Temporal Pair Consistency for Flow Matching

Chika Maduabuchi
William & Mary

Jindong Wang
William & Mary
jdw@wm.edu

Abstract

Flow-matching-style generative models learn time-dependent vector fields, but standard objectives supervise each timestep independently, leaving shared-path temporal structure unused. We introduce *Temporal Pair Consistency* (TPC), a simple training-time objective that couples velocity predictions at paired timesteps along the same probability path. TPC leaves the architecture, probability path, sampler, and inference-time computation unchanged. Under matched backbones, training budgets, solvers, and numbers of function evaluations, TPC improves sample quality across flow matching, OT-CFM, rectified flow, DiT-style backbones, and MeanFlow. On CIFAR-10, TPC improves FM from 6.35 to 3.19 FID and OT-CFM from 3.58 to 2.90 FID at the same NFE. Mechanism diagnostics show lower measured gradient variance, higher paired-gradient correlation, and reduced temporal roughness, while random and local pairing controls are substantially weaker. Higher-resolution ImageNet experiments further show consistent gains for DiT-XL/2 at 256 and 512 resolution and for MeanFlow at 256 resolution, all at matched inference cost.

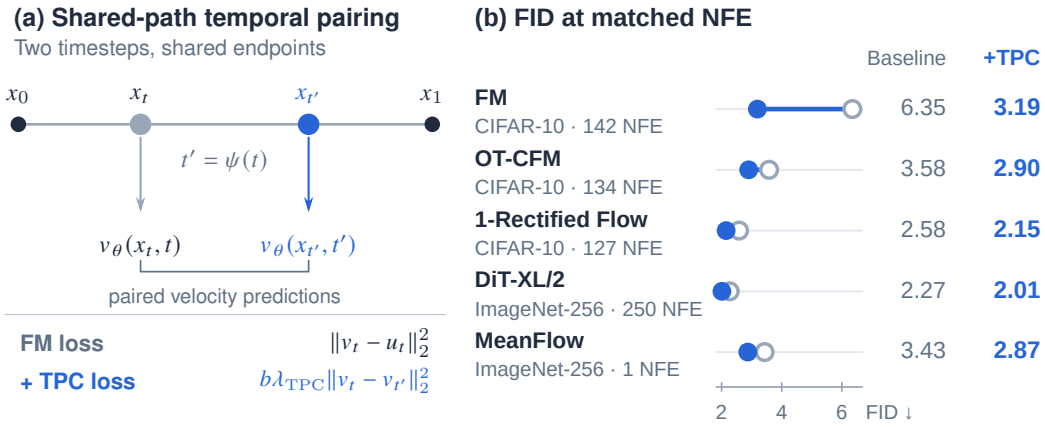


Figure 1: **Temporal Pair Consistency (TPC) at a glance.** (a) TPC couples velocity predictions at two timesteps on the same conditional probability path, adding a gated consistency penalty to the standard flow-matching objective. (b) TPC improves FID across five representative settings at matched inference NFE within each comparison. The architecture and sampler remain unchanged; CIFAR-10 TPC training requires approximately 1.68–1.70× baseline wall time.

1 Introduction

Continuous-time generative models have become a central framework for high-fidelity image synthesis, spanning diffusion models (DMs), flow matching (FM), rectified flow (RF), and related

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

deterministic or stochastic probability-flow formulations. Diffusion models have achieved strong performance across a wide range of benchmarks, benefiting from stable training objectives and well-understood stochastic dynamics Ho et al. [2020], Rombach et al. [2022], Yang et al. [2023], but often require long sampling trajectories and substantial inference-time computation. In parallel, deterministic formulations based on ordinary differential equations, including flow matching [Lipman et al., 2023] and rectified flow [Liu et al., 2023], have shown that carefully designed probability paths can provide flexible training objectives and competitive sample quality Li et al. [2023].

Despite this progress, standard flow-matching-style objectives usually supervise each timestep independently. For a sampled endpoint pair and timestep, the model is trained to match the corresponding velocity target, but predictions at other timesteps along the same probability path are not explicitly coupled. This leaves a simple source of temporal structure unused: different timesteps on the same path share the same endpoint randomness and often exhibit correlated training signals. Empirically, we observe that independent timestep training can lead to noisy training dynamics and temporally uneven velocity fields, especially in regimes where sample quality is sensitive to the learned trajectory. The goal of this paper is to test whether directly coupling such shared-path evaluations during training improves the learned vector field under controlled, matched inference settings.

We introduce *Temporal Pair Consistency* (TPC), a lightweight training-time objective for flow matching. Given two timesteps sampled along the same probability path, TPC adds a quadratic consistency term between the corresponding velocity predictions. The method is deliberately simple: it does not change the model architecture, probability path, training target, sampler, or inference-time computation. Instead, it augments the training objective with a shared-path temporal pairing term, encouraging the learned vector field to produce compatible predictions at coupled points on the same path. This makes TPC easy to add to existing FM and RF implementations, while preserving the same inference procedure used by the original model.

We study TPC as an empirical, matched-cost training modification. Across all main comparisons, the TPC-augmented model uses the same architecture, probability path, sampler, training budget, and number of function evaluations as its baseline. This design isolates whether shared-path temporal pairing improves the learned vector field without changing inference. We further use pairing controls, coupling-strength ablations, and training-dynamics diagnostics to test whether the gains are tied to structured temporal coupling rather than arbitrary regularization.

Contributions. Our contributions are threefold:

(i) We introduce *Temporal Pair Consistency* (TPC), a simple shared-path temporal pairing objective for flow matching. TPC couples velocity predictions at paired timesteps along the same probability path while leaving the architecture, probability path, sampler, and inference-time NFE unchanged.

(ii) We provide a controlled empirical study of TPC's training behavior. Under matched settings, TPC lowers measured gradient variance, increases paired-gradient correlation, and reduces temporal roughness; random and local pairing controls are weaker, and coupling-strength ablations reveal the expected over-regularization regime.

(iii) We show that TPC improves sample quality at matched inference cost across multiple FM-style settings, including FM, OT-CFM, RF, DiT-style diffusion backbones, and MeanFlow. The gains persist from CIFAR-10 and ImageNet benchmarks to higher-resolution 256×256 and 512×512 evaluations.

2 Related Work

Temporal and trajectory regularization in continuous-time models. TPC is related to prior work on temporal smoothness, consistency regularization, and trajectory regularization in continuous-time generative models. Several approaches regularize learned ODE dynamics by penalizing path length, kinetic energy, Jacobians, or other properties of the velocity field, often with the goal of improving numerical stability or invertibility in continuous normalizing flows Chen et al. [2018], Grathwohl et al. [2019], Kidger [2022], Yang et al. [2025]. Related ideas also appear in diffusion models, where smoothness of the score or drift over time is encouraged through architectural choices, noise schedules, or discretization design Song et al. [2021], Karras et al. [2022]. Recent work on flow-based generation further studies path design and straight-trajectory learning through rectified, latent-augmented, or variational formulations of flow matching Liu et al. [2023], Ma et al. [2025].

TPC differs from these approaches in where the modification is applied. Rather than changing the probability path, sampler, architecture, or ODE solver, TPC adds a training-time paired consistency term between velocity predictions evaluated at two timesteps along the same conditional probability path. The objective is therefore not a generic Jacobian or path-length penalty on the vector field, but a shared-path coupling of existing flow-matching evaluations. This makes TPC easy to add to standard flow-matching-style training while keeping the inference procedure unchanged. In our experiments, this distinction is tested directly through random and local pairing controls, which isolate whether improvements come from arbitrary temporal smoothing or from the structured shared-path pairing used by TPC.

Consistency regularization and consistency-based generative models. TPC is also connected to consistency regularization methods in supervised and self-supervised learning, where predictions are encouraged to agree under input perturbations, augmentations, or stochastic transformations Laine and Aila [2017], Sohn et al. [2020]. In generative modeling, consistency models and consistency-style flow methods enforce agreement across noise levels or timesteps to enable few-step or one-step generation Lu and Song [2025], Issenhuth et al. [2024]. In particular, Consistency Flow Matching (Consistency-FM) enforces velocity consistency along model-induced flow trajectories to facilitate straight or few-step generation Yang et al. [2025].

TPC has a different goal: it couples velocity predictions at two timesteps on the same prescribed conditional probability path, reusing the same endpoint randomness. It retains the original flow-matching regression target and does not modify the inference sampler or NFE. The objectives are complementary, as supported by our matched two-NFE composition experiment with Consistency-FM. Few-step consistency methods remain useful reference points, though direct matched-NFE comparisons are uncommon; our composition experiment provides one such comparison. Our main comparisons therefore emphasize matched architecture, path, solver, training budget, and NFE, while the matched two-NFE Consistency-FM composition directly tests whether the objectives complement one another.

Relation to flow matching, rectified flow, and modern flow pipelines. Flow matching Lipman et al. [2023], rectified flow Liu et al. [2023], stochastic-interpolant methods Albergo and Vanden-Eijnden [2023], and recent temporally structured flow constructions Ifriqi et al. [2025] improve continuous-time generative modeling primarily through the design of probability paths, interpolation schemes, or trajectory geometry. These choices determine the target velocity field and the dynamics that are integrated at sampling time.

TPC is complementary to this line of work. It does not propose a new probability path or a new solver; instead, it modifies the training objective by coupling two velocity predictions drawn from the same existing path sampler. This allows the same idea to be applied on top of different FM-style formulations, including standard flow matching, OT-CFM, rectified flow, DiT-style diffusion backbones trained with flow objectives, and average-velocity matching frameworks such as MeanFlow. This positions TPC as a training-objective modification whose role is evaluated under matched inference settings in Sections 3–4.

3 Method

Flow matching learns a time-indexed vector field that transports a simple reference distribution into the target data distribution by integrating an ordinary differential equation. We introduce *Temporal Pair Consistency* (TPC), a training-time objective that couples velocity predictions at paired timesteps along the same probability path. TPC does not change the probability path, model architecture, sampler, or inference-time computation. It simply adds a shared-path consistency term during training. We first review conditional flow matching in Section 3.1, then define shared-path temporal pairing in Section 3.2, describe fixed and learned pairing rules in Section 3.3, and give the full training objective in Sections 3.4–3.5. Figure 2 summarizes the difference between standard flow matching, TPC training, and unchanged inference.

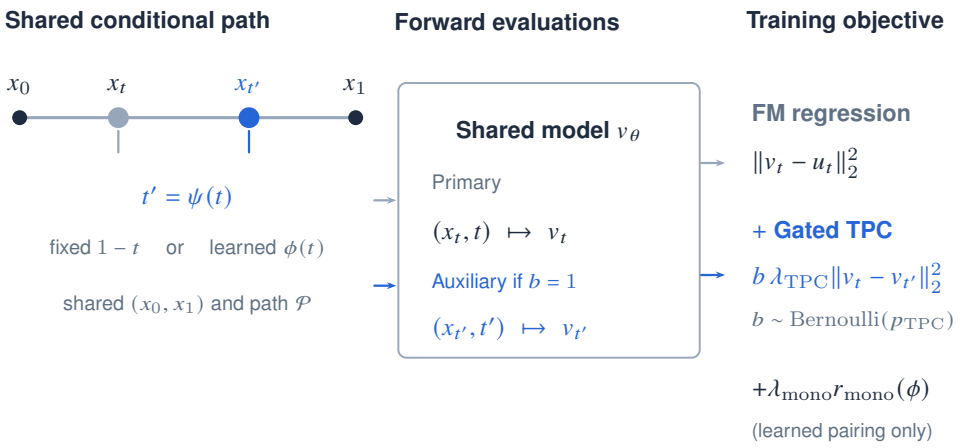


Figure 2: **Temporal Pair Consistency (TPC) training pipeline.** Two timesteps t and $t' = \psi(t)$ are evaluated along the same conditional probability path using shared endpoints and network parameters. The primary prediction retains standard flow-matching supervision, while a Bernoulli-gated auxiliary evaluation contributes the TPC consistency loss. Pairing can be fixed or learned, with optional monotonicity regularization for learned pairing only. Inference uses the original architecture, probability path, sampler, solver, and NFE.

3.1 Conditional Flow Matching

Let $x_0 \sim p_0$ denote a base distribution, such as $\mathcal{N}(0, I)$, and let $x_1 \sim p_1$ denote the data distribution. A conditional probability path is induced by a map $x_t = \Phi_t(x_0, x_1), \quad t \in [0, 1]$, with associated conditional velocity $u_t(x_t) = \partial_t \Phi_t(x_0, x_1)$. Flow matching parameterizes a velocity field $v_\theta(x, t)$ and minimizes

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\rho,\,(x_0,x_1)} \left\| v_\theta(x_t, t) - u_t(x_t) \right\|_2^2, \tag{1}$$

where ρ is typically uniform on $[0, 1]$. Eq. (1) supervises the velocity field at the sampled timestep t. In standard training, predictions at other timesteps along the same endpoint pair are not explicitly coupled.

3.2 Temporal Pairing for Shared-Path Consistency

TPC uses the endpoint pair already sampled by flow matching to construct a second point on the same probability path. For $t \sim \rho$, let $t' = \psi(t)$ be an auxiliary timestep. Using the same endpoints (x_0, x_1), the path sampler produces $(x_t, u_t) = \mathcal{P}(x_0, x_1, t)$ and $(x_{t'}, u_{t'}) = \mathcal{P}(x_0, x_1, t')$. The corresponding model predictions are $v_t = v_\theta(x_t, t)$ and $v_{t'} = v_\theta(x_{t'}, t')$. The TPC penalty is the squared discrepancy between these two velocity predictions,

$$\ell_{\text{TPC}}(t, t') = \left\| v_t - v_{t'} \right\|_2^2. \tag{2}$$

This term encourages the learned field to produce compatible velocity predictions at two coupled states generated from the same endpoint randomness. Importantly, the target velocity in Eq. (1), the probability path, and the sampling solver are unchanged. The empirical sections test whether this shared-path coupling improves sample quality and training diagnostics under matched inference settings.

3.3 Pairing Mechanisms

TPC is defined by a pairing operator $\psi : [0, 1] \to [0, 1]$ that maps a primary timestep t to an auxiliary timestep $t' = \psi(t)$. We study two practical choices.

Fixed symmetric pairing. The core, parameter-free TPC method uses the fixed map

$$\psi_{\text{fix}}(t) = 1 - t. \tag{3}$$

This rule couples early and late portions of the same probability path and cannot collapse to self-pairing.

Learned monotone pairing. As an optional adaptive extension, we consider a low-capacity learned map $\phi : [0,1] \to [0,1]$, defining $\psi_{\text{learn}}(t) = \phi(t)$. To preserve the temporal ordering structure of the path, we parameterize ϕ as a monotone function,

$$\phi(t) = \sigma\left(\sum_{i=1}^{H} a_i\, \sigma(t + b_i) + c\right), \qquad a_i = \text{softplus}(\tilde{a}_i) > 0, \tag{4}$$

where σ is the sigmoid, $b_i \in \mathbb{R}$ controls the transition location of each hidden unit, and $c \in \mathbb{R}$ is a global offset. The positive coefficients make each hidden unit contribute non-negatively to the slope before the outer sigmoid, yielding a compact monotone parameterization.

For learned pairing only, monotonicity is encouraged using a smooth grid-based penalty rather than a hard indicator. For grid points $\{g_k = k/K\}_{k=1}^{K}$, with $K = 32$ in our experiments, we use

$$r_{\text{mono}}(\phi) = \sum_{k=1}^{K-1} \text{softplus}(\phi(g_k) - \phi(g_{k+1}))\,. \tag{5}$$

For fixed pairing, we set $r_{\text{mono}} = 0$; learned monotone pairing is an optional adaptive extension and gave the strongest result.

3.4 Temporal Consistency Objective

For a sampled path and timestep pair, draw $b \sim \text{Bernoulli}(p_{\text{tpc}})$ and define the stochastically gated TPC penalty

$$\tilde{\ell}_{\text{TPC}}(t, t') = b\, \ell_{\text{TPC}}(t, t'). \tag{6}$$

The single-sample loss is then

$$\ell_{\text{sample}}(\theta; \psi) = \left\| v_\theta(x_t, t) - u_t \right\|_2^2 + \lambda_{\text{tpc}} \tilde{\ell}_{\text{TPC}}(t, t') + \lambda_{\text{mono}} r_{\text{mono}}, \tag{7}$$

where $\lambda_{\text{tpc}} \geq 0$ controls the coupling strength and $\lambda_{\text{mono}} \geq 0$ is used only for learned pairing, with $r_{\text{mono}} = r_{\text{mono}}(\phi)$ as defined in Eq. (5); for fixed pairing, $r_{\text{mono}} = 0$. The gate probability p_{tpc} controls how often coupling is applied. Eq. (7) is a training loss, not a modified sampler or probability path; inference uses the baseline solver and NFE.

3.5 Full Training Objective

Combining flow matching, stochastic temporal pairing, and the optional monotonicity penalty gives the objective

$$\mathcal{L}(\theta; \psi) = \mathbb{E}_{t \sim \rho,\, (x_0, x_1),\, b \sim \text{Bernoulli}(p_{\text{tpc}})} \begin{bmatrix} \left\| v_\theta(x_t, t) - u_t \right\|_2^2 \\ + \lambda_{\text{tpc}} \tilde{\ell}_{\text{TPC}}(t, t') \\ + \lambda_{\text{mono}} r_{\text{mono}} \end{bmatrix}, \tag{8}$$

where $t' = \psi(t)$ and both path points use the same endpoint pair (x_0, x_1) and path sampler $\mathcal{P}$. Thus the expected objective is exactly the expectation of the single-sample loss in Eq. (7); it adds no inference-time computation. Algorithmic details are provided in Appendix 6.4.

4 Experiments

We organize the empirical evaluation around three questions: whether TPC improves sample quality under matched inference cost, whether the gains are associated with measurable changes in training dynamics, and whether the same training-time modification transfers beyond the original low-resolution FM setting. We first describe the matched setup in Section 4.1, then report unconditional FM and RF results in Section 4.2, mechanism diagnostics and pairing controls in Section 4.3, noise-augmented ImageNet evaluations and higher-resolution DiT/MeanFlow extensions in Section 4.4, and ablations over pairing and coupling strength in Section 4.5. Figure 3 summarizes the matched-cost FID improvements across the main experimental settings.

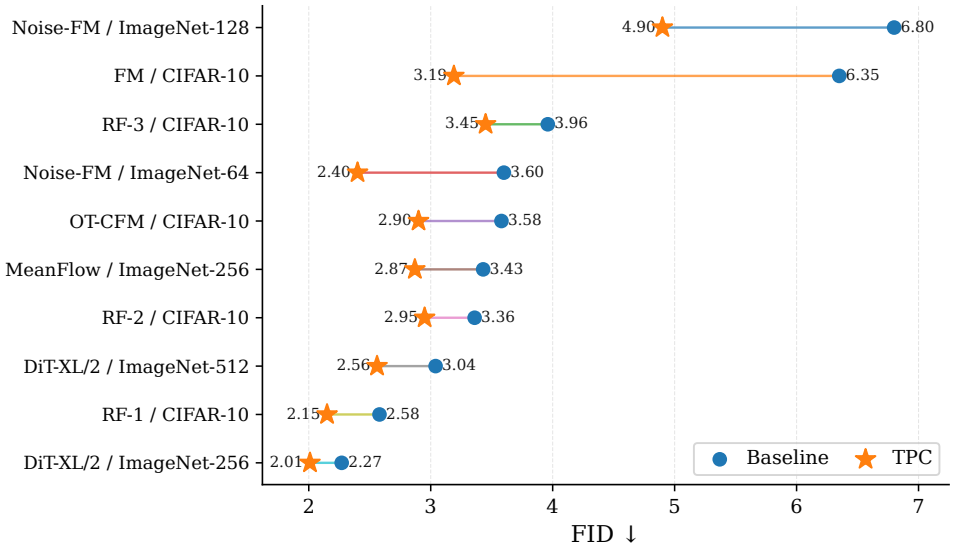


Figure 3: **TPC improves sample quality at matched inference cost.** Each row compares a baseline model with its TPC-augmented counterpart using the same architecture, probability path, sampler, and number of function evaluations. Endpoint labels show FID values; exact NFEs are reported in the corresponding tables. TPC consistently shifts FID lower across standard FM, OT-CFM, rectified flow, noise-augmented ImageNet flow matching, DiT backbones, and MeanFlow without changing inference-time computation.

4.1 Setup

We evaluate Temporal Pair Consistency (TPC) on standard image-generation benchmarks following established protocols in diffusion and continuous-time generative modeling Ho et al. [2020], Song et al. [2021], Lipman et al. [2023], Liu et al. [2023]. The main experiments use CIFAR-10 Krizhevsky and Hinton [2009] and ImageNet Deng et al. [2009] at 32×32, 64×64, and 128×128, with additional higher-resolution extensions at 256×256 and 512×512. Unless otherwise stated, baseline and TPC-augmented models use the same backbone, probability path, solver, training budget, and sampling budget. This matched setting is intentional: our goal is to isolate the effect of the training objective rather than confound it with architecture changes, altered probability paths, or increased inference-time computation.

Matched steps do not imply matched training compute: CIFAR-10 TPC uses 1.68–$1.70\times$ relative wall time and 1.04–$1.05\times$ peak memory, but leaves inference architecture, solver, and NFE unchanged. Equal- and near-equal-wall-time controls show learned TPC reaches FID 3.19, versus 5.48 for longer-trained FM and 5.72 for second-timestep supervision without consistency (Appendix Table 5); this distinguishes step, wall-clock, and inference-cost matching.

We report negative log-likelihood (NLL) Lipman et al. [2023] where applicable, Fréchet Inception Distance (FID) Heusel et al. [2017] for sample quality, Inception Score (IS) Salimans et al. [2016] and recall where used by the baseline protocol, and the number of function evaluations (NFE) Grathwohl et al. [2019] to document the inference cost. NFE is held fixed when comparing TPC to each matched baseline; the central empirical claim is that TPC improves sample quality at matched inference cost. For flow-matching models, samples are generated by solving the learned generative ODE from noise to data using the same solver settings as the corresponding baseline: Flow Matching models use adaptive ODE solvers with absolute and relative tolerances set to 10^{-5} Lipman et al. [2023], while Rectified Flow models use adaptive Runge–Kutta (RK45) solvers for full simulation and fixed-step Euler solvers for one-step and distillation settings Liu et al. [2023]. All FID, IS, and recall metrics are computed using standard evaluation protocols over 50k samples.

To diagnose the empirical mechanism, we measure gradient variance, paired-gradient correlation, and temporal roughness under the same CIFAR-10 backbone, training budget, and solver settings. These diagnostics are used only as empirical measurements of how shared-path temporal pairing

Table 1: **Unconditional diffusion and flow-matching results on CIFAR-10, ImageNet** 32×32, ImageNet 64×64, and ImageNet 128×128. Blank entries indicate methods that do not report the corresponding unconditional setting. The ImageNet-128×128 panel compares FM w/ OT and TPC-FM. Table 3 reports the separate noise-augmented conditional ImageNet evaluations, with ImageNet-128×128 in panel (a) and ImageNet-64×64 in panel (b).

	CIFAR-10			ImageNet 32×32			ImageNet 64×64		
Model	NLL↓	FID↓	NFE↓	NLL↓	FID↓	NFE↓	NLL↓	FID↓	NFE↓
DDPM Ho et al. [2020]	3.12	7.48	274	3.54	6.99	262	3.32	17.36	264
Score Matching Lipman et al. [2023]	3.16	19.94	242	3.56	5.68	178	3.40	19.74	441
i-DODE Zheng et al. [2023]	2.56	11.20	162	3.69	10.31	138	–	–	–
ScoreFlow Lipman et al. [2023]	3.09	20.78	428	3.55	14.14	195	3.36	24.95	601
FM w/ Diffusion Lipman et al. [2023]	3.10	8.06	183	3.54	6.37	193	3.33	16.88	187
FM w/ OT Lipman et al. [2023]	2.99	6.35	142	3.53	5.02	122	3.31	14.45	138
I-CFM Tong et al. [2024]	–	3.66	146	–	–	–	–	–	–
OT-CFM Tong et al. [2024]	–	3.58	134	–	–	–	–	–	–
Discrete FM Gat et al. [2024]	–	3.63	1024	–	–	–	–	–	–
V-RFM Guo and Schwing [2025]	–	3.58	1000	–	–	–	–	–	–
TPC-FM (Ours)	**2.99**	**3.19**	**142**	**3.53**	**4.22**	**122**	**3.31**	**13.14**	**138**
TPC + OT-CFM (Ours)	–	**2.90**	**134**	–	–	–	–	–	–

Model (IN 128 × 128)	**NLL↓**	**FID↓**
FM w/ OT Lipman et al. [2023]	2.90	20.9
TPC-FM (Ours)	**2.90**	**18.6**

Table 2: **Matched rectified-flow results and CIFAR-10 mechanism diagnostics**. Panel (a) reports full-simulation RF comparisons under matched adaptive RK45 settings. Panel (b) reports pairing controls and diagnostic measurements under matched CIFAR-10 settings. Panel (c) reports coupling-strength diagnostics for learned TPC-FM; Eff. wt. denotes $p_{\mathrm{tpc}}\lambda_{\mathrm{tpc}}$.

(a) Rectified flow

Method	**NFE↓**	**IS↑**	**FID↓**	**Rec.↑**
1-Rectified Flow	127	9.60	2.58	0.57
2-Rectified Flow	110	9.24	3.36	0.54
3-Rectified Flow	104	9.01	3.96	0.53
TPC-1RF	127	**9.78**	**2.15**	**0.60**
TPC-2RF	110	9.42	2.95	0.58
TPC-3RF	104	9.18	3.45	0.55

(b) Pairing controls and diagnostics

Method	**FID↓**	**NFE↓**	**Grad var/FM↓**	**Grad corr↑**	**Rough./FM↓**
FM w/ OT	6.35	142	1.00 ± 0.03	0.18 ± 0.02	1.00 ± 0.04
FM + random pairing	5.95	142	0.92 ± 0.03	0.25 ± 0.03	0.91 ± 0.03
FM + local pairing	5.20	142	0.82 ± 0.03	0.39 ± 0.03	0.78 ± 0.03
OT-CFM	3.58	134	0.84 ± 0.03	0.31 ± 0.03	0.86 ± 0.03
TPC-FM	3.19	142	0.66 ± 0.02	0.61 ± 0.03	0.58 ± 0.03
TPC + OT-CFM	**2.90**	134	$\mathbf{0.58 \pm 0.02}$	$\mathbf{0.67 \pm 0.03}$	$\mathbf{0.51 \pm 0.03}$

(c) Coupling strength

p_{tpc}	λ_{tpc}	Eff. wt.	**FID↓**	**Grad var/FM↓**	**Grad corr↑**	**Rough./FM↓**
0.00	0.00	0.000	6.350	1.00	0.18	1.00
0.25	0.10	0.025	3.922	0.87	0.30	0.85
0.50	0.25	0.125	3.885	0.74	0.47	0.70
0.75	0.10	0.075	**3.193**	0.66	0.61	0.58
0.25	0.50	0.125	4.767	0.69	0.56	0.61
1.00	1.00	1.000	6.223	**0.63**	**0.64**	**0.55**

changes training dynamics. We also include random and local pairing controls, coupling-strength ablations, a closest consistency-style reference point, high-resolution DiT-style extensions Peebles and Xie [2023], MeanFlow extensions Geng et al. [2026], and qualitative comparisons under matched sampling settings. Family-specific implementation details for applying TPC to FM, OT-CFM, rectified flow, DiT-style velocity models, noise-augmented FM, and MeanFlow are provided in Appendix 6.2.

4.2 Unconditional Image Generation

Flow Matching Table 1 reports unconditional image generation results on CIFAR-10 and ImageNet at resolutions 32×32, 64×64, and 128×128. Across datasets and resolutions, TPC-FM improves the FM w/ OT baseline under matched sampling budgets. On CIFAR-10, TPC-FM reduces FID from 6.35 to 3.19 at the same NFE of 142, while TPC + OT-CFM improves a stronger OT-CFM baseline from 3.58 to 2.90 at the same NFE of 134. Similar matched-NFE gains are observed on ImageNet 32×32 and 64×64, and TPC-FM improves ImageNet 128×128 FID from 20.9 to 18.6 while matching likelihood performance. These results support the intended empirical claim: TPC is a training-time modification that improves sample quality without changing the inference architecture, probability path, solver, or NFE.

Rectified Flow Table 2(a) reports rectified-flow results under full-simulation adaptive RK45 sampling. Across all rectification depths, adding TPC during training improves FID and recall at identical NFE. TPC-1RF improves FID from 2.58 to 2.15 at the same NFE of 127, TPC-2RF improves from

3.36 to 2.95 at NFE 110, and TPC-3RF improves from 3.96 to 3.45 at NFE 104. These results show that TPC improves the learned vector field under matched inference settings, without changing the solver or using additional sampling steps. The complete one-step and full-simulation RF results are provided in Appendix Table 9.

4.3 Mechanism Diagnostics and Pairing Controls

We next analyze whether the observed improvements are tied to structured temporal pairing rather than arbitrary regularization. All diagnostics in this subsection use the same CIFAR-10 backbone, training budget, and solver settings. Gradient variance and temporal roughness are normalized by the FM w/ OT baseline, while paired-gradient correlation measures how strongly the paired evaluations align during training.

Table 2(b) shows a consistent ordering across both sample quality and diagnostics. Random pairing, which uses an independent $t' \sim U[0,1]$, gives only a modest improvement over FM, while local pairing is stronger but still below structured TPC. TPC-FM reduces measured gradient variance to $0.66\times$ FM, increases paired-gradient correlation from 0.18 to 0.61, and reduces temporal roughness to $0.58\times$ FM while improving FID from 6.35 to 3.19 at the same NFE. Adding TPC on top of OT-CFM further improves FID from 3.58 to 2.90 at the same NFE of 134, indicating that TPC is complementary to a stronger FM-family baseline. Appendix Table 10 shows the same training-wide trend: relative gradient variance drops from $0.84\times$ to $0.66\times$, while paired-gradient correlation rises from 0.34 to 0.61.

For the linear OT conditional path, paired points share the same conditional displacement target, providing a basis for correlated training signals. Table 2(c) shows that moving from effective coupling 0.000 to 0.075 lowers gradient variance from 1.00 to 0.66 and raises paired-gradient correlation from 0.18 to 0.61, alongside improved FID from 6.350 to 3.193. Increasing coupling further to 1.000 changes variance only from 0.66 to 0.63 and correlation from 0.61 to 0.64, while FID worsens to 6.223. This empirical pattern is consistent with a bias–variance trade-off: structured coupling can reduce variance and align training signals, but because the optimal marginal velocity field varies with state and time, excessive coupling can suppress genuine temporal variation and introduce fitting bias. These observations support this interpretation but do not establish a formal causal proof.

Appendix Table 7 reports a direct same-codebase composition experiment at matched 2-NFE inference cost. Fixed and learned TPC improve reproduced Consistency-FM FID from 5.34 to 4.70 and 4.29, respectively. Learned TPC yields a 1.05 FID improvement over the reproduction and beats the equal-wall-time longer-training control (5.08 FID at $1.70\times$ wall time) by 0.79 FID.

4.4 Extending TPC to Modern Flow Pipelines

The preceding sections evaluate standard probability–flow sampling. To test whether the same training-time objective remains useful in more modern ImageNet-style pipelines, we also study noise-augmented flow matching with score-based denoising at sampling time, a design used by several high-resolution diffusion and flow-based models Dhariwal and Nichol [2021], Song et al. [2021], Zhai et al. [2025].

We evaluate TPC as an *orthogonal training-time addition* under this protocol: the baseline and TPC model use the same noise augmentation, denoising rule, architecture, and sampling settings. As in prior work, denoising is performed using the model-implied score of the noise-augmented probability flow, so the matched comparison isolates the effect of adding TPC to the flow-matching training objective.

We evaluate this setting on conditional ImageNet generation at 64×64 and 128×128. The most relevant comparison is between the matched FM baseline and TPC-FM under the same noise-augmented and denoised protocol. We also show diffusion Dhariwal and Nichol [2021], Nichol and Dhariwal [2021] and consistency-model Song et al. [2023], Song and Dhariwal [2024] results from prior work for context; these use their reported settings and are not matched comparisons or primary evidence for TPC's contribution. The combined results are reported in Table 3, with ImageNet-128×128 in panel (a) and ImageNet-64×64 in panel (b).

Table 3: **Conditional ImageNet generation performance under noise-augmented training and score-based denoising.** External diffusion and consistency-model entries are shown for context using their reported settings; the matched comparison is FM + noise + denoising versus TPC-FM + noise + denoising.

(a) ImageNet 128×128

Model	Class	FID ↓
ADM-G Dhariwal and Nichol [2021]	Diff/FM	2.97
CDM Ho et al. [2022]	Diff/FM	3.52
Simple Diffusion Hoogeboom et al. [2023]	Diff/FM	1.94
RIN Jabri et al. [2023]	Diff/FM	2.75
FM + noise + denoising	FM	6.8
TPC-FM + noise + denoising	**FM**	**4.9**

(b) ImageNet 64×64

Model	Class	FID ↓
EDM Karras et al. [2022]	Diff/FM	1.55
iDDPM Nichol and Dhariwal [2021]	Diff/FM	2.92
ADM (dropout) Dhariwal and Nichol [2021]	Diff/FM	2.09
CD (LPIPS) Song et al. [2023]	CM	4.70
iCT-deep Song and Dhariwal [2024]	CM	3.25
FM + noise + denoising	FM	3.6
TPC-FM + noise + denoising	**FM**	**2.4**

Noise specification. In these noise-augmented evaluations, we adopt additive Gaussian noise during training to match modern high-resolution flow-based pipelines Song et al. [2021], Zhai et al. [2025]. Specifically, states along the probability path are perturbed as $x_t \leftarrow x_t + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. The noise scale σ is selected to balance denoising strength and flow fidelity and is chosen *exclusively* to optimize FID under the baseline flow-matching setup. Unless otherwise stated, we use $\sigma = 0.05$ for ImageNet-64 and $\sigma = 0.15$ for ImageNet-128. These values are then held fixed when training TPC-FM, with no additional tuning, so the TPC comparison is not confounded by additional noise-hyperparameter tuning.

Higher-resolution and modern-backbone extensions. To test whether the same training-time modification transfers beyond the original U-Net FM setting, we evaluate TPC on DiT-XL/2 at 256×256 and 512×512, and on MeanFlow at 256×256, under matched inference settings. Table 4(c) reports consistent FID improvements: DiT-XL/2 improves from 2.27 to 2.01 at 256×256 and from 3.04 to 2.56 at 512×512, while MeanFlow improves from 3.43 to 2.87 at 1 NFE. Since each baseline and TPC-augmented variant uses the same NFE, these results support improved sample quality at matched inference cost rather than reduced sampling cost.

Qualitative comparisons. In addition to the quantitative results, the appendix collects representative unconditional samples on CIFAR-10, ImageNet-32, ImageNet-64, and ImageNet-128 (Figures 6 and 7), class-conditional ImageNet-256 generations (Figure 8), and multi-page class-conditional ImageNet-512 side-by-side DiT/TPC-DiT comparisons (Figure 9). Across these figures, the TPC-augmented models produce samples that are visually sharper, more coherent, and more class-consistent, especially in the higher-resolution conditional settings. The qualitative evidence is therefore consistent with the matched-FID improvements reported above and suggests that the gains reflect perceptible improvements in sample quality rather than only numerical changes in the evaluation metric.

4.5 Ablation Studies

Table 4 examines the temporal-pairing probability p_{tpc} and TPC coupling strength λ_{tpc} for fixed and learned pairing on CIFAR-10. Across a broad range of settings, TPC-FM improves over the FM w/ OT baseline FID of 6.35, but the gains are not monotonic in coupling strength. Moderate coupling gives the best sample quality, while overly strong coupling can over-regularize the vector field and degrade FID, consistent with the diagnostic pattern in Table 2(c). The learned pairing variant gives the best overall result at $p_{\text{tpc}} = 0.75$ and $\lambda_{\text{tpc}} = 0.10$ with FID 3.19, improving over the best fixed-pairing setting of 3.59. Appendix Table 8 further verifies that the best learned pairing does not collapse to the identity map.

Table 4: **CIFAR-10 ablations and modern-backbone extensions for TPC-FM.** Panel (a) reports the fixed-pairing ablation, and panel (b) reports the learned-pairing ablation across temporal-pairing probability p_{tpc} and coupling strength λ_{tpc}. Panel (c) reports modern-backbone extensions under matched inference settings.

(a) Fixed pairing

p_{tpc}	λ_{tpc}				Best
	0.1	0.25	0.5	1.0	
0.25	3.895	3.833	4.104	4.314	3.833
0.50	5.101	4.794	4.756	4.216	4.216
0.75	4.089	4.101	4.075	3.931	3.931
1.00	4.309	**3.594**	4.179	6.545	**3.594**

(b) Learned pairing

p_{tpc}	λ_{tpc}				Best
	0.1	0.25	0.5	1.0	
0.25	3.922	4.249	4.767	4.513	3.922
0.50	3.895	3.885	4.266	4.723	3.885
0.75	**3.193**	3.620	4.016	3.898	**3.193**
1.00	4.402	4.382	4.914	6.223	4.382

(c) Modern backbones

Family	Base FID	**TPC FID**	Δ FID	NFE
DiT-XL/2 @ 256	2.27	**2.01**	0.26	250
DiT-XL/2 @ 512	3.04	**2.56**	0.48	250
MeanFlow @ 256	3.43	**2.87**	0.56	1

5 Conclusion

We introduced Temporal Pair Consistency (TPC), a simple shared-path training objective for flow-matching-style generative models. TPC couples velocity predictions at paired timesteps along the same probability path while leaving the model architecture, probability path, sampler, and inference-time computation unchanged. Across flow matching, OT-CFM, rectified flow, DiT-style backbones, and MeanFlow, TPC improves sample quality under matched inference settings. Mechanism diagnostics further show that these gains are accompanied by lower measured gradient variance, higher paired-gradient correlation, and reduced temporal roughness, while random and local pairing controls are substantially weaker. Overall, our results support TPC as a lightweight and modular training-time modification for improving continuous-time generative models at matched inference cost.

Limitations

Broader evaluation on text-to-image generation and other modalities remains future work.

Acknowledgments and Disclosure of Funding

This work was partially supported by the Commonwealth Cyber Initiative (CCI) program (H-2Q25-020), the William & Mary Faculty Research Award, the Modal Academic Compute Award, and computational resources provided by the NSF ACCESS program with support from the NCSA Delta and DeltaAI systems.

Competing Interests

The authors declare that they have no known competing interests.

References

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 133345–133385. Curran Associates, Inc., 2024. doi: 10.52202/079017-4239. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f0d629a734b56a642701bba7bc8bb3ed-Paper-Conference.pdf.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Proceedings of the 39th International Conference on Neural Information Processing Systems*, NIPS '25, Red Hook, NY, USA, 2026. Curran Associates Inc. doi: 10.52202/085713-2534. URL https://doi.org/10.52202/085713-2534.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: free-form continuous dynamics for scalable reversible generative models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

Pengsheng Guo and Alex Schwing. Variational rectified flow matching. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Rk18ZikrFI.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: end-to-end diffusion for high resolution images. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Tariq Berrada Ifriqi, John Nguyen, Karteek Alahari, Jakob Verbeek, and Ricky T. Q. Chen. Flowception: Temporally expansive flow matching for video generation, 2025. URL https://arxiv.org/abs/2512.11438.

Thibaut Issenhuth, Ludovic Dos Santos, Jean-Yves Franceschi, and Alain Rakotomamonjy. Improving consistency models with generator-induced coupling. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. URL https://openreview.net/forum?id=wt5ymd3h8h.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 26565–26577. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/a98846e9d9cc01cfb87eb694d946ce6b-Paper-Conference.pdf.

Patrick Kidger. On neural differential equations, 2022. URL https://arxiv.org/abs/2202.02435.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJ6oOfqge.

Lingxiao Li, Samuel Hurault, and Justin Solomon. Self-consistent velocity matching of probability flows. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=C6fvJ2RfsL.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LyJi5ugyJx.

Chenrui Ma, Xi Xiao, Tianyang Wang, Xiao Wang, and Yanning Shen. Learning straight flows: Variational flow matching for efficient generation, 2025. URL https://arxiv.org/abs/2511.17583.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL https://openreview.net/forum?id=-NEXDKk8gZ.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023. doi: 10.1109/ICCV51070.2023.00387.

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 (CVPR)*, pages 10684–10695, June 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 596–608. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/06964dce9addb1c5cb5d6e3d9838f733-Paper.pdf.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4), November 2023. ISSN 0360-0300. doi: 10.1145/3626235. URL https://doi.org/10.1145/3626235.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Jingwei Liu, Minkai Xu, Chenlin Meng, Stefano Ermon, Wentao Zhang, and Bin CUI. Consistency flow matching: Defining straight flows with velocity consistency, 2025. URL https://openreview.net/forum?id=bS76qaGbel.

Shuangfei Zhai, Ruixiang ZHANG, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M. Susskind. Normalizing flows are capable generative models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2uheUFcFsM.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Improved techniques for maximum likelihood estimation for diffusion odes. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Appendix

Appendix Contents

Appendix entry	Page
6. Supplementary Experimental Protocol	14
6.1. Matched-cost evaluation	14
Table 5: training-overhead and equal-compute controls	15
6.2. Implementation across model families	15
Table 6: applying TPC across model families	16
Algorithm 1: family-agnostic TPC wrapper	16
6.3. Compute hardware	16
6.4. Training algorithm	17
Algorithm 2: Flow Matching with Temporal Pair Consistency	17
6.5. TPC variants and pairing controls	17
6.6. Diagnostic measurements	17
7. Supplementary Quantitative Results	18
7.1. Consistency-FM composition experiment	18
Table 7: matched 2-NFE Consistency-FM/TPC comparison	18
7.2. Learned-pairing non-collapse diagnostic	18
Table 8: learned-pairing non-collapse diagnostic	19
7.3. Full rectified-flow results	19
Table 9: full RF one-step and full-simulation results	19
7.4. Training-phase diagnostics	19
Table 10: early/mid/late diagnostic trends	19
7.5. Training-dynamics and matched-cost visualizations	20
Figure 4: training diagnostic behavior	20
Figure 5: matched-cost teaser visualization	20
8. Supplementary Qualitative Results	21
Figure 6: CIFAR-10, ImageNet-32, and ImageNet-64 samples	21
Figure 7: ImageNet-128 samples	21
Figure 8: ImageNet-256 class-conditional samples	22
Figure 9: ImageNet-512 DiT/TPC-DiT comparisons	22–24
9. Impact Statement	24

6 Supplementary Experimental Protocol

This appendix contains supplementary protocol details, quantitative results, and qualitative samples. The main empirical claim of the paper is matched-cost improvement: TPC changes the training objective by adding a shared-path paired velocity-consistency term, while leaving the model architecture, probability path, sampler, and inference-time computation unchanged.

6.1 Matched-cost evaluation

For the primary comparisons, each TPC-augmented model is evaluated against the corresponding baseline using the same backbone, data preprocessing, probability path, solver family, and number of function evaluations (NFE). We therefore report NFE as an accounting variable for inference cost, not as a claim that TPC universally reduces the number of solver steps. This protocol is intended to isolate the effect of the training-time paired-consistency objective.

The cleanest matched comparisons are the FM-family and RF-family comparisons in the main text. For example, TPC-FM is compared to FM with the same NFE, TPC+OT-CFM is compared to OT-CFM with the same NFE, and TPC-RF is compared to the corresponding RF baseline under the same one-step or full-simulation setting. The higher-resolution DiT-style and MeanFlow extensions follow the same principle: baseline and TPC-augmented variants use matched inference settings.

Table 5: Equal-compute CIFAR-10 controls. Measurements use matched hardware, precision, batch size, and data pipelines. Relative steps, wall time, and peak memory are normalized to FM. FID values are shown as reported.

Method	**Relative steps**	**Relative wall time**	**Peak memory**	**FID↓**
FM	$1.00\times$	$1.00\times$	$1.00\times$	6.35 ± 0.11
FM, longer training	$1.70\times$	$1.70\times$	$1.00\times$	5.48 ± 0.09
FM + second-timestep supervision, no consistency	$1.00\times$	$1.68\times$	$1.04\times$	5.72 ± 0.10
FM + random pairing	$1.00\times$	$1.68\times$	$1.04\times$	5.95 ± 0.12
FM + local pairing	$1.00\times$	$1.68\times$	$1.04\times$	5.20 ± 0.09
Fixed TPC	$1.00\times$	$1.68\times$	$1.04\times$	3.59 ± 0.06
Learned TPC	$1.00\times$	$1.70\times$	$1.05\times$	3.19 ± 0.05

6.2 Implementation Details for Applying TPC Across Model Families

TPC is implemented as a training-time paired-evaluation wrapper around an existing continuous-time generative objective. It does not change the model architecture, probability path, sampler, solver tolerance, number of function evaluations, or inference-time computation. For every baseline family, the implementation consists of four steps: (i) sample the same training tuple used by the original method; (ii) evaluate the model at the original timestep t; (iii) reuse the same endpoint, noise, or path randomness to evaluate an auxiliary timestep $t' = \psi(t)$; and (iv) add a consistency penalty between the two predictions after expressing them in the same velocity or velocity-equivalent parameterization.

Given a path sampler $\mathcal{P}$ and endpoint randomness (x_0, x_1), we write

$$(x_t, u_t) = \mathcal{P}(x_0, x_1, t), \qquad (x_{t'}, u_{t'}) = \mathcal{P}(x_0, x_1, t'), \qquad t' = \psi(t).$$

The baseline supervision is unchanged, $\mathcal{L}_{\text{base}} = \|v_\theta(x_t, t) - u_t\|_2^2$. TPC adds a gated paired term,

$$\mathcal{L}_{\text{TPC}} = b\,\lambda_{\text{TPC}} \|v_\theta(x_t, t) - v_\theta(x_{t'}, t')\|_2^2, \qquad b \sim \text{Bernoulli}(p_{\text{TPC}}),$$

so that the full objective is

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{TPC}} + \lambda_{\text{mono}} r_{\text{mono}}(\psi).$$

The monotonicity term is used only for the learned pairing rule. Unless otherwise stated, we use the learned monotone pairing with $p_{\text{TPC}} = 0.75$, $\lambda_{\text{TPC}} = 0.10$, $\lambda_{\text{mono}} = 10^{-3}$, and $K = 32$ grid points for the monotonicity penalty. The fixed-pairing variant uses $\psi(t) = 1 - t$ and sets $\lambda_{\text{mono}} = 0$.

Flow matching and OT-CFM. For standard FM and OT-CFM, TPC directly reuses the original conditional path sampler. The paired point $(x_{t'}, u_{t'})$ is generated from the same endpoint pair or minibatch-OT assignment as (x_t, u_t). The primary supervision remains the usual velocity regression at t; the auxiliary evaluation at t' contributes only through the TPC consistency term. No additional target supervision is applied at t' unless it is already part of the baseline method.

Rectified flow. For rectified flow, the same rectified interpolation path and target velocity used by the baseline are retained. TPC samples the paired timestep $t' = \psi(t)$ along the same rectified path and applies the consistency term between $v_\theta(x_t, t)$ and $v_\theta(x_{t'}, t')$. The rectification stage, sampler, solver, tolerance, and NFE are unchanged.

DiT-style velocity models. For DiT backbones trained with a native velocity-prediction objective, including the DiT-XL/2 extensions reported in the paper, TPC is applied directly to the predicted velocity outputs at t and t'. Thus both evaluations use the same DiT architecture, patch size, class/text conditioning, noise schedule, solver, and NFE as the baseline; only the training loss receives the additional paired term.

For DiT implementations that predict ϵ or x_0 rather than velocity, we first convert both predictions to the velocity-equivalent target associated with the baseline interpolation before computing the TPC penalty. For a diffusion interpolation $x_t = \alpha_t x_0 + \sigma_t \epsilon$, the velocity target is $u_t = \dot{\alpha}_t x_0 + \dot{\sigma}_t \epsilon$. Therefore an x_0-prediction $\hat{x}_0$ is converted using $\hat{\epsilon} = (x_t - \alpha_t \hat{x}_0)/\sigma_t$ and $\hat{u}_t = \dot{\alpha}_t \hat{x}_0 + \dot{\sigma}_t \hat{\epsilon}$; an ϵ-prediction $\hat{\epsilon}$ is converted using $\hat{x}_0 = (x_t - \sigma_t \hat{\epsilon})/\alpha_t$ and the same formula for $\hat{u}_t$. The paired loss is always computed in this common vector-field space, not directly between heterogeneous parameterizations.

Table 6: **Applying TPC across model families.** TPC is always computed between two predictions expressed in the same velocity or velocity-equivalent space and evaluated along the same probability path/randomness. The sampler, solver, architecture, and NFE are unchanged for every family.

Family	Paired quantity	Same-path randomness	Inference change?
FM	velocity $v_\theta(x_t, t)$	same endpoint pair (x_0, x_1)	none
OT-CFM	velocity $v_\theta(x_t, t)$	same minibatch-OT pair / endpoint assignment	none
Rectified Flow	rectified-flow velocity	same rectified path endpoints	none
DiT native velocity model	native velocity prediction	same noise/data pair and timestep schedule	none
DiT ϵ/x_0 model	converted velocity-equivalent prediction	same noise/data pair and timestep schedule	none
Noise-augmented FM	velocity under fixed noise protocol	same endpoints and fixed noise augmentation	none
MeanFlow	native average-velocity prediction	same noise/data randomness	none

Noise-augmented ImageNet flow matching. For noise-augmented FM experiments, the same additive noise protocol and denoising rule are used for both baseline and TPC models. The paired point is generated with the same endpoint randomness and the same noise-augmentation protocol as the original point. The noise scale σ is selected under the baseline setting and then held fixed for TPC, so the comparison isolates the effect of temporal pairing rather than noise-hyperparameter tuning.

MeanFlow. For MeanFlow-style average-velocity matching, TPC is applied to the model's average-velocity prediction in the native MeanFlow parameterization. The paired timestep uses the same data/noise randomness and the same sampling schedule as the baseline MeanFlow run. The one-step sampler and NFE remain unchanged; TPC modifies only the training objective by requiring the paired average-velocity predictions to be consistent.

What is held fixed. Across all families, the model backbone, path sampler, endpoint pairing, data preprocessing, optimizer, training steps, batch size, sampler, solver tolerance, and NFE are held fixed between the baseline and the TPC variant. TPC adds one extra model evaluation only on minibatches where $b = 1$. At inference time, the learned vector field is sampled exactly as in the corresponding baseline.

Algorithm 1 Family-agnostic TPC wrapper

Require: Baseline path sampler $\mathcal{P}$, model prediction map v_θ, pairing rule ψ, p_{TPC}, λ_{TPC}.
Sample baseline training randomness (x_0, x_1) and timestep t.
Compute baseline point and target: $(x_t, u_t) = \mathcal{P}(x_0, x_1, t)$.
Predict primary velocity: $v_t = v_\theta(x_t, t)$.
Compute baseline loss: $\mathcal{L}_{\mathrm{base}} = \|v_t - u_t\|_2^2$.
Sample gate $b \sim \mathrm{Bernoulli}(p_{\mathrm{TPC}})$.
if $b = 1$ **then**
    Set $t' = \psi(t)$.
    Reuse the same path randomness: $(x_{t'}, u_{t'}) = \mathcal{P}(x_0, x_1, t')$.
    Predict paired velocity: $v_{t'} = v_\theta(x_{t'}, t')$.
    Compute $\mathcal{L}_{\mathrm{TPC}} = \lambda_{\mathrm{TPC}} \|v_t - v_{t'}\|_2^2$.
else
    Set $\mathcal{L}_{\mathrm{TPC}} = 0$.
end if
Add monotonicity penalty only if ψ is learned.
Update using $\mathcal{L}_{\mathrm{base}} + \mathcal{L}_{\mathrm{TPC}} + \lambda_{\mathrm{mono}} r_{\mathrm{mono}}(\psi)$.

6.3 Compute Hardware

All reported experiments were run on an internal GPU cluster using NVIDIA H200 GPUs with 141GB HBM3e memory per GPU and local high-throughput training storage. Lower-resolution CIFAR-10 and ImageNet runs used fewer GPUs, while the largest reported high-resolution run, DiT-XL/2 on ImageNet-512×512, used 32 NVIDIA H200 GPUs. Our main comparisons report NFE to account for matched inference cost; this subsection documents the compute environment used for the reported experiments. The project also included smaller pilot and tuning runs, but the paper reports the final matched experimental runs.

6.4 Training algorithm

Algorithm 2 Flow Matching with Temporal Pair Consistency (TPC)

Require: Model f_θ, path sampler $\mathcal{P}$, pairing rule ψ, TPC probability p_{tpc}, weights λ_{tpc} and λ_{mono}.

Fixed pairing: $\psi(t) = 1 - t$.
Learned pairing: $\psi(t) = \phi(t)$, with ϕ defined by Eq. (4).
for each minibatch (x_1) **do**
  Sample x_0 and $t \sim \mathcal{U}(0, 1)$.
  $(x_t, u_t) = \mathcal{P}(x_0, x_1, t)$.
  $v_t = f_\theta(x_t, t)$.
  $\mathcal{L}_{\text{FM}} = \|v_t - u_t\|_2^2$.
  Sample $b \sim \text{Bernoulli}(p_{\text{tpc}})$.
  if $b = 1$ **then**
    $t' = \psi(t)$.
    $(x_{t'}, u_{t'}) = \mathcal{P}(x_0, x_1, t')$.
    $v_{t'} = f_\theta(x_{t'}, t')$.
    $\mathcal{L}_{\text{TPC}} = \|v_t - v_{t'}\|_2^2$.
  else
    $\mathcal{L}_{\text{TPC}} = 0$.
  end if
  if using learned pairing **then**
    Construct grid $g_1, \ldots, g_K$.
    $r_{\text{mono}}(\phi) = \sum_{i=1}^{K-1} \text{softplus}(\phi(g_i) - \phi(g_{i+1}))$.
  else
    $r_{\text{mono}}(\phi) = 0$.
  end if
  $\mathcal{L} = \mathcal{L}_{\text{FM}} + \lambda_{\text{tpc}}\mathcal{L}_{\text{TPC}} + \lambda_{\text{mono}} r_{\text{mono}}(\phi)$.
  Update trainable parameters of f_θ and, when learned pairing is used, ϕ.
end for
return f_θ.

6.5 TPC variants and pairing controls

We evaluate TPC using both fixed and learned pairing rules. The fixed rule uses the parameter-free symmetric map $\psi_{\text{fix}}(t) = 1 - t$, which adds no learned parameters and cannot collapse to self-pairing. The learned variant uses a low-capacity monotone map $\phi(t)$, as described in the method section, and gives the strongest CIFAR-10 result in our ablations. We therefore report both variants: fixed pairing serves as a non-collapsible structured reference, while learned pairing tests whether adaptive temporal pairing can further improve the TPC objective.

To test whether the gains come from the specific shared-path pairing structure rather than arbitrary temporal coupling, we include two controls. Random pairing samples an independent auxiliary timestep $t' \sim U[0, 1]$. Local pairing samples an auxiliary timestep near the primary timestep t. These controls use the same training setup as TPC and are reported in the main diagnostic table. Their weaker FID and diagnostic improvements support the interpretation that the structured pairing choice matters empirically.

6.6 Diagnostic measurements

The diagnostic tables in the main text report empirical measurements of the training dynamics under matched CIFAR-10 settings. Gradient variance is reported as a ratio to the matched FM baseline. Paired-gradient correlation measures alignment between the two paired training evaluations. Temporal roughness measures the finite-difference variation of velocity predictions along paired path states and is also normalized by the FM baseline. These quantities are diagnostic measurements only: they are used to characterize how TPC changes training behavior, not as formal guarantees.

The primary empirical interpretation is improved training-time vector-field estimation, supported by gradient-variance, paired-gradient-correlation, temporal-roughness, and pairing-control mea-

surements. Excessive coupling may introduce fitting bias by forcing evaluations of genuinely different marginal fields toward agreement. These diagnostics alone do not establish a quantitative bias–variance decomposition or predict the coupling strength that minimizes FID. A local Taylor/ODE truncation argument applies to nearby temporal pairs under appropriate smoothness assumptions; it does not establish a global result for the nonlocal fixed rule $\psi(t) = 1 - t$. Accordingly, high-NFE improvements under adaptive ODE solvers with 10^{-5} tolerance are interpreted primarily as training-time gains, not as evidence of reduced numerical discretization error.

7 Supplementary Quantitative Results

This section contains additional tabular and diagnostic results that complement the main matched-cost comparisons.

7.1 Consistency-FM composition experiment

We compare Consistency-FM with fixed- and learned-TPC compositions in the same codebase on CIFAR-10. All runs use the same backbone, dataset, optimizer, evaluation protocol, and 2-NFE sampler; relative training wall time is normalized to the reproduced Consistency-FM run. Table 7 reports mean $\pm$ standard deviation over five independent training seeds. The Consistency-FM result of 5.34 ± 0.10 is our independent reproduction; its mean coincides with the published FID of 5.34.

Table 7: Same-codebase CIFAR-10 Consistency-FM composition comparison. All variants use the same backbone, dataset, optimizer, evaluation protocol, and 2-NFE sampler. Relative training wall time is normalized to the Consistency-FM reproduction. FID values are mean $\pm$ standard deviation over five independent training seeds; the reproduced baseline mean coincides with the published FID of 5.34.

Method	**NFE↓**	**Relative training wall time**	**FID↓**
Consistency-FM	2	$1.00\times$	5.34 ± 0.10
Consistency-FM, longer training	2	$1.70\times$	5.08 ± 0.09
Consistency-FM + fixed TPC	2	$1.68\times$	4.70 ± 0.08
Consistency-FM + learned TPC	**2**	$1.70\times$	4.29 ± 0.07

7.2 Learned-pairing non-collapse diagnostic

Because the learned pairing map could in principle approach the identity map $\phi(t) = t$, we include a direct diagnostic of the final learned maps. We evaluate each pairing rule on a fixed grid $\{g_k\}_{k=1}^{K}$ and report the mean separation from identity,

$$\Delta_{\mathrm{id}} = \frac{1}{K} \sum_{k=1}^{K} |\psi(g_k) - g_k|,$$

the near-identity fraction,

$$\pi_{\mathrm{id}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\{|\psi(g_k) - g_k| < 0.05\},$$

and the range of the learned auxiliary timesteps,

$$\mathrm{Range}(\psi) = \max_k \psi(g_k) - \min_k \psi(g_k).$$

A collapsed learned map would have small Δ_{id} and large π_{id}.

Table 8: Learned-pairing non-collapse diagnostic. The best learned TPC model maintains substantial separation from identity while achieving the lowest FID. In contrast, the overcoupled setting is much closer to identity and has poor FID, indicating that near-collapse is associated with a failed regime rather than the best-performing learned model.

Variant	**FID** $\downarrow$	**Mean** $\lvert\psi(t)-t\rvert$ $\uparrow$	**Near-id. frac.** $\downarrow$	**Range of** $\psi(t)$ $\uparrow$
Identity control $\psi(t)=t$	6.35	0.00	1.00	1.00
Fixed symmetric $\psi(t)=1-t$	3.59	0.50	0.05	1.00
Learned TPC, best	**3.19**	**0.24**	**0.04**	**0.83**
Learned TPC, moderate	3.88	0.20	0.12	0.74
Learned TPC, overcoupled	6.22	0.06	0.75	0.45

Table 8 shows that the best learned TPC run does not collapse to self-pairing: it has mean separation $\Delta_{\mathrm{id}} = 0.24$, near-identity fraction $\pi_{\mathrm{id}} = 0.04$, and auxiliary-timestep range 0.83. By contrast, the overcoupled setting is much closer to identity, with mean separation 0.06 and near-identity fraction 0.75, and its FID degrades to 6.22. Thus, identity collapse is not the source of the best learned-pairing result; when near-collapse appears, it corresponds to a poor-performing overcoupled regime.

7.3 Full rectified-flow results

Table 9: Full rectified-flow and probability-flow comparisons. The main paper reports the adaptive-RK45 full-simulation subset; this table includes both one-step/distillation and full-simulation regimes.

Method	**NFE** ($\downarrow$)	**IS** ($\uparrow$)	**FID** ($\downarrow$)	**Recall** ($\uparrow$)
One-Step Generation + Distil (Euler solver, N=1)				
1-Rectified Flow	1	9.08	6.18	0.45
2-Rectified Flow	1	9.01	4.85	0.50
3-Rectified Flow	1	8.79	5.21	0.51
TPC-1RF (ours)	**1**	**9.21**	**5.86**	**0.47**
TPC-2RF (ours)	**1**	**9.14**	**4.55**	**0.53**
TPC-3RF (ours)	**1**	**8.92**	**4.83**	**0.53**
Full Simulation (Runge–Kutta RK45), Adaptive N				
1-Rectified Flow	127	9.60	2.58	0.57
2-Rectified Flow	110	9.24	3.36	0.54
3-Rectified Flow	104	9.01	3.96	0.53
TPC-1RF (ours)	**127**	**9.78**	**2.15**	**0.60**
TPC-2RF (ours)	**110**	**9.42**	**2.95**	**0.58**
TPC-3RF (ours)	**104**	**9.18**	**3.45**	**0.55**

7.4 Training-phase diagnostics

Table 10: Training-phase diagnostics for TPC-FM relative to FM on CIFAR-10. The variance and correlation effects strengthen over training.

Training phase	**TPC-FM / FM grad var**$\downarrow$	**Grad corr**$\uparrow$	**Mid-time var ratio** ($[0.3, 0.7]$) $\downarrow$
Early	0.84 ± 0.03	0.34 ± 0.03	0.79 ± 0.03
Mid	0.71 ± 0.02	0.52 ± 0.03	0.66 ± 0.02
Late	0.66 ± 0.02	0.61 ± 0.03	0.60 ± 0.02

7.5 Training-dynamics and matched-cost visualizations

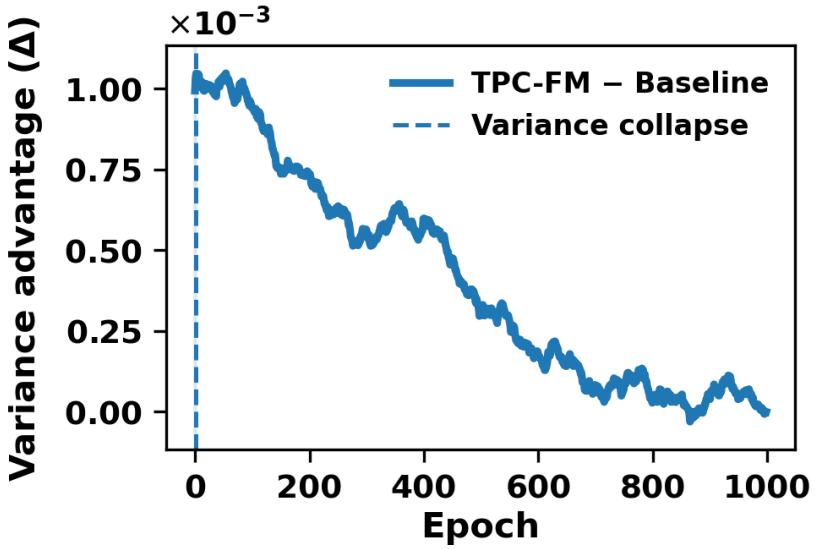


Figure 4: **Training diagnostic behavior.** TPC-FM shows lower measured gradient variance than the matched FM baseline throughout training.

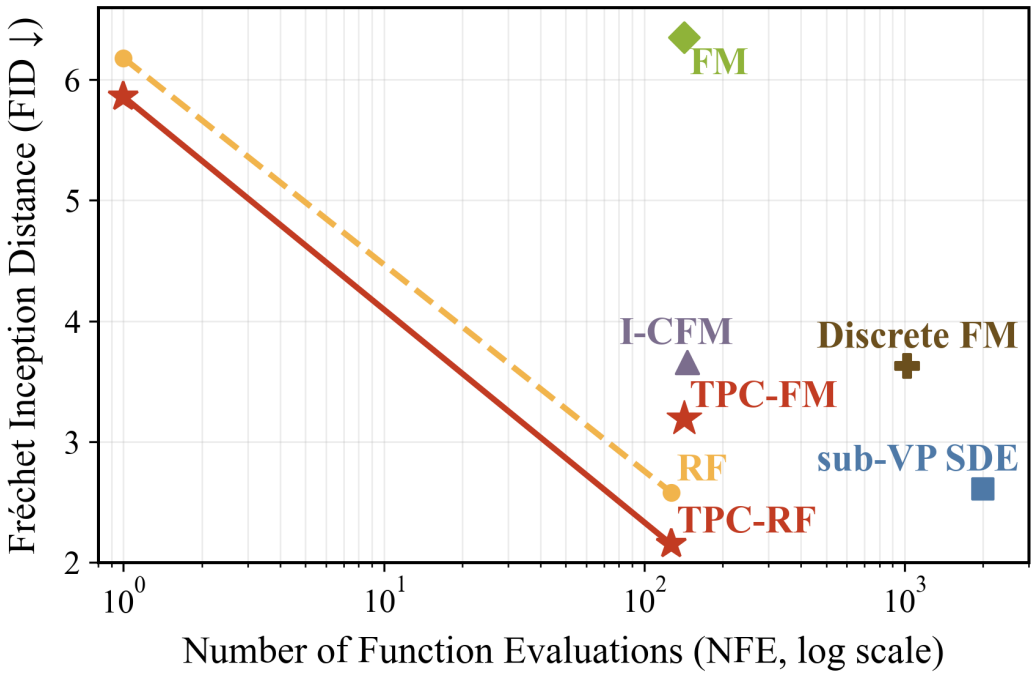
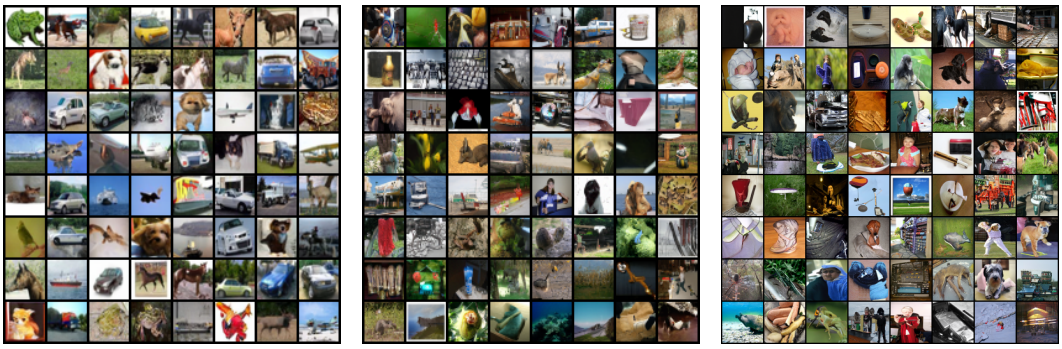


Figure 5: **Sample quality at matched inference cost**. Fréchet Inception Distance (FID ↓) versus number of function evaluations (NFE, log scale) on CIFAR-10. Temporal Pair Consistency (TPC) is applied only during training and leaves the inference-time architecture, probability path, sampler, and NFE unchanged. Under the same sampling settings as the corresponding baselines, TPC variants achieve lower FID.

8 Supplementary Qualitative Results

This section collects representative qualitative samples for unconditional generation and higher-resolution class-conditional generation.



(a) CIFAR-10 (b) ImageNet-32 (c) ImageNet-64

Figure 6: **Qualitative samples.** Unconditional generations on CIFAR-10, ImageNet-32, and ImageNet-64.

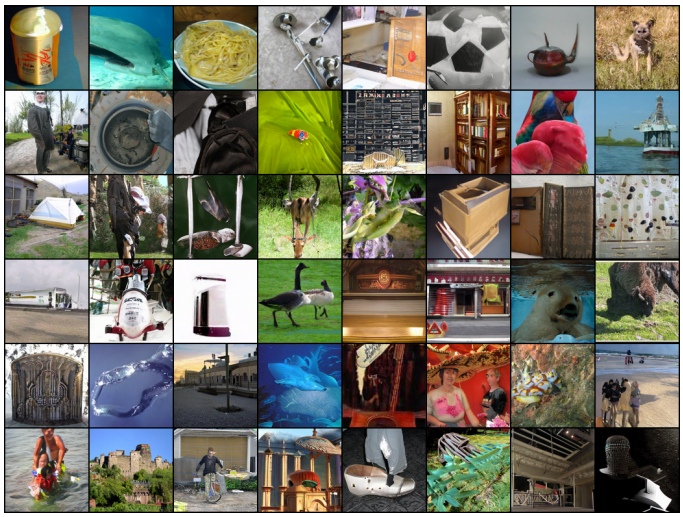


Figure 7: **ImageNet-128 qualitative samples.** Unconditional generations on ImageNet-128.



Figure 8: **Class-conditional ImageNet-256 qualitative samples.** Representative class-conditional generations on ImageNet-256.

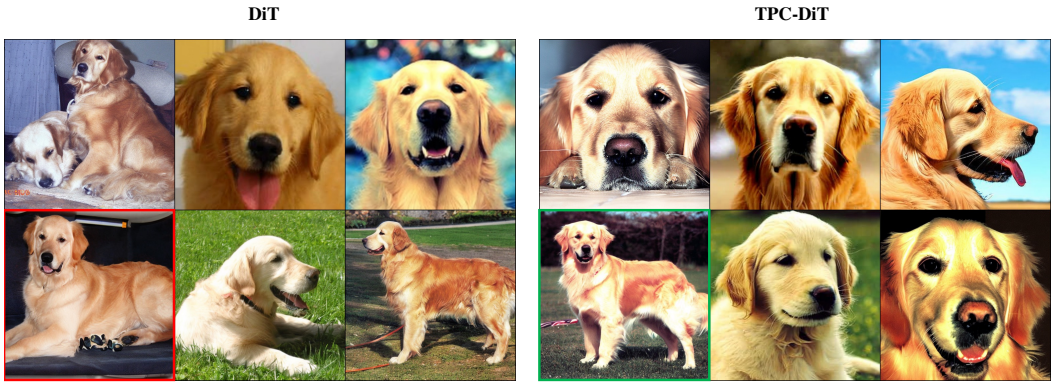


Figure 9: **Class-conditional ImageNet-512 qualitative samples.** Multi-page appendix figure with class-conditional ImageNet-512 comparisons; DiT samples are shown on the left and TPC-DiT samples are shown on the right.

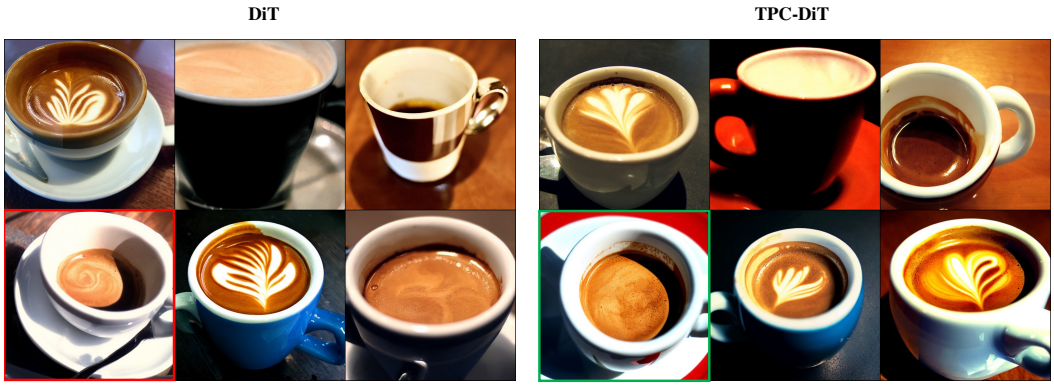


Figure 9: **Class-conditional ImageNet-512 qualitative samples.** Continued.



Figure 9: **Class-conditional ImageNet-512 qualitative samples.** Continued.



Figure 9: **Class-conditional ImageNet-512 qualitative samples.** Continued.

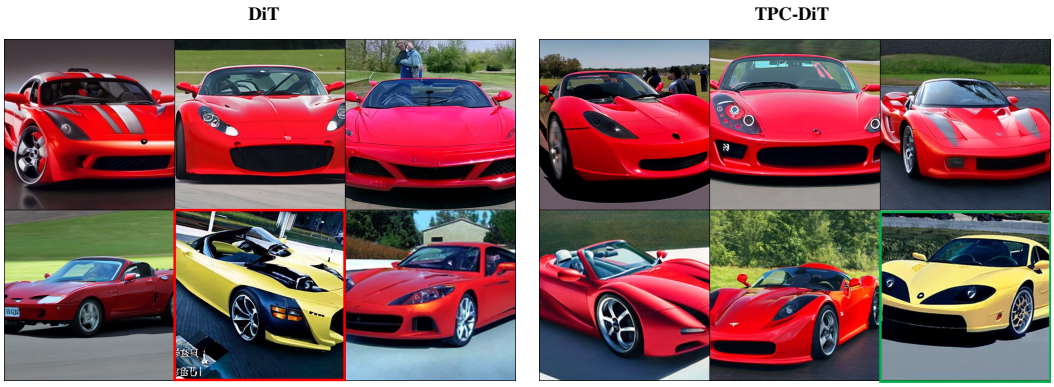


Figure 9: **Class-conditional ImageNet-512 qualitative samples.** Continued.

9 Impact Statement

This paper presents work whose goal is to advance the field of machine learning. The method is a training-time modification for flow-matching-style generative models and does not introduce new data sources, deployment settings, or application-specific risks beyond those generally associated with generative modeling. We do not identify additional societal consequences that require special discussion.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Introduction 1 state the matched-cost empirical scope, and the claims are supported by Method 3 and Experiments 4. The paper does not claim formal theory or universal NFE reduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion includes a limitations statement on broader text-to-image and modality coverage, and Experiments 4 explicitly frame the matched-setting scope. The appendix further separates protocol details, diagnostics, and qualitative/contextual results.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results. The equations in Method 3 define the training objective and pairing mechanisms used in the empirical study.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Method 3, Experiments 4, and Appendix Sections 6– 7 specify the objective, pairing rules, algorithm, datasets, metrics, solver/NFE accounting, diagnostic measurements, and matched-cost protocol.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: No public code release accompanies this paper. The manuscript nevertheless provides the algorithmic objective, evaluation protocol, and public benchmark setup needed to guide reproduction.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experiments 4 specifies datasets, metrics, solvers, NFE accounting, noise settings, and matched comparison protocol, while Appendix Section 6 gives the training algorithm and diagnostic definitions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Diagnostic quantities in Table 2 and Appendix Table 10 include variability estimates.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix 6.3 reports the compute environment used for the reported experiments, including the internal GPU cluster, NVIDIA H200 GPUs with 141GB HBM3e memory per GPU, local high-throughput training storage, and the 32-GPU H200 configuration used for the largest ImageNet-512×512 DiT-XL/2 run. It also notes that lower-resolution CIFAR-10 and ImageNet runs used fewer GPUs and that smaller pilot/tuning runs were conducted outside the final reported runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses standard public image-generation benchmarks and does not involve human subjects, private data, surveillance, or personally identifying information. The study conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The appendix includes an Impact Statement. The work is foundational generative-modeling research; it improves training of existing model classes and does not introduce new deployment settings, data sources, or application-specific risks beyond those generally associated with generative models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained image generator, scraped dataset, or other high-risk asset requiring release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Existing datasets, baselines, and model families are credited through the corresponding citations in Experiments 4 and the references. The work uses standard public benchmarks and does not repackage or redistribute those assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce or release a new dataset, benchmark, codebase, or pretrained model as a new asset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the core method, experiments, or scientific claims. Any ordinary writing or editing assistance would not affect the methodology or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.