t^{-1}(x \mid x_1) = \frac{x - \mu_t}{\sigma_t}. \tag{95}$$

The Jacobian matrix of the inverse map with respect to x is

$$D_x \mathcal{T}_t^{-1}(x \mid x_1) = \frac{\partial}{\partial x}\left(\frac{x - \mu_t}{\sigma_t}\right).$$

Since μ_t and σ_t are fixed with respect to x, we obtain

$$D_x \mathcal{T}_t^{-1}(x \mid x_1) = \frac{1}{\sigma_t} I_d. \tag{96}$$

Its determinant is therefore

$$\det D_x \mathcal{T}_t^{-1}(x \mid x_1) = \det\left(\frac{1}{\sigma_t} I_d\right) = \left(\frac{1}{\sigma_t}\right)^d. \tag{97}$$

Because $\sigma_t > 0$, the absolute Jacobian determinant is

$$\left|\det D_x \mathcal{T}_t^{-1}(x \mid x_1)\right| = \sigma_t^{-d}. \tag{98}$$

By the multivariate change-of-variables formula, the push-forward density at x is

$$p_t(x \mid x_1) = p_0\left(\mathcal{T}_t^{-1}(x \mid x_1)\right)\left|\det D_x \mathcal{T}_t^{-1}(x \mid x_1)\right| \tag{99}$$

$$= p_0\left(\frac{x-\mu_t}{\sigma_t}\right)\sigma_t^{-d}. \tag{100}$$

Substituting the Gaussian source density into this expression yields

$$p_t(x \mid x_1) = \frac{\sigma_t^{-d}}{(2\pi)^{d/2}|\Lambda|^{1/2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_t}{\sigma_t}\right)^\top \Lambda^{-1}\left(\frac{x-\mu_t}{\sigma_t}\right)\right]. \tag{101}$$

Because σ_t is a scalar, the quadratic form in the exponent can be simplified as

$$\left(\frac{x-\mu_t}{\sigma_t}\right)^\top \Lambda^{-1}\left(\frac{x-\mu_t}{\sigma_t}\right) = \frac{1}{\sigma_t}(x-\mu_t)^\top \Lambda^{-1}\frac{1}{\sigma_t}(x-\mu_t) \tag{102}$$

$$= \frac{1}{\sigma_t^2}(x-\mu_t)^\top \Lambda^{-1}(x-\mu_t). \tag{103}$$

Consequently,

$$p_t(x \mid x_1) = \frac{1}{(2\pi)^{d/2}\sigma_t^d|\Lambda|^{1/2}} \exp\left[-\frac{1}{2\sigma_t^2}(x-\mu_t)^\top \Lambda^{-1}(x-\mu_t)\right]. \tag{104}$$

It remains to show that this expression is precisely the density of a Gaussian distribution with covariance $\sigma_t^2\Lambda$. Define

$$\Sigma_t = \sigma_t^2\Lambda. \tag{105}$$

Since Λ is symmetric positive definite and $\sigma_t > 0$, Σ_t is also symmetric positive definite. Its inverse is

$$\Sigma_t^{-1} = (\sigma_t^2\Lambda)^{-1} = \frac{1}{\sigma_t^2}\Lambda^{-1}. \tag{106}$$

Moreover, using the identity

$$|c\Lambda| = c^d|\Lambda|$$

for a scalar c and a $d \times d$ matrix Λ, we have

$$|\Sigma_t| = |\sigma_t^2\Lambda| = \sigma_t^{2d}|\Lambda|. \tag{107}$$

Taking the square root gives

$$|\Sigma_t|^{1/2} = \sigma_t^d|\Lambda|^{1/2}, \tag{108}$$

where we again use $\sigma_t > 0$. Therefore, the density can be rewritten as

$$p_t(x \mid x_1) = \frac{1}{(2\pi)^{d/2}|\Sigma_t|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_t)^\top \Sigma_t^{-1}(x-\mu_t)\right] \tag{109}$$

$$= \mathcal{N}(x \mid \mu_t, \Sigma_t). \tag{110}$$

Substituting back $\mu_t = \mu_t(x_1)$ and $\Sigma_t = \sigma_t(x_1)^2\Lambda$, we obtain

$$p_t(x \mid x_1) = \mathcal{N}\left(x \mid \mu_t(x_1), \sigma_t(x_1)^2\Lambda\right). \tag{111}$$

Hence,

$$\left[\mathcal{T}_t(\cdot \mid x_1)\right]_{\#} p_0 = p_t(\cdot \mid x_1), \tag{112}$$

which proves that the affine conditional flow map pushes the target-guided Gaussian source $\mathcal{N}(0, \Lambda)$ forward to the conditional Gaussian path $\mathcal{N}(\mu_t(x_1), \sigma_t(x_1)^2\Lambda)$.

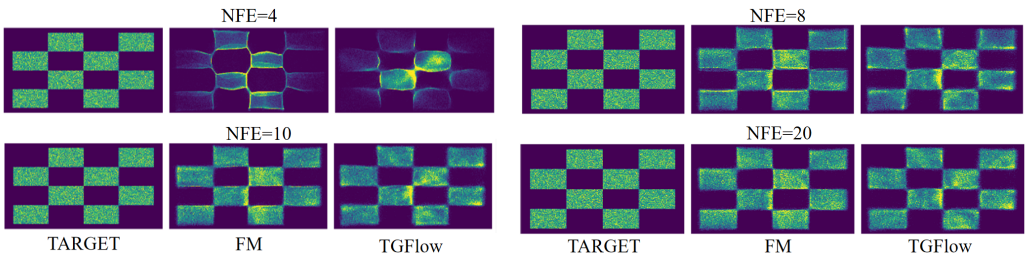

Figure 6: Generation comparison of TGFlow and FM on 2D checkerboard data with NFE $= 4$, NFE $= 10$, and NFE $= 20$.

D CONDITIONAL GENERATION EXPERIMENTS

We further evaluate the conditional image generation performance of TGFlow on the ImageNet dataset. Specifically, we conduct an upsampling task that reconstructs 128×128-resolution images from 32×32 low-resolution inputs. During inference, TGFlow takes low-resolution images as conditional constraints and outputs the corresponding high-resolution images via the learned conditional transport process. We follow the experimental configuration from Lipman et al. (2023), the visualization results are presented in Figure 11.

E EXPERIMENTS DETAILS

E.1 2D SYNTHETIC DATASETS EXPERIMENT DETAILS

Following Lipman et al. (2023), we compare TGFlow with FM on the two-dimensional checkerboard dataset. Both methods use the same five-layer MLP with 512 hidden units per layer and share identical training parameters, optimization settings, training budgets, and numerical solvers. FM uses the standard Gaussian source distribution, whereas TGFlow constructs its target-guided source distribution using the method described in Section 3. We visualize the learned transport trajectories at selected intermediate time points using the same sampling and visualization protocol for both methods. Meanwhile, Figure 6 presents the generation results of TGFlow and FM on 2D checkerboard data at different NFE values.

E.2 2D SYNTHETIC DATASETS OPTIMAL TRANSPORT EXPERIMENT DETAILS

We evaluate TGFlow on the two-dimensional transport benchmark of Tong et al. (2024). The method of Tong et al. (2024) is used as the baseline. Unless otherwise specified, both methods share the same datasets, data splits, network architecture, optimization settings, training budget, numerical solver, and evaluation protocol.

Datasets and transport tasks. Following Tong et al. (2024), we consider four two-dimensional transport tasks: $\mathcal{N}\rightarrow$8gaussians , $\mathcal{N}\rightarrow$moons, $\mathcal{N}\rightarrow$scurve and moons$\rightarrow$8gaussians. Here, $\mathcal{N}$ denotes the two-dimensional standard Gaussian distribution. For each task, we use 10,000 training samples, 10,000 validation samples, and 10,000 test samples. The data generation and preprocessing procedures follow Tong et al. (2024).

Compared methods and source distributions. We compare TGFlow with the method of Tong et al. (2024). The baseline uses the original source distribution of each task, whereas TGFlow uses the target-guided source distribution described in Section 3. For the moons$\rightarrow$8gaussians task, the original source is the non-Gaussian `moons` distribution. Both methods use the same target data and data splits.

Table 3: Comparison on ImageNet 128×128. Due to the prohibitive computational cost, the baseline does not report NLL in the original paper, so NLL is omitted in this table.

	ImageNet 128×128	
Model	NLL↓	FID↓
MGAN (Hoang et al., 2018)	–	58.9
PacGAN2 (Lin et al., 2018)	–	57.5
Logo-GAN-AE (Sage et al., 2018)	–	50.9
Self-cond. GAN (Lučić et al., 2019)	–	41.7
Uncond. BigGAN (Lučić et al., 2019)	–	25.3
PGMGAN (Armandpour et al., 2021)	–	21.7
FM/ OT (Lipman et al., 2023)	2.90	20.9
TGFlow(Ours)	**2.35**	**15.07**

Conditional path and coupling. For each pair (x_0, x_1), we sample $t \sim \mathcal{U}(0,1)$. Following Tong et al. (2023), the baseline uses the Brownian-bridge marginal

$$p_t(x \mid x_0, x_1) = \mathcal{N}\left(x;\, (1-t)x_0 + tx_1,\, \sigma^2 t(1-t)I\right), \qquad \sigma = 1,$$

from which

$$x_t = (1-t)x_0 + tx_1 + \sigma\sqrt{t(1-t)}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

TGFlow uses the target-guided conditional path defined in Section 3. For both methods, source and target minibatches of size 512 are coupled using exact minibatch optimal transport with the squared Euclidean cost

$$c(x_0, x_1) = \|x_0 - x_1\|_2^2.$$

Training objective. Both methods use the conditional flow-matching objective

$$\mathcal{L}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\|v_\theta(t, x_t) - u_t(x_t \mid x_0, x_1)\|_2^2\right], \tag{113}$$

where v_θ is the learnable velocity field and $u_t(x_t \mid x_0, x_1)$ is the conditional target velocity associated with the selected probability path. The two methods have the same objective form but use different source-distribution constructions.

The minibatch size is 512, and each model is trained for 1,000 epochs. The same architecture and optimization budget are used for TGFlow and the baseline.

Sampling. At inference time, samples are initialized from the corresponding source distribution and transported from $t = 0$ to $t = 1$ according to

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(t, x_t). \tag{114}$$

We use the explicit Euler method with 100 uniformly spaced integration steps:

$$\Delta t = \frac{1}{100}, \qquad x_{t+\Delta t} = x_t + \Delta t\, v_\theta(t, x_t). \tag{115}$$

For each method, we generate 10,000 endpoint samples.

Evaluation metrics. We evaluate endpoint generation using the empirical 2-Wasserstein distance. Let $\widehat{p}_1$ and $\widehat{q}_1$ denote the empirical distributions of the generated samples and target test samples, respectively:

$$W_2(\widehat{p}_1, \widehat{q}_1) = \left[\min_{\pi \in \Pi(\widehat{p}_1, \widehat{q}_1)} \int \|x - y\|_2^2 \,\mathrm{d}\pi(x, y)\right]^{1/2}. \tag{116}$$

A smaller W_2 indicates better agreement with the target distribution. We also report the normalized path energy:

$$\mathrm{NPE}(p_0, p_1; v_\theta) = \frac{\left|\mathbb{E}_{x_0 \sim p_0}\left[\int_0^1 \|v_\theta(t, x_t)\|_2^2 \,\mathrm{d}t\right] - W_2^2(p_0, p_1)\right|}{W_2^2(p_0, p_1)}. \tag{117}$$

Table 4: FID and NFE values on CIFAR-10 under alternative parameter settings.

NFE / sample →	100	1000	Adaptive	
Algorithm ↓	FID	FID	FID	NFE
VP-FM (Lipman et al., 2023)	7.772	4.048	4.335	525.92
OT-FM (Lipman et al., 2023)	4.640	3.822	3.655	143.00
S.I. (Albergo et al., 2025)	4.488	4.132	4.009	146.12
I-CFM (Tong et al., 2023)	4.461	3.643	3.659	146.42
OT-CFM (Tong et al., 2023)	4.443	3.741	3.577	133.94
A.T.(Wang & Vastola, 2024)	2.235	2.463	2.011	**38.57**
TGFlow (Ours)	**2.179**	**1.968**	**1.642**	107.85

Here, x_t follows the probability-flow ODE in equation 114. A smaller NPE indicates that the learned transport energy is closer to the dynamic optimal-transport energy.

Statistical reporting. Each experiment is repeated with five independent random seeds. For metric values $m_1, \ldots, m_5$, we report

$$\overline{m} = \frac{1}{5}\sum_{k=1}^{5} m_k, \qquad s_m = \sqrt{\frac{1}{4}\sum_{k=1}^{5}(m_k - \overline{m})^2}. \tag{118}$$

All results are reported as $\overline{m} \pm s_m$. Lower values are better for both W_2 and NPE. The exact data generation and preprocessing procedures follow the implementation of Tong et al. (2024).

E.2.1 2D TRAJECTORY EXPERIMENTS

Construction of the target ground-truth distribution p_1. We construct an imbalanced three-cluster distribution in polar coordinates to mimic the manifold properties of high-dimensional images, with a dedicated low-density minor mode for observing the mode-discrepancy phenomenon. The cluster center angles (in radians) are given by $\boldsymbol{\theta}_{\text{center}} = [4.4956,\ 5.9167,\ 1.1018]$, and the per-cluster sampling probability weights are $\boldsymbol{w}_c = [0.05,\ 0.30,\ 0.65]$, where the 0-th cluster is the low-density minor mode with weight 0.05. To sample a single target sample, we first randomly draw a cluster index k from the categorical distribution parameterized by $\boldsymbol{w}_c$, retrieve the base angle $\theta_0 = \theta_{\text{center},k}$, add Gaussian angular perturbation via $\theta = \theta_0 + \mathcal{N}(0, 0.15^2)$, sample the radius as $r \sim \chi(\text{df} = 625)$, convert polar coordinates to Cartesian coordinates with $x = r\cos\theta$ and $y = r\sin\theta$, and finally output the 2-dimensional vector $\boldsymbol{x}_1 = [x, y]$. For batched sampling, given the number of samples N, the function outputs a target-sample tensor of shape $\mathbb{R}^{N\times 2}$.

Construction of the two source distributions p_0. Source a is the standard Gaussian (baseline), where $p_0 = \mathcal{N}(0, I)$. Source b is the Gaussian distribution matched to the variances of the ground-truth distribution: we first independently draw 20000 samples from the target distribution p_1, compute the marginal variances over these target samples (see Section 3.1), construct the diagonal covariance matrix Λ, and then sample TGFlow source samples following $\boldsymbol{x}_0 \sim \mathcal{N}(0, \Lambda)$.

Training pipeline. We adopt the conditional probability flow in equation 13, and use mini-batch optimal transport OT-CFM (equation 20) for conditional pairing. The conditional velocity field is given by $u_t = x_1 - x_0$. The MLP vector-field network takes a 3-dimensional vector $[x,\ y,\ t]$ as input; its architecture is $\text{Linear}(3 \to 128) \to \text{ReLU} \to \text{Linear}(128 \to 128) \to \text{ReLU} \to \text{Linear}(128 \to 2)$, which outputs the predicted velocity field $\boldsymbol{v}_\theta \in \mathbb{R}^2$. The CFM loss function is defined in equation 5. Both experiments share identical fixed training hyper-parameters: Adam optimizer with learning rate 10^{-4}, total training iterations of 20000, and mini-batch size of 16. At inference time, we employ the Dormand-Prince adaptive-step ODE solver over the integration interval $t \in [0, 1]$ with absolute tolerance $\text{abs_tol} = 10^{-4}$ and relative tolerance $\text{rel_tol} = 10^{-4}$; generated samples are obtained by integrating the ODE starting from source samples $\boldsymbol{x}_0$.

Evaluation metrics. For each experimental run, the model generates 2000 samples X_{gen} and we simultaneously sample 2000 held out ground truth target samples X_{gt}, smaller metric values indicate

Table 5: Quantitative evaluation for 2D flow trajectory experiments

Source	Method	AMD↓	FailureRate↓	MDD↓	Norm.W↓
Standard Gaussian	OT-CFM	0.32 ± 0.05	7.41 ± 1.76	0.1 ± 0.002	0.15 ± 0.06
TGFlow(Ours)	OT-CFM	0.26 ± 0.03	4.91 ± 1.21	0.003 ± 0.005	0.14 ± 0.02

better generation quality. For AMD (Average Minimum Distance),

$$\mathrm{AMD} = \frac{1}{N} \sum_{i=1}^{N} \min_{j} \|x_{\mathrm{gen},i} - x_{\mathrm{gt},j}\|_2, \tag{119}$$

where $N = 2000$, which averages the minimal L_2 distance from each generated sample to all ground-truth samples. For Failure Rate, a generated sample is marked as failed if its minimal L_2 distance to all ground-truth samples exceeds the threshold 1,

$$\mathrm{FailureRate} = \frac{\text{number of failed samples}}{N} \times 100\%. \tag{120}$$

For MDD (Mode Distribution Divergence), we pre-compute the Cartesian centers of the three modes by taking the expectation of $\chi(\mathrm{df} = 625)$ as radius for each $\theta_{\text{center},k}$ to obtain the mode-center matrix $M \in \mathbb{R}^{3\times 2}$, all samples (ground-truth and generated) are assigned to their nearest mode center according to L_2 distance. We compute mode-proportion vectors P for ground-truth samples and Q for generated samples, adding a small epsilon $\epsilon = 10^{-10}$ to all entries for numerical stability, then

$$\mathrm{MDD} = D_{\mathrm{KL}}(Q\|P) = \sum_{k=0}^{2} Q_k \log \frac{Q_k}{P_k}, \tag{121}$$

where higher MDD implies severe mode collapse or mode-discrepancy. For normalized Wasserstein-1 distance, we compute the two-dimensional Wasserstein-1 distance W_1 between generated and ground-truth samples via the exact optimal-transport solver `ot.emd2` from the POT library, calculate avg as the average pairwise L_2 distance among all ground-truth samples, and define the normalized metric

$$\mathrm{Norm.W1} = \frac{W_1}{\mathrm{avg}}. \tag{122}$$

where normalization removes global dataset-scale effects to enable cross-condition comparison of distribution discrepancy. Each source distribution setting is independently run 10 times with distinct random seeds. Metrics are reported in the form mean $\pm$ std. The final experimental results are shown in Table 5.

E.3 Implementation details for imagenet experiments

Following Lipman et al. (2023), we evaluate TGFlow on CIFAR-10 and ImageNet at resolutions 32×32, 64×64, and 128×128. We compare TGFlow with the methods considered in Flow Matching. TGFlow and the corresponding baselines use the same data preprocessing, network architectures, optimization settings, training budgets, and evaluation protocols. The only difference is that TGFlow constructs its source distribution using the target-guided method described in Section 3. Images are center-cropped, resized to the target resolution, and normalized to $[-1, 1]$. The dataset-specific configurations are summarized in Table 6. All models use the U-Net architecture of Dhariwal & Nichol (2021) and are optimized using Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and zero weight decay. CIFAR-10 and ImageNet-32 use full 32-bit precision, whereas ImageNet-64 and ImageNet-128 use mixed 16-bit precision. All TGFlow experiments are conducted on three NVIDIA H200 GPUs. Samples are generated by solving the learned ODE using the adaptive Dormand–Prince solver (`dopri5`) with $\texttt{atol} = \texttt{rtol} = 10^{-5}$. We report negative log-likelihood in bits per dimension (BPD), Fréchet Inception Distance (FID), and the number of function evaluations (NFE). FID and NFE are evaluated using $50{,}000$ generated samples. The same solver, tolerances, and evaluation code are used for TGFlow and all baselines.

Table 6: Hyperparameters used for training each model.

	CIFAR-10	**ImageNet-32**	**ImageNet-64**	**ImageNet-128**
Channels	256	256	192	256
Depth	2	3	3	3
Channel multipliers	$1, 2, 2, 2$	$1, 2, 2, 2$	$1, 2, 3, 4$	$1, 1, 2, 3, 4$
Heads	4	4	4	4
Channels per head	64	64	64	64
Attention resolutions	16	$16, 8$	$32, 16, 8$	$32, 16, 8$
Dropout	0.0	0.0	0.0	0.0
Effective batch size	256	1024	2048	1536
Hardware	$3 \times$ H200	$3 \times$ H200	$3 \times$ H200	$3 \times$ H200
Epochs	1000	200	100	80
Iterations	391K	250K	157K	500K
Learning rate	5×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Learning-rate schedule	Polynomial decay	Polynomial decay	Constant	Polynomial decay
Warm-up steps	45K	20K	–	20K

E.4 Additional experimental details for CIFAR-10

We evaluate TGFlow on unconditional CIFAR-10 generation and compare it with the baselines reported by Tong et al. (2023). Each baseline retains its original probability-path construction, while TGFlow uses the target-guided source construction described in Section 3. We use a U-Net with 128 base channels, depth 2, channel multipliers $[1, 2, 2, 2]$, four attention heads, 64 channels per head, attention resolution 16, and dropout 0.1. All models are trained for 400,000 iterations with batch size 128 using Adam with a constant learning rate of 2×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and zero weight decay. We clip the gradient norm to 1.0 and use exponential moving average weights with decay 0.9999. For the CFM-based baselines, the conditional-path bandwidth is set to $\sigma = 0$. All TGFlow experiments are conducted on three NVIDIA H200 GPUs. At inference time, we generate samples by integrating the learned ODE with fixed-step Euler solvers using 100 and 1,000 steps, and with the adaptive Dormand–Prince solver (`dopri5`) using $\texttt{atol} = \texttt{rtol} = 10^{-5}$. We report Fréchet Inception Distance (FID) for each solver. For adaptive integration, the number of function evaluations (NFE) is averaged over 100 batches of 500 generated samples. TGFlow and all baselines use identical solver settings and evaluation procedures.

E.5 Diversity evaluation details on ImageNet-128

We evaluate the final TGFlow and FM-0T models using Precision, Recall, Density, and Coverage. For each method, we generate $50{,}000$ ImageNet-128 samples from its final checkpoint. The generated samples of TGFlow and FM-0T are evaluated separately against the same $50{,}000$ ImageNet validation images. Both methods use the adaptive Dormand–Prince solver (`dopri5`) with $\texttt{atol} = \texttt{rtol} = 10^{-5}$. We extract 2048-dimensional pool3 features using a pretrained Inception-V3 network and compute PRDC metrics with $k = 5$ nearest neighbors. Higher values indicate better performance for all metrics.

E.6 Experimental details for comparison across different samplers

We evaluate the trained TGFlow and FM models on CIFAR-10 using the DPM-Solver++, Heun, and STORK samplers (Detail in Appendix E.3). The number of neural function evaluations (NFE) is fixed at 100, and $50{,}000$ images are generated using the same random seeds, with generation quality evaluated by FID. Specifically, Heun performs 50 second-order update steps, STORK uses 100 sampling steps with the solver order set to 2, and DPM-Solver++ employs a third-order single-step solver. During sampling, the model parameters remain fixed, with no additional training or fine-tuning.

F Additional figures

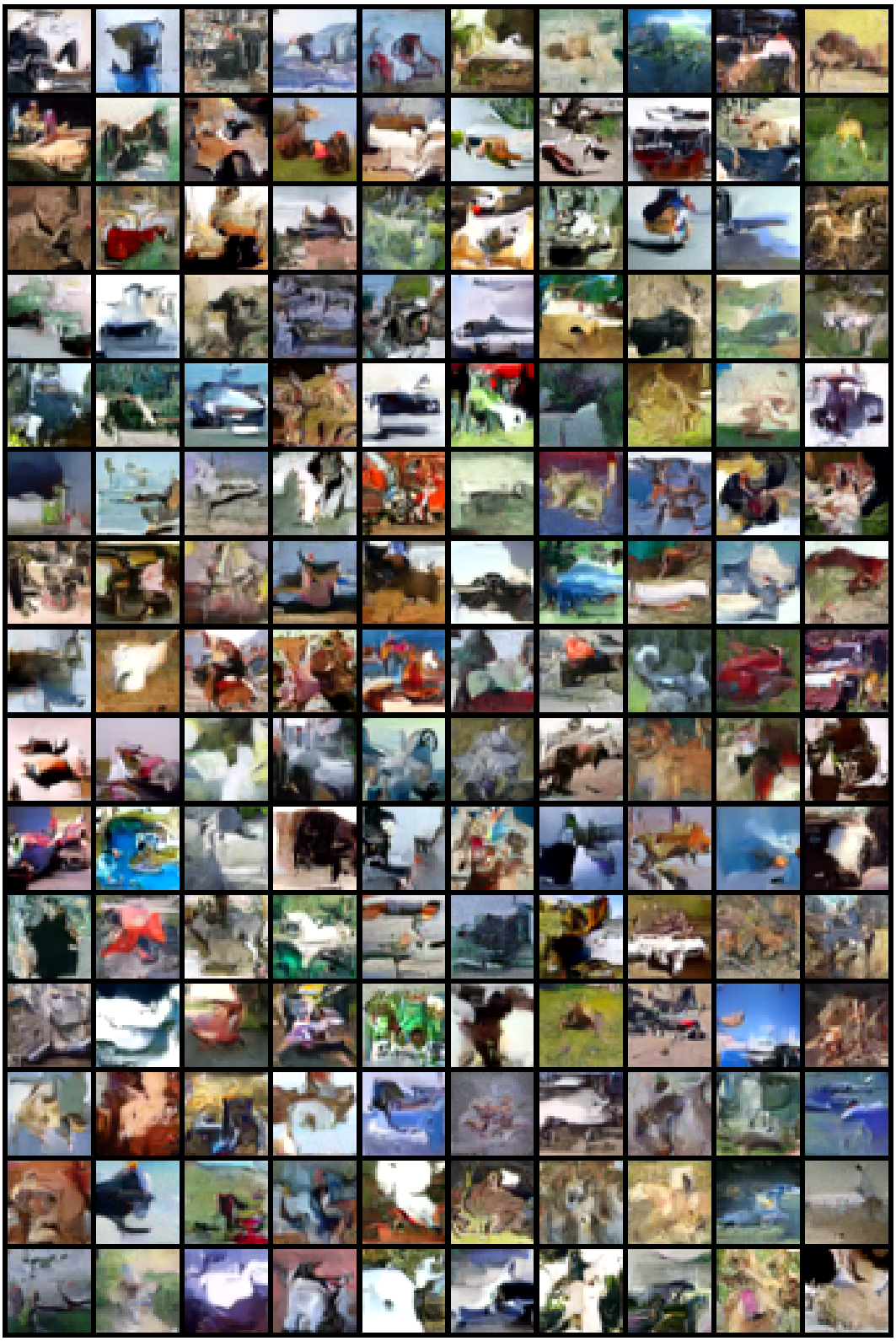

Figure 7: Non-curated unconditional CIFAR-10 generated images of a CNF trained with TGFlow-OT.

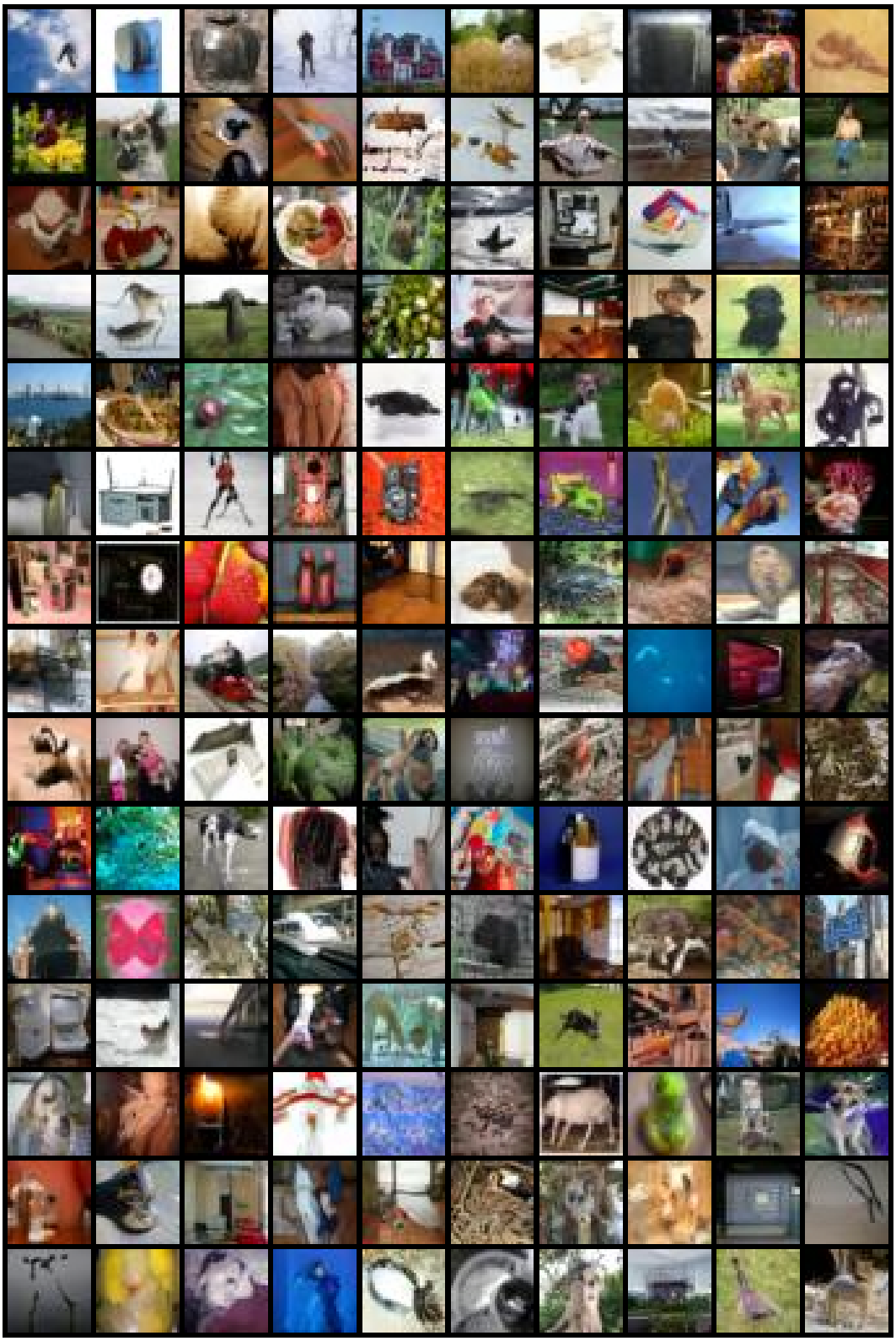

Figure 8: Non-curated unconditional ImageNet-32 generated images of a CNF trained with TGFlow-OT.

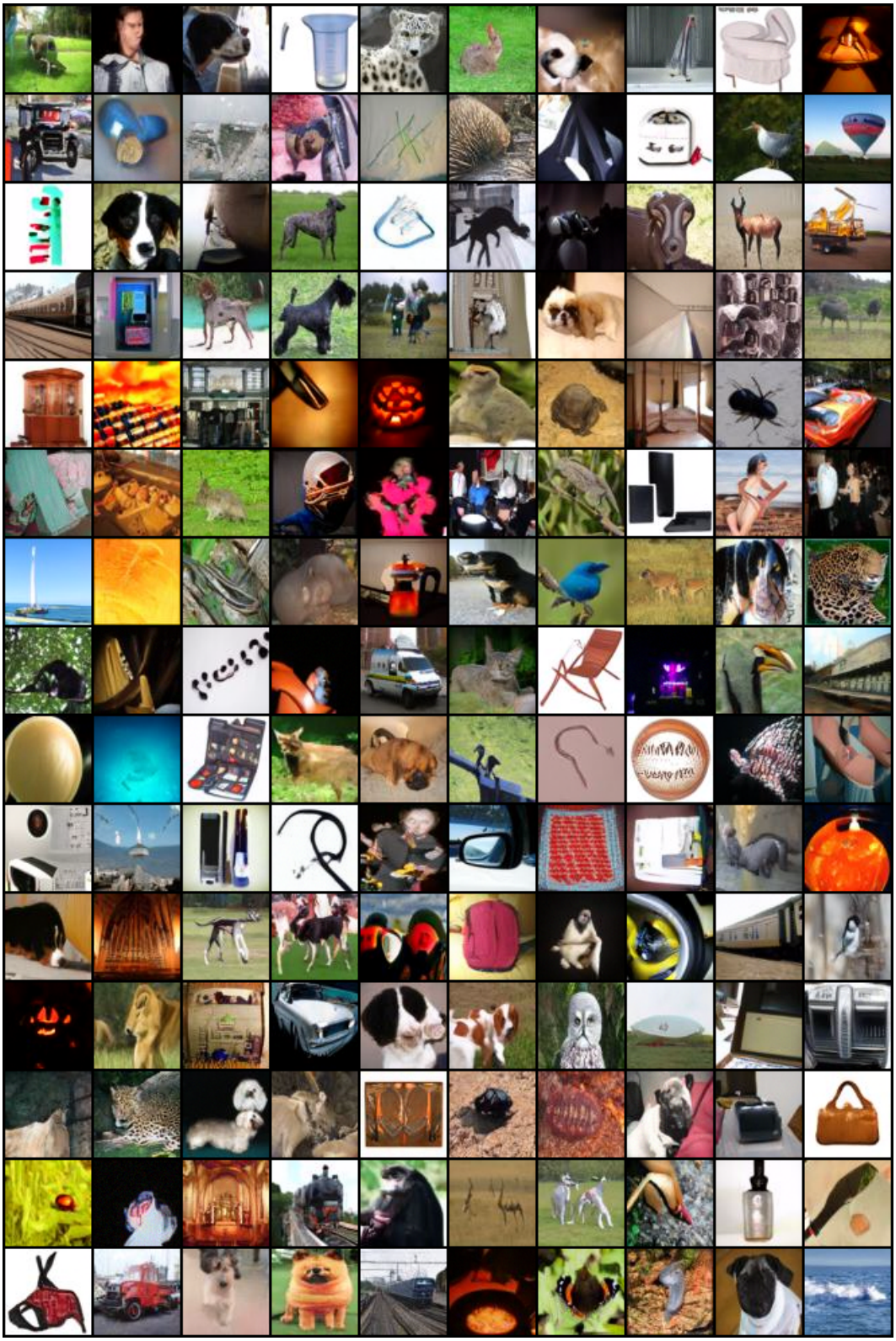

Figure 9: Non-curated unconditional ImageNet-64 generated images of a CNF trained with TGFlow-OT.

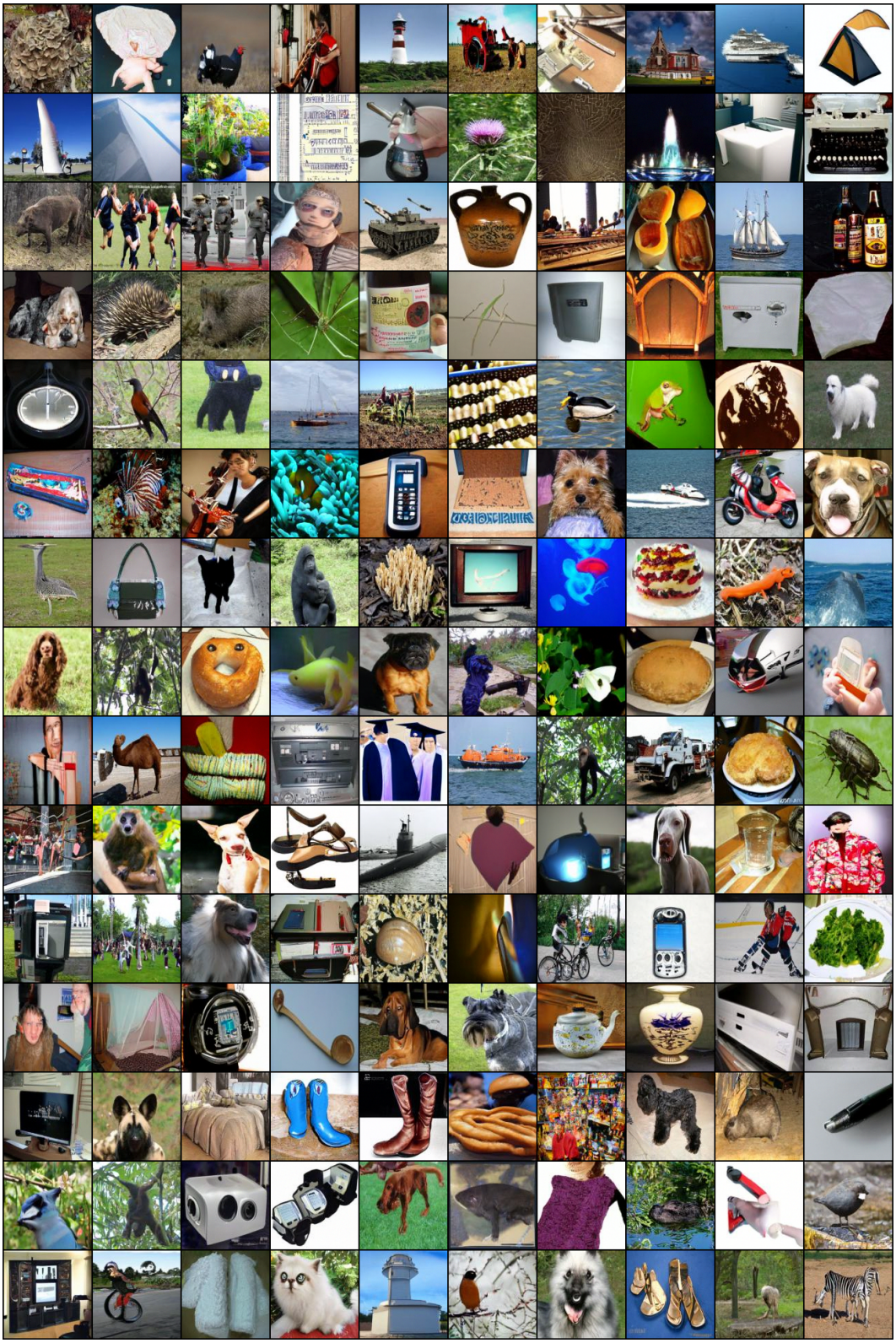

Figure 10: Non-curated unconditional ImageNet-128 generated images of a CNF trained with TGFlow-OT.

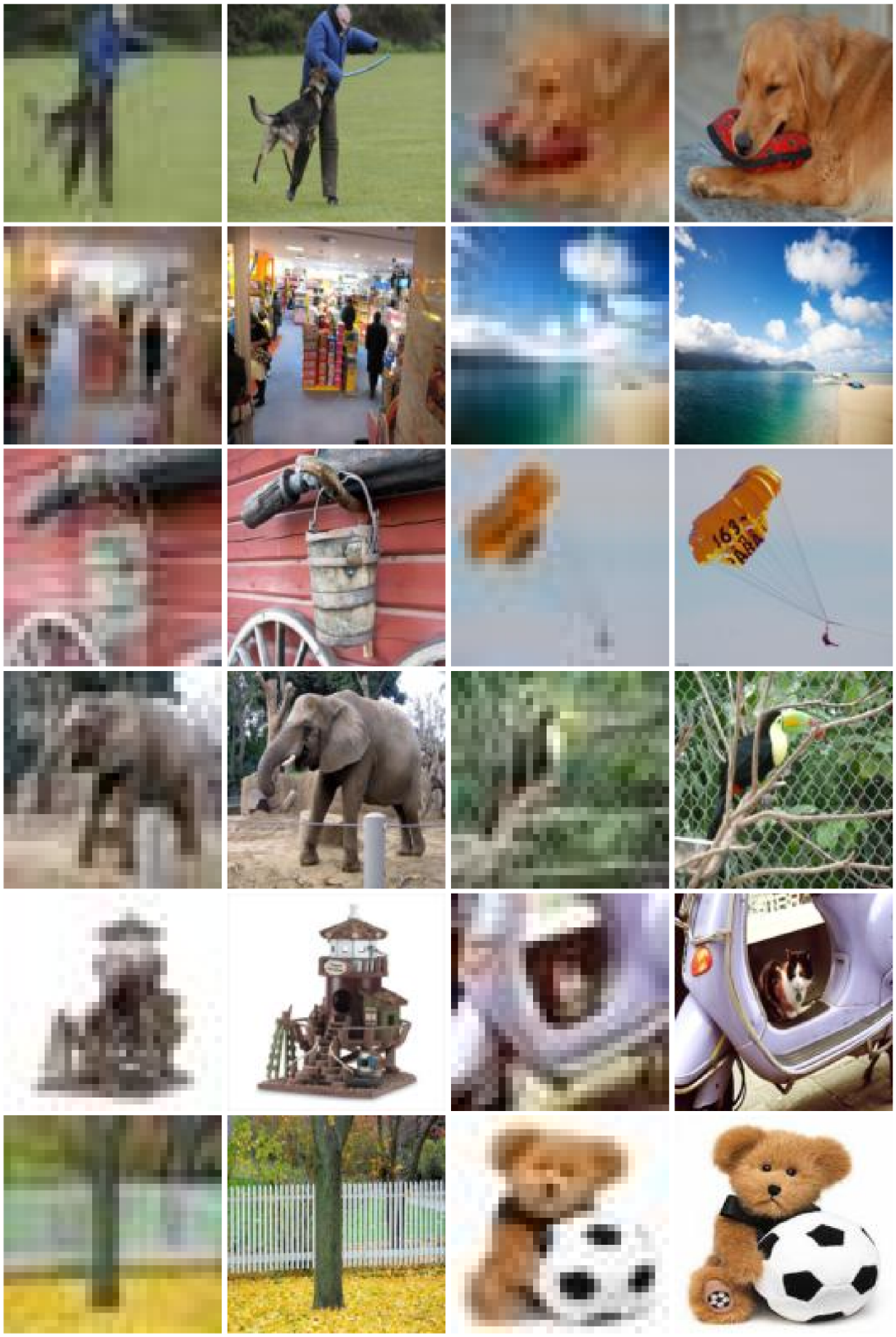

Figure 11: Conditional generation task: $32 \times 32 \rightarrow 128 \times 128$. This figure shows the high-resolution upsampled images obtained as outputs when TGFlow-OT takes low-resolution images from the validation set as inputs.