XYZFlow: Scaling Multidimensional Shortcut Flows for Efficient Generative Modeling

Anonymous CVPR submission

Paper ID 21506

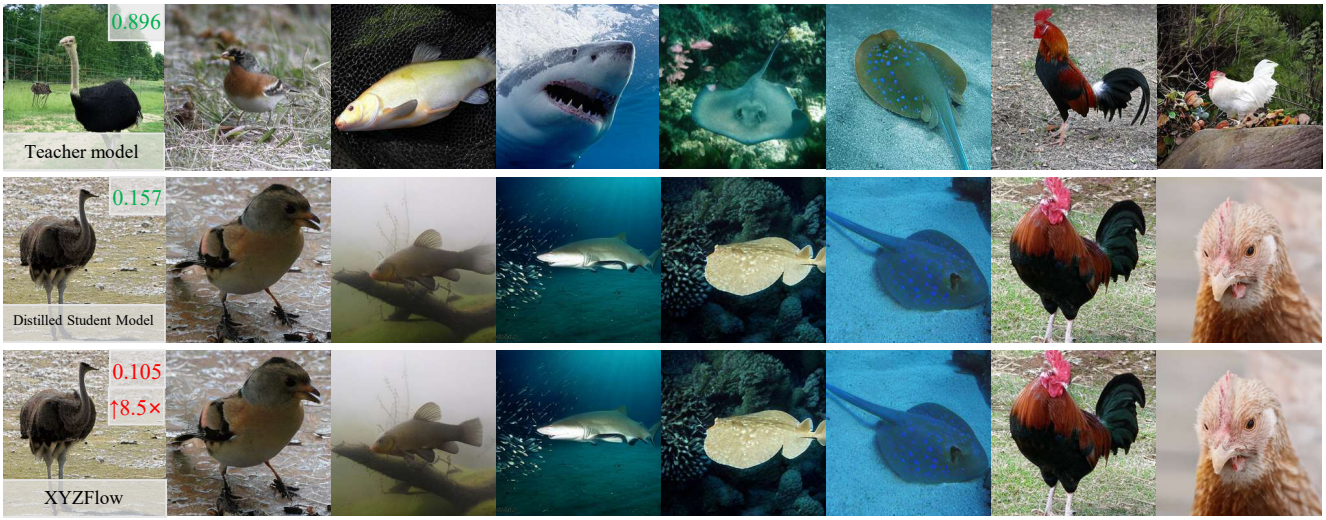

Figure 1. Visual comparison demonstrating the efficiency of XYZFlow. Our 1.1B-parameter model achieves an **8.5× faster** generation time than the teacher model and an additional **1.5× speedup** over the base student distillation, with no perceptible loss in quality.

Abstract

The pursuit of high-fidelity image generation faces a fundamental trade-off between sampling speed and output quality. While diffusion models excel in quality, their iterative nature incurs high computational costs. Current efficient methods primarily focus on distilling pre-trained models into few-step samplers; however, this distillation process is challenging and heavily reliant on teacher model quality. In this paper, we introduce XYZFlow, a novel framework that rethinks this paradigm through multidimensional scaling of flow matching. Unlike MeanFlow's single-step deterministic mapping, our approach intensively scales the expressive power of generative models by enhancing the uniqueness and learnability of probability paths through structured, multidimensional conditioning. Theoretically, we frame autoregressive modeling as an implicit flow straightening mechanism, where expanding contextual constraints reduce trajectory ambiguity. XYZFlow implements this via two orthogonal scaling dimensions: (1) Temporal scaling through non-Markovian conditioning on the full denoising history, and (2) Spatial scaling through next-shortcut prediction, where patches are generated sequentially using the complete denoising trajectories of preceding patches as priors. This multidimensional conditioning constructs a high-dimensional coordinate system for probability flows, enforcing mapping uniqueness. Extensive evaluations demonstrate XYZFlow achieves state-of-the-art performance, with 7.2–8.5× speedup over teachers while maintaining competitive FID. Notably, XYZFlow-B (172M) outperforms the one-step model MeanFlow-XL/2+ (676M), demonstrating that our structured shortcut design establishes a more parameter-efficient scaling dimension and achieves superior quality-latency trade-offs compared to simply enlarging models or compressing sampling steps.

1. Introduction

Generative models, particularly diffusion probabilistic models, have revolutionized synthetic data generation across various modalities [11, 12, 25, 29, 32]. The dominant paradigm involves a gradual forward process that incremen-

CVPR #21506

tally corrupts data with noise, followed by a learned reverse process for iterative data reconstruction. While models like DDPM [11] and Score-SDE [30] achieve remarkable quality, this performance comes at a substantial computational cost [13, 19, 20], often requiring hundreds of neural function evaluations per sample. Such cost makes these models impractical for real-time applications.

The pursuit of efficiency has centered on a key insight: few-step generation quality fundamentally depends on the uniqueness of the noise-to-data trajectory mapping [2, 6, 16, 26]. This uniqueness enables effective distillation by reducing the problem from learning complex distributions to fitting deterministic functions. Pioneering methods like Rectified Flows [16], Consistency Models [35] and Shortcut Models [6] address this by constructing straight, deterministic probability flows through novel training objectives. However, these approaches primarily focus on improvements to distillation algorithms themselves, which is a challenging and model-dependent endeavor [7, 26, 28].

Despite recent progresses, we identify another fundamental challenge that remains largely unexplored: *how can we scale the expressive power of generative models under strict sampling step constraints, without relying solely on distillation strategies? More profoundly, can we design probability flows that are intrinsically more unique and learnable through model architecture?* As conceptually visualized in Figure 2(a), the ambiguous trajectories in conventional denoising stem from a lack of focused constraints.

In this paper, we consider a new scaling paradigm. Instead of extensive scaling through added parameters or steps, we scale *intensively* by enhancing probability flow expressivity via structured, multidimensional conditionalization. We reinterpret autoregressive modeling not merely as a generative strategy, but as an implicit mechanism for flow enhancement and uniqueness enforcement [15]. The expanding autoregressive context imposes progressively specific constraints, reducing flow variance and straightening trajectories. This perspective inherits the insight from flow straightening methods [1, 6, 16] where deterministic paths are crucial for efficient distillation, demonstrating that *structured conditioning* enforces such uniqueness.

Guided by this insight, we introduce **XYZFlow**, a framework that scales flow matching along two orthogonal dimensions for high-fidelity few-step generation, complementary to the prevailing path of distillation-based step compression. **(1) Temporal Scaling:** We condition each flow step on the complete history of previous states, creating a temporal coordinate system that straightens trajectories. This transforms denoising from Markovian to non-Markovian, where the KV cache of past states acts as a conditioning anchor, inspired by recent advances in recurrent diffusion processes [9]. **(2) Spatial Scaling:** We propose *Next Shortcut Prediction* based on next-patch prediction.

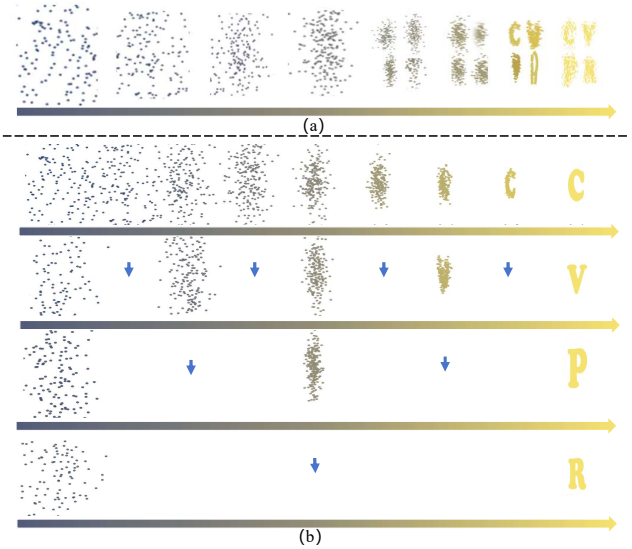

Figure 2. (a) Conventional one-shot denoising suffers from overlapping and ambiguous probability paths (blurred results) as the model attempts to denoise the entire image at once. (b) Our **Next Shortcut Prediction** paradigm: Denoising proceeds sequentially patch-by-patch (e.g., for patches **C, V, P, R**). The **rightward small arrows** trace the denoising trajectory of each patch over time. Crucially, the **downward blue arrows** transfer the complete denoising trajectory of the preceding patch as a powerful prior. This allows subsequent patches to leverage the established context, straightening their paths and requiring fewer denoising steps (longer horizontal sequences) to achieve high fidelity.

By dividing images into grids (e.g., 2×2), this mechanism sequentially generates patches. Unlike standard patch-wise generation that treats patches independently, our method explicitly transfers the *denoising trajectory* (not just the final output) as an effective prior for subsequent patches. As illustrated in Figure 2(b), the full denoising trajectory of the first patch serves as a conditional guidance, enabling faster generation without quality loss.

Theoretically, we demonstrate that multidimensional scaling equips probability paths with high-dimensional coordinate systems. While flow straightening methods seek unique points in one-dimensional flows, our approach secures uniqueness through orthogonal dimensional coordinates. We ground our method in two principles: (1) increasing condition drives reverse process variance toward zero, ensuring mapping uniqueness [8, 18, 31, 34, 40]; (2) autoregressive trajectories of preceding patches guide subsequent predictions, straightening paths in few-step regimes [9, 23, 37, 39]. Our contributions are threefold:

- **Novel Scaling Paradigm.** We propose to scale generative models by enhancing probability flow expressivity through structured multidimensional conditionalization, advocating that scaling *constraint dimensionality* provides a principled path to mapping uniqueness.

- **XYZFlow Framework.** We introduce a practical framework implementing temporal and spatial flow scaling, featuring **Next Shortcut Prediction** for efficient inference-time cross-patch implicit trajectory guidance.
- **Principled Theoretical Justification.** We establish a theoretical framework formalizing autoregressive modeling as explicit flow enhancement, and empirically demonstrate competitive few-step generation on ImageNet 256×256, showing dimensional scaling as a promising alternative to conventional methods.

2. Related Work

Few-step Diffusion and Flow Matching. Diffusion models [11, 29, 32, 33] and their flow matching extensions [1, 16, 17] have established a powerful framework for generative modeling. Current research on efficient sampling primarily follows a path of *extensive scaling*, focusing on refining distillation algorithms or training objectives. Distillation-based methods [7, 21, 26, 28, 41] aim to compress pre-trained models, while consistency-type approaches [8, 18, 31, 34, 40] enforce self-consistency constraints along trajectories. Recent works like Flow Map Matching [2] and Shortcut Models [6] further explore self-consistency principles to straighten probability paths, with Inductive Moment Matching [43] modeling the self-consistency of stochastic interpolants at different time steps. While these methods have advanced the state of the art, their reliance on distillation algorithm improvements represents a form of extensive scaling that faces fundamental challenges in model dependency and optimization complexity. In contrast, our work addresses the core challenge of trajectory uniqueness by introducing *intensive scaling*—enhancing flow expressivity through multidimensional conditionalization rather than pursuing better distillation algorithms for existing flows.

Autoregressive Models for Visual Generation. Autoregressive image generation has evolved from discrete tokenization [5, 14, 22] to continuous representations that avoid quantization errors [15, 23, 24, 38]. Methods like MAR [15] and DISA [42] combine autoregressive modeling with diffusion processes, while acceleration techniques focus on caching [39] or speculative decoding [37]. FAR [9] replaces the diffusion head of MAR [15] with a shortcut model, accelerating through architectural changes. While these approaches employ autoregression primarily as a generative mechanism, our framework reinterprets autoregressive modeling in multiple dimension as an implicit mechanism for flow enhancement and uniqueness enforcement.

3. The Proposed XYZFlow Framework

We introduces XYZFlow, a framework designed to enhance the expressivity of flow models via multidimensional conditioning. Building on the concept that autoregressive modeling acts as an effective mechanism for flow straightening by incrementally imposing constraints, we propose a novel training objective called Next Shortcut Prediction. This objective facilitates efficient generation through multi-dimensional conditioning. We conceptualize the expanding autoregressive context as a series of progressively specific constraints that reduce variance in the probability flow and straighten trajectories. This view extends existing insights from flow straightening methods by showing how structured conditional information ensures path uniqueness. Within this framework, Next Shortcut Prediction operationalizes the principle of intensive scaling. Specifically, it leverages spatial constraints to construct a high-dimensional coordinate system that effectively enforces flow uniqueness and straightens trajectories.

3.1. Conceptual Foundation: Autoregressive Modeling as Flow Enhancement

Traditional autoregressive approaches frame image generation as a sequence of conditional predictions. Given an image divided into patches $\langle \mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^P \rangle$, the generation process is formulated as:

$$p(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^P) = \prod_{p=1}^{P} p(\mathbf{x}^p \mid \mathbf{x}^1, \ldots, \mathbf{x}^{p-1}). \quad (1)$$

While this formulation is mathematically sound, we reconceptualize it through the lens of flow enhancement. The growing context $\mathbf{x}^1, \ldots, \mathbf{x}^{p-1}$ acts as a set of progressively stronger constraints, which reduces the variance of the conditional distribution $p(\mathbf{x}^p|...)$ and straightens the probability flow path from noise to data. This conceptual shift allows us to leverage autoregressive structure not just for sequential prediction, but for intrinsically making the flow more unique and deterministic.

Formally, we define flow enhancement as the process where conditional information C transforms a base probability flow $p(\mathbf{x})$ into a conditioned flow $p(\mathbf{x}|C)$ with reduced path variance: $\mathbb{V}[\mathbf{x}_t|C] < \mathbb{V}[\mathbf{x}_t]$, leading to straighter and more deterministic trajectories. This variance reduction directly contributes to mapping uniqueness. For detailed theoretical proofs, please refer to our supplementary.

3.2. Motivating Observation: Progressive Constraint Strengthening

Our approach is motivated by the empirical observation that as more patches are generated, the conditional distribution becomes more constrained, making subsequent patches easier to sample. As illustrated in Figure 3, this manifests in three key phenomena:

(1) Next patches can be better predicted at later generation stages. When we probe the conditional strength

CVPR
#21506

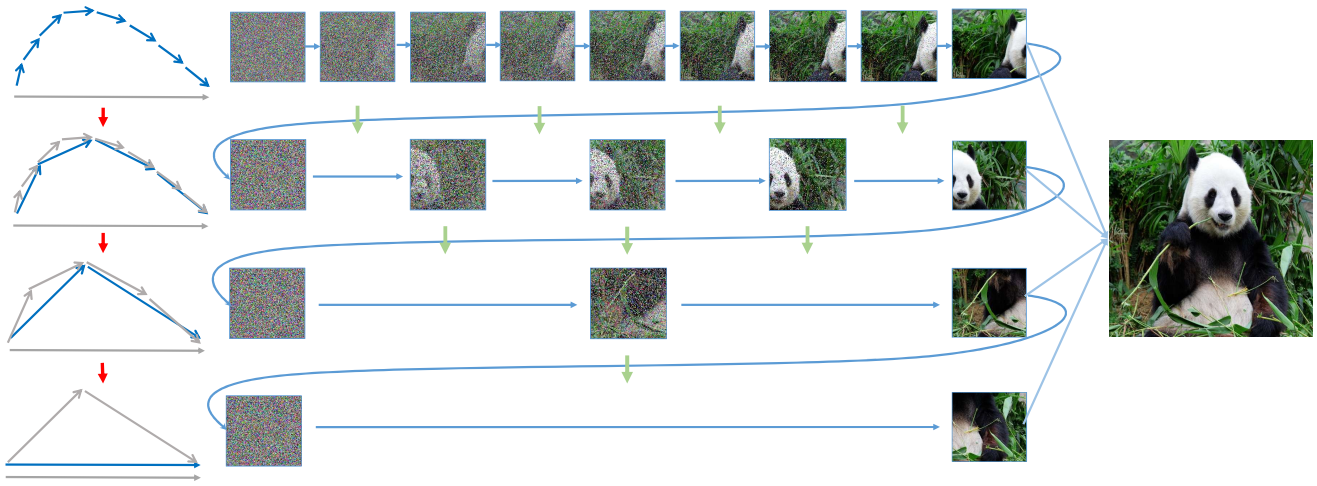

Figure 3. Next Shortcut Prediction in XYZFlow Framework. (Top-Left) Flow diagram showing the generation sequence, where a blue curve represents progressively strengthening constraints. (Top-Right) Visualization of a non-uniform patch-based denoising process: the first image patch undergoes the most denoising steps, while subsequent patches are generated with fewer steps (”shortcuts”). This forms a long autoregressive sequence where the denoising flow from prior patches (green and blue arrows) guides the denoising of subsequent ones, providing a strong prior.

by predicting $\mathbf{x}^p$ based on the representation of previously generated patches, predictions for early positions are blurry and lack detail, while predictions for later positions become increasingly precise. This demonstrates that accumulated context provides stronger conditional guidance, as shown in the panda image sequence (top-right of Figure 3).

(2) The variance of latent patch distributions decreases for later patches. When sampling multiple possible patches for each position during generation, the variance among sampled patches is high for early positions but decreases significantly for later positions, indicating a more concentrated distribution under strong conditioning. This is visually represented by the progression from noisy to clean patches in the bottom row of Figure 3.

(3) Denoising paths become straighter for later patches. Following Rectified Flow theory, we measure path straightness using:

$$S(\{\mathbf{x}_t\}_{t=0}^{1}, \mathbf{z}) = \mathbb{E}_{t\sim[0,1]}\left[\|(\mathbf{x}_1 - \mathbf{x}_0) - \mathbf{v}_\theta(\mathbf{x}_t \mid t, \mathbf{z})\|^2\right]. \quad (2)$$

Our experiments have demonstrated that S decreases for patches generated later in the sequence, validating that strong contextual conditioning can effectively straighten the flow. The blue curve in Figure 3 (top-left) illustrates this straightening effect.

3.3. Multidimensional Conditioning

Based on these observations, XYZFlow implements a dual-path conditioning architecture that enhances the probability flow along both **temporal** and **spatial** dimensions through Next Shortcut Prediction.

Temporal Conditioning: Intra-Patch Trajectory Conditioning We enhance the flow matching process for each patch by conditioning it on its own generation history. Specifically, for a patch $\mathbf{x}^p$, the conditioning signal at time t is its entire state trajectory from the beginning of generation up to, but not including, the current time t. We denote this history as $\mathcal{H}_t^p = \{\mathbf{x}_\tau^p\}_{\tau=0}^{t-\Delta t}$. The temporal conditioning loss for the patch is then defined as the deviation of the predicted flow from the true conditional flow, given this historical context:

$$\mathcal{L}_{\text{temp}}^p = \mathbb{E}_{t,\mathbf{x}_0^p,\mathbf{x}_1^p}\|v_\theta(\mathbf{x}_t^p|t, \mathcal{H}_t^p) - (\mathbf{x}_1^p - \mathbf{x}_0^p)\|^2 \quad (3)$$

This self-conditioning acts as a strong prior, stabilizing the generation path by providing a temporal coordinate system for the flow.

Spatial Conditioning: Inter-Patch Trajectory Conditioning The spatial dimension implements conditioning where each patch's generation depends on the complete trajectories of all previously generated patches. As illustrated in Figure 4, the key innovation is that each patch conditions not only on the final content of previous patches, but on their complete generation trajectories, providing a much richer contextual signal that enhances flow expressivity across the spatial domain:

$$p(\mathbf{x}^p|\mathbf{x}^1, \ldots, \mathbf{x}^{p-1}) = p(\mathbf{x}^p|\mathcal{T}_{<p}) \quad (4)$$

where $\mathcal{T}_{<p} = \{\tau^1, \tau^2, \ldots, \tau^{p-1}\}$ and $\tau^i = \{\mathbf{x}_t^i\}_{t=0}^1$ represents the complete generation trajectory of patch i. Conditioning on full trajectories $\mathcal{T}_{<p}$ rather than just final states

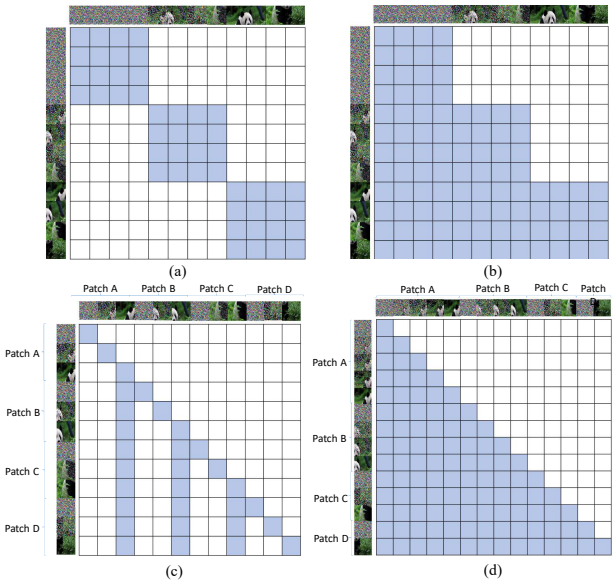

Figure 4. Illustration of attention mechanisms for image generation. (a) Vanilla Image Generation: Standard full-image denoising with independent patch processing. (b) Autoregressive in Denoising Dimension: Sequential denoising across patches over time. (c) Next-Patch Prediction: Complete denoising of one patch before starting the next. (d) Next Shortcut Prediction: Early patches undergo more denoising steps, with full denoising trajectories of previous patches conditioning subsequent ones.

provides significantly stronger constraints: each trajectory τ^i adds multiple temporal anchor points that collectively reduce the solution space for generating $\mathbf{x}^p$, making the reverse process more deterministic. The attention patterns shown in Figure 4 demonstrate how different attention mechanisms can effectively integrate trajectory condition.

3.4. Distillation for Next Shortcut Prediction

The core innovation of XYZFlow is Next Shortcut Prediction, which implements a paradigm shift from resource-intensive scaling to constraint-intensive scaling. Unlike traditional approaches that increase model size or training steps, we scale the constraint dimensionality by training the model to generate effectively under progressively stronger constraints from $\mathcal{T}_{<p}$. As conceptualized in Figure 3, Next Shortcut Prediction trains the model to leverage rich contextual information for accelerated generation. We define a progressive denoising schedule where each patch p is assigned a decreasing number of denoising steps:

$$T(p) = T_{\text{full}} - \Delta T \cdot (p-1) \quad \text{for} \quad p = 1, \ldots, P, \tag{5}$$

with the constraint $T(p) \geq T_{\min} > 0$.

Our training objective is formulated as teacher model trajectory distillation. The key insight is to enhance the uniqueness and straighten the probability flow by imposing powerful, structured constraints. We achieve this through an autoregressive formulation:

$$p(\mathbf{x}_0^p \mid \mathcal{T}_{<p}) = p_{\text{prior}}(\mathbf{x}_{T(p)}^p) \times \prod_{t=1}^{T(p)} p(\mathbf{x}_{t-1}^p \mid \mathbf{x}_{T(p):t}^p, \mathcal{T}_{<p}), \tag{6}$$

where $\mathbf{x}_{T(p):t}^p = [\mathbf{x}_{T(p)}^p, \mathbf{x}_{T(p)-1}^p, \ldots, \mathbf{x}_t^p]$ denotes the historical denoising trajectory. This formulation provides two fundamental advantages that embody our intensive scaling principle: (i) It equips each denoising step with a high-dimensional coordinate system. The combination of the spatial context from previous patches ($\mathcal{T}_{<p}$) and the temporal context from the entire historical trajectory ($\mathbf{x}_{T(p):t}^p$) imposes a highly specific constraint. This drastically reduces the variance of the reverse process, transforming the mapping from $\mathbf{x}_t^p$ to $\mathbf{x}_{t-1}^p$ from an ambiguous, one-to-many problem into a nearly deterministic, one-to-one function, thereby straightening the probability flow path. (ii) It enables synergistic information fusion. To predict $\mathbf{x}_{t-1}^p$ at every step, the model learns to integrate both coarse-grained and fine-grained information. The recent denoised sample $\mathbf{x}_t^p$ is the best source for fine-grained details, while the historical trajectory closer to $\mathbf{x}_{T(p)}^p$ provides better coarse-grained structural information.

We aim to estimate $p(\mathbf{x}_{t-1}^p \mid \mathbf{x}_{T(p):t}^p, \mathcal{T}_{<p})$, which, under our strong conditioning, approximates a Dirac delta distribution. This is achieved within the Flow Matching framework by defining the mapping function:

$$\mathbf{x}_{t-1}^p = G(\mathbf{x}_{T(p):t}^p, \mathcal{T}_{<p}, t) := \mathbf{x}_t^p + (\gamma(t-1) - \gamma(t)) \cdot v_\theta(\mathbf{x}_{T(p):t}^p, t, \mathcal{T}_{<p}), \tag{7}$$

which is approximated by our student neural network v_θ using an Euler step. Here, γ is the noise schedule. The complete training objective integrates multidimensional conditioning with this progressive schedule. It distills the teacher's trajectory by regressing the target sample:

$$\mathcal{L}_{\text{NextShortcut}} = \mathbb{E}_{p\sim[1,P]} \left[\sum_{t=1}^{T(p)} \left\| G_\theta(\mathbf{x}_{T(p):t}^p, t, \mathcal{T}_{<p}) - \mathbf{x}_{t-1}^p \right\|_2^2 \right]. \tag{8}$$

Here, $G_\theta(\mathbf{x}_{T(p):t}^p, t, \mathcal{T}_{<p}) = \mathbf{x}_t^p + (\gamma(t-1) - \gamma(t)) \cdot v_\theta(\mathbf{x}_{T(p):t}^p, t, \mathcal{T}_{<p})$ represents the student's one-step prediction. The transformer architecture allows computing G_θ for all t simultaneously by using an attention mask. We design the attention mask to be block-wise causal, allowing the model to use the entire trajectory history $\mathbf{x}_{T(p):t}^p$ as context, which is the most flexible and effective option. This objective directly embodies our intensive scaling principle: it trains the student network to predict the optimal denoising path using both temporal (historical trajectory) and spatial (previous patches) conditioning. The yellow arrows in Figure 3 (bottom) illustrate this accelerated generation path. Our framework can also benefit from an additional discriminator loss applied to the final generated patch $\hat{\mathbf{x}}_0^p$. This ad-

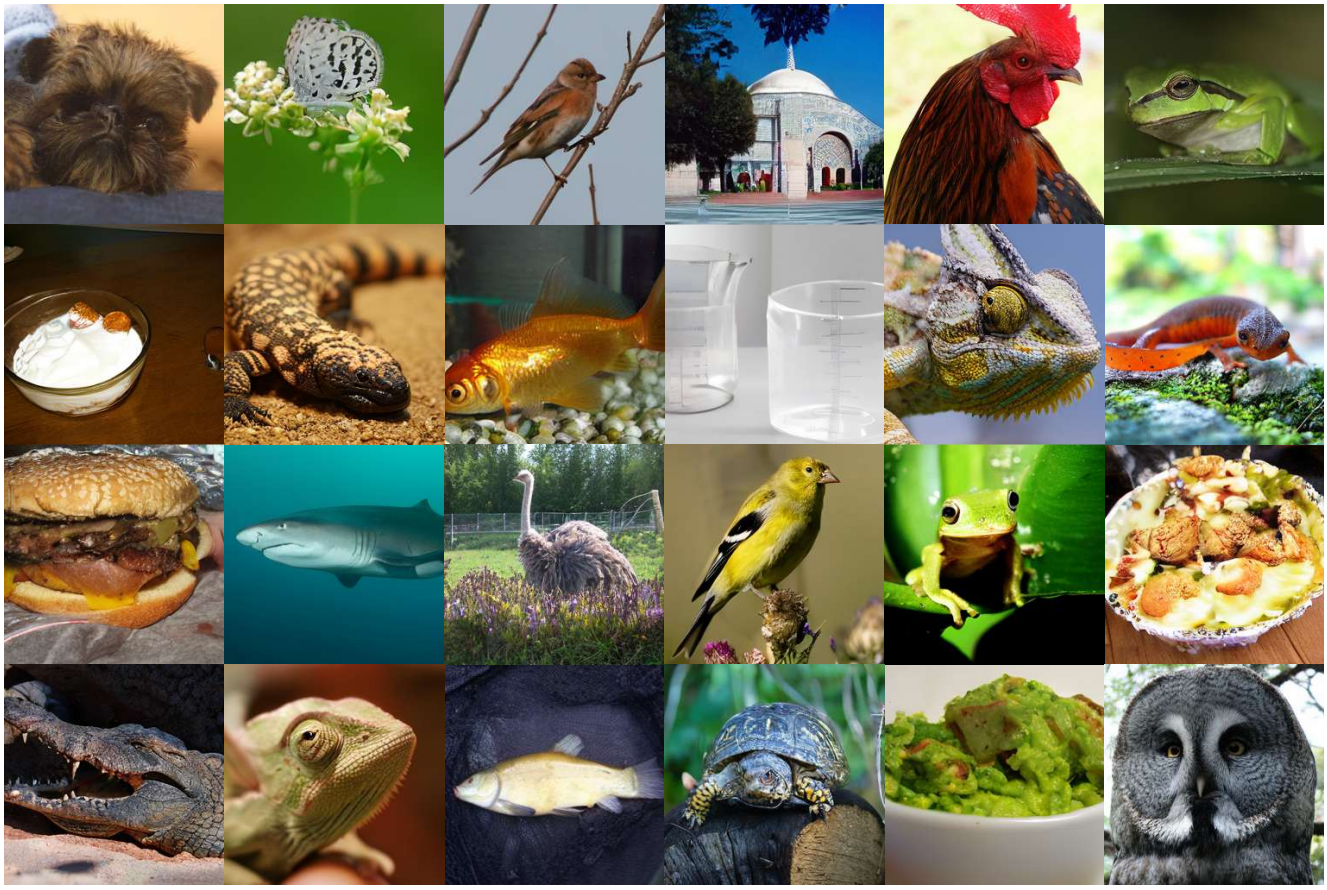

Figure 5. Randomly selected examples of generated images from XYZFlow. XYZFlow shows high-quality generative modeling abilities.

versarial training, which uses real data as supervision, further enhances the high-frequency details in the generated outputs.

During inference, the model generates the first patch with the full step budget $T(1) = T_{\text{full}}$ to establish a robust anchor. Each subsequent patch $p \geq 2$ is generated autoregressively: starting from $\mathbf{x}^p_{T(p)} \sim p_{\text{prior}}$, at each step t, the model predicts $\hat{\mathbf{x}}^p_{t-1} = G_\theta(\hat{\mathbf{x}}^p_{T(p):t}, t, \mathcal{T}_{<p})$ based on the entire history of predictions $\hat{\mathbf{x}}^p_{T(p):t} = [\mathbf{x}^p_{T(p)}, \hat{\mathbf{x}}^p_{T(p)-1}, \ldots, \hat{\mathbf{x}}^p_t]$. The information of the historical predictions is efficiently managed using a key-value cache. This process leverages the accumulated context $\mathcal{T}_{<p}$ and the learned ability to exploit constraints, achieving significant speedup while maintaining generation quality through learned constraint exploitation.

4. Experiments and Results

We empirically validate the efficacy of **XYZFlow**, focusing on the theoretical claims in Section 3. Our experiments demonstrate that: (1) Multidimensional conditioning straightens the probability flow for subsequent patches, enabling better generation quality; (2) The Next Shortcut Prediction objective effectively trains models to utilize accumulated context for accelerated generation by decreasing denoising steps; (3) XYZFlow achieves promising efficiency-quality trade-offs in image synthesis.

4.1. Experimental Setup

We train and evaluate XYZFlow on the ImageNet 256×256 class-conditional generation benchmark [3]. Training is conducted on 8 NVIDIA H100 GPUs for 300K steps with a batch size of 128 and a learning rate of 0.0001. To better evaluate scalability, we employ three autoregressive teacher models of varying sizes, including Base (172M), Large (608M), and Huge (1.1B) [24]. ODE trajectory data is generated by running each teacher for 50 steps with a classifier-free guidance scale of 2.3, pre-computing 2.5M trajectories for distillation. We comprehensively evaluate sample quality using four established metrics: Fréchet Inception Distance (FID) [10], Inception Score (IS) [27], and Precision/Recall [4] to quantify fidelity and diversity. Inference time (shown in seconds) and speed-up relative to baseline models are reported to measure efficiency.

Model	#params	AR steps	Diff steps	FID↓	IS↑	Pre.↑	Rec.↑	Time (s)↓	Speed-Up↑
Base Models (170M-208M parameters)									
MAR-B [15]	208M	256	100	2.31	281.7	0.82	0.57	0.650	1.0×
		64	50	2.39↑0.08	281.0↓0.7	0.82	0.57	0.134↓0.516	4.9×↑3.9
FlowAR-S [23]	170M	5	25	3.70↑1.39	235.1↓46.6	0.81↓0.01	0.51↓0.06	0.024↓0.626	27.1×↑26.1
xAR-B [24]	172M	4	50	1.67↓0.64	265.2↓16.5	0.80↓0.02	0.62↑0.05	0.130↑0.520	5.0×↑4.0
XYZFlow-B (w/o GAN)	172M	4	5→2	2.02↓0.29	261.1↓20.6	0.80↓0.02	0.58↑0.01	0.018↓0.632	36.1×↑35.1
XYZFlow-B (w/ GAN)	172M	4	5→2	1.63↓0.68	268.5↓13.2	0.81↓0.01	0.62↑0.05	0.018↓0.632	36.1×↑35.1
Large Models (479M-676M parameters)									
DiT/XL-2 (25-step)	676M	-	25	2.89	230.2	0.80	0.57	0.494	1.0×
MeanFlow-XL/2 (w/o CFG)	676M	-	1	3.43↑0.54	-	-	-	0.009↓0.485	54.9×↑53.9
MeanFlow-XL/2	676M	-	1	2.93↑0.04	-	-	-	0.018↓0.476	27.4×↑26.4
MeanFlow-XL/2+	676M	-	1	2.20↓0.69	-	-	-	0.018↓0.476	27.4×↑26.4
Step Distill	676M	-	2	10.92↑8.03	167.0↓63.12	0.68↓0.12	0.52↓0.05	0.033↓0.461	15.0×↑14.0
ARD (w/o GAN)	676M	-	2	6.29↑3.40	188.0↓42.15	0.74↓0.06	0.56↓0.01	0.034↓0.460	14.5×↑13.5
Step Distill (w/o GAN)	676M	-	4	10.25↑7.36	181.6↓48.62	0.70↓0.10	0.47↓0.10	0.065↓0.429	7.6×↑6.6
ARD (w/o GAN)	676M	-	4	4.32↑1.43	209.0↓21.2	0.77↓0.03	0.57	0.066↓0.428	7.5×↑6.5
Step Distill (w/ GAN)	676M	-	4	3.84↑0.95	221.1↓9.1	0.78↓0.02	0.55↓0.02	0.065↓0.429	7.6×↑6.6
ARD (w/ GAN)	676M	-	4	1.84↓1.05	235.8↓5.6	0.80	0.62↑0.05	0.066↓0.428	7.5×↑6.5
MAR-L [15]	479M	256	100	1.78↓1.11	296.0↑65.8	0.81↑0.01	0.60↑0.03	1.102↑0.608	0.4×↓0.6
		64	50	1.86↓1.03	294.0↑63.8	0.80	0.61↑0.04	0.250↓0.244	2.0×↑1.0
FlowAR-L [23]	589M	5	25	1.87↓1.02	273.1↑42.9	0.80	0.62↑0.05	0.124↓0.370	4.0×↑3.0
xAR-L [24]	608M	4	50	1.28↓1.61	292.5↑62.3	0.82↑0.02	0.62↑0.05	0.394↑0.100	1.3×↑0.3
XYZFlow-L (w/o GAN)	608M	4	5→2	1.79↓1.10	265.2↓35.0	0.81↑0.01	0.61↑0.04	0.050↓0.444	9.9×↑8.9
XYZFlow-L (w/ GAN)	608M	4	5→2	1.25↓1.64	295.8↑65.6	0.83↑0.03	0.63↑0.06	0.050↓0.444	9.9×↑8.9
Huge Models (943M-2.0B parameters)									
FlowAR-H [23]	1.9B	5	50	1.67	276.3	0.80	0.62	0.423	1.0×
VAR-d30 [36]	2.0B	10	-	1.92↑0.25	323.1↑46.8	0.82↑0.02	0.59↓0.03	0.039↓0.384	10.8×↑9.8
MAR-H [15]	943M	256	100	1.55↓0.12	303.7↑27.4	0.81↑0.01	0.62	1.957↑1.534	0.2×↓0.8
		64	50	1.65↓0.02	299.8↑23.5	0.80	0.62	0.462↑0.039	0.9×↓0.1
xAR-H [24]	1.1B	4	50	1.24↓0.43	301.6↑25.3	0.83↑0.03	0.64↑0.02	0.896↑0.473	0.5×↓0.5
XYZFlow-H (w/o GAN)	1.1B	4	5→2	1.73↓0.06	271.5↓4.8	0.82↑0.02	0.62	0.105↓0.318	4.0×↑3.0
XYZFlow-H (w/ GAN)	1.1B	4	5→2	1.22↓0.45	304.2↑27.9	0.84↑0.04	0.64↑0.02	0.105↓0.318	4.0×↑3.0

Table 1. **System-level method comparison** on ImageNet 256×256. Models are organized by parameter count from small to large. Colored numbers indicate performance change relative to baseline models.

4.2. Main Results and Analysis

Table 1 presents a comprehensive system-level comparison on ImageNet 256 × 256, demonstrating XYZFlow's framework advantages across model scales. The consistent performance improvements in both standard and GAN-enhanced configurations validate XYZFlow's core methodology. Compared to the teacher models, XYZFlow achieves substantial acceleration improvements while maintaining or enhancing quality: XYZFlow attains 36.1× speed-up (7.2× improvement over xAR-B's 5.0×) with better FID (1.63 vs. 1.67), XYZFlow-L reaches 9.9× (7.6× improvement over xAR-L's 1.3×), and XYZFlow-H achieves 4.0× (8× improvement over xAR-H's 0.5×). The framework's effectiveness is further demonstrated by its scalable performance across categories: at the Base level, XYZFlow establishes superior efficiency-quality trade-offs over FlowAR-S (36.1× vs. 27.1× acceleration with FID 1.63 vs. 3.70); at the Large level, it outperforms MAR-L (9.9× vs. 2.0×) and FlowAR-L (9.9× vs. 4.0×); at the Huge level, it achieves better FID than VAR (1.22 vs. 1.92) with balanced acceleration. Moreover, against one-step approaches, XYZFlow-B (172M parameters) matches MeanFlow-XL/2+ (676M parameters) in inference speed (0.018s) while achieving superior FID (1.63 vs. 2.20), demonstrating our multidimensional approach's efficacy in trajectory straightening.

Figure 5 presents samples generated by xAR (trained on ImageNet 256 × 256). These results collectively validate XYZFlow's multidimensional conditioning approach in maintaining straighter trajectories and delivering consistent speed-up advantages while preserving sample quality across all model scales and highlight XYZFlow's ability to generate images with exceptional visual quality.

4.3. Ablation Study

Component Importance Analysis. Table 2 presents a systematic evaluation of XYZFlow's core components. Our analysis reveals the distinct contributions of each component through controlled ablations: **(1) Full History Guidance** emerges as the most critical component. Removing full history guidance ($\mathcal{T}_{<p}$) causes FID to degrade by approximately 0.5 across all model sizes (e.g., 2.02→3.51 for Base), demonstrating that inter-patch trajectory conditioning is essential for maintaining generation quality. This validates our hypothesis that complete trajectory information provides richer contextual signals than final patch content alone. **(2) Shortcut Prediction** shows an interesting dual characteristic: while having minimal impact on final quality (FID differences < 0.03), it provides substantial acceleration benefits. The "- Shortcut" variant maintains similar FID scores but requires 20 total steps compared to

Method	Params	Steps	FID↓	IS↑	Pre↑	Rec↑	Total
Teacher-Base	172M	50	1.72	280.4	0.82	0.59	200
Distilled-Base	172M	5	3.03	225.3	0.78	0.55	20
+ Local History	172M	5→2	2.25↓0.78	249.8↑24.5	0.78	0.54↓0.01	14
- Full History	172M	5→2	3.51↑0.48	219.9↓5.4	0.77↓0.01	0.52↓0.03	14
- Shortcut	172M	5	2.05↓0.98	258.5↑33.2	0.80↑0.02	0.58↑0.03	20
XYZFlow-B	172M	5→2	2.02↓1.01	261.1↑35.8	0.80↑0.02	0.58↑0.03	14
XYZFlow-B (GAN)	172M	5→2	**1.63**↓1.40	**268.5**↑43.2	**0.81**↑0.03	**0.62**↑0.07	14
Teacher-Large	608M	50	1.28	292.5	0.82	0.62	200
Distilled-Large	608M	5	2.85	235.1	0.79	0.57	20
+ Local History	608M	5→2	2.02↓0.83	254.3↑19.2	0.79	0.57	14
- Full History	608M	5→2	3.35↑0.50	229.3↓5.8	0.78↓0.01	0.54↓0.03	14
- Shortcut	608M	5	1.82↓1.03	263.8↑28.7	0.81↑0.02	0.61↑0.04	20
XYZFlow-L	608M	5→2	1.79↓1.06	265.2↑30.1	0.81↑0.02	0.61↑0.04	14
XYZFlow-L (GAN)	608M	5→2	**1.25**↓1.60	**295.8**↑60.7	**0.83**↑0.04	**0.63**↑0.06	14
Teacher-Huge	1.1B	50	1.24	301.6	0.83	0.64	200
Distilled-Huge	1.1B	5	2.75	240.8	0.80	0.59	20
+ Local History	1.1B	5→2	1.96↓0.79	259.1↑18.3	0.80	0.57↓0.02	14
- Full History	1.1B	5→2	3.25↑0.50	234.6↓6.2	0.79↓0.01	0.56↓0.03	14
- Shortcut	1.1B	5	1.76↓0.99	268.2↑27.4	0.82↑0.02	0.61↑0.02	20
XYZFlow-H	1.1B	5→2	1.73↓1.02	271.5↑30.7	0.82↑0.02	0.62↑0.03	14
XYZFlow-H (GAN)	1.1B	5→2	**1.22**↓1.53	**304.2**↑63.4	**0.84**↑0.04	**0.64**↑0.05	14

Table 2. Ablation study of XYZFlow components. FID↓ is lower-better; IS↑, Pre↑, Rec↑ are higher-better. Total Steps↓ represents the cumulative inference steps. Colored numbers indicate performance change relative to baseline (Distilled) models.In the table, '+' and '-' denote the baseline model with and without the corresponding component, respectively.

Schedule $T(p)$	FID↓	IS↑	Pre↑	Rec↑	Total Steps↓
Teacher (50 steps)	1.72	280.4	0.82	0.59	200
5→4→3→2 (Ours)	<u>1.63</u>↓0.09	<u>268.5</u>↓11.9	**0.81**↓0.01	**0.62**↑0.03	14↓186
8→4→2→1 (Uniform)	1.75↑0.03	255.2↓25.2	0.77↓0.05	0.57↓0.02	15↓185
4→4→4→4	1.84↑0.12	248.9↓31.5	0.74↓0.08	0.55↓0.04	16↓184
4→3→2→1	1.88↑0.16	245.3↓35.1	0.73↓0.09	0.54↓0.05	10↓190
8→8→8→8	**1.61**↓0.11	**269.0**↓11.4	**0.81**↓0.01	**0.62**↑0.03	32↓168
5→5→5→5 (Constant)	<u>1.63</u>↓0.09	267.9↓12.5	**0.81**↓0.01	<u>0.61</u>↑0.02	20↓180
5→4→4→2	1.64↓0.08	266.2↓14.2	0.80↓0.02	0.60↑0.01	15↓185
5→2→2→2 (Aggressive)	1.70↓0.02	258.6↓21.8	0.78↓0.04	0.58↓0.01	11↓189

Table 3. Ablation study of Next Shortcut Prediction strategies on Base Model (172M). FID↓ is lower-better; IS↑, Pre↑, Rec↑ are higher-better. Total Steps↓ represents the cumulative inference steps. Colored numbers indicate performance change relative to baseline (50 steps). **Bold** indicates best performance, <u>underline</u> indicates second best.

XYZFlow's 14 steps. This confirms that shortcut prediction primarily enhances efficiency rather than quality, aligning with its design purpose of leveraging straightened paths for faster convergence. **(3) Local History Conditioning** ($\mathcal{H}_t^p$) contributes moderately to performance, with its removal causing FID degradation of approximately 0.2-0.3. This suggests that while intra-patch temporal conditioning provides useful stabilization, the spatial conditioning across patches plays a more significant role in the overall framework. **(4) Adversarial component** consistently improves all metrics across model sizes, this demonstrates that XYZFlow's straightened paths provide a favorable foundation for adversarial training on real data, enabling the student to potentially exceed the teacher's capabilities.

Analysis of Next Shortcut Prediction Strategy Our ablation study of shortcut prediction strategies, summarized in Table 3, yields four key insights that validate our design choices: (1) **Gradual reduction achieves optimal efficiency-quality trade-off**. Our proposed 5→4→3→2 strategy achieves nearly identical quality to the constant 5-step approach (FID 1.63 vs. 1.63) but with 30% fewer total steps (14 vs. 20), demonstrating that sampling can be accelerated more aggressively in later stages without compromising quality. (2) **Initial step configuration is critical**. The comparable performance of constant strategies (5→5→5→5 vs. 4→4→4→4) highlights that an initial step of T(1)=5—a divisor of the teacher's 50-step trajectory—provides the optimal starting point for effective distillation. (3) **Aggressive reduction harms diversity**. The 5→2→2→2 strategy shows degraded recall (0.58 vs. 0.62), confirming that overly aggressive step reduction compromises sample diversity, while our gradual approach better preserves solution space coverage. (4) **Computational cost must be balanced**. Although the 8→8→8→8 strategy achieves the lowest FID (1.62), it requires 32 total steps—over twice our method's cost—validating our focus on optimal efficiency-quality trade-offs rather than pure quality maximization.

Theoretical Validation The results confirm the progressive constraint strengthening phenomenon theorized in Section 3. The progressive step reduction strategy effectively balances trajectory completeness and computational efficiency, showing that accumulated contextual constraints enable fewer-step generation for later patches. For more theoretical analysis, please refer to our supplementary materials

5. Concluding Remarks

In this work, we challenge the paradigm of extensive scaling for generative modeling by proposing a novel alternative: intensive scaling through enhanced probability flow expressivity. We introduce the ***XYZFlow*** framework, which scales probability flows along orthogonal temporal and spatial dimensions using historical state conditioning and Next Shortcut Prediction. This creates high-dimensional coordinate systems that uniquely determine data trajectories. Theoretically, increasing conditional information reduces the variance of the reverse process, yielding straighter paths better suited for few-step generation. Looking ahead, scaling the *dimensionality of constraints*, rather than merely model size or distillation steps, provides a principled path toward efficient, high-fidelity generation. This work opens new research avenues in structured conditionalization and flow design as a new way to scale generative models. Advanced distillation methods like meanflow can also work together with our method to enhance generation quality.

References

[1] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3

[2] Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. Flow map matching. *arXiv preprint arXiv:2406.07507*, 2024. 2, 3

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 6

[5] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 3

[6] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025. 2, 3

[7] Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. *Neural Information Processing Systems (NeurIPS)*, 36, 2024a. 2, 3

[8] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024b. 2, 3

[9] Tiankai Hang, Jianmin Bao, Fangyun Wei, and Dong Chen. Fast autoregressive models for continuous latent generation. *arXiv preprint arXiv:2504.18391*, 2025. 2, 3

[10] 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. *Advances in neural information processing systems*, 30, 2017. 6

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 3

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

[13] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Neural Information Processing Systems (NeurIPS)*, 2022. 2

[14] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022. 3

[15] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 2, 3, 7

[16] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3

[17] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 3

[18] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations (ICLR)*, 2025. 2, 3

[19] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 2

[20] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 2

[21] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Neural Information Processing Systems (NeurIPS)*, 2024. 3

[22] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 3

[23] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. *arXiv preprint arXiv:2412.15205*, 2024. 2, 3, 7

[24] Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025. 3, 6, 7

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[26] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. 2, 3

[27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6

[28] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3

[29] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015. 1, 3

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[31] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 3

[32] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Neural Information Processing Systems (NeurIPS)*, 2019. 1, 3

[33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[34] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023. 2, 3

[35] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023. 2

[36] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2025. 7

[37] Zili Wang, Robert Zhang, Kun Ding, Qi Yang, Fei Li, and Shiming Xiang. Continuous speculative decoding for autoregressive image generation. *arXiv preprint arXiv:2411.11925*, 2024. 2, 3

[38] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Zhonghua Wu, Qingyi Tao, Wentao Liu, Wei Li, and Chen Change Loy. Harmonizing visual representations for unified multimodal understanding and generation. *arXiv preprint arXiv:2503.21979*, 2025. 3

[39] Feihong Yan, Qingyan Wei, Jiayi Tang, Jiajun Li, Yulin Wang, Xuming Hu, Huiqi Li, and Linfeng Zhang. Lazymar: Accelerating masked autoregressive models via feature caching. *arXiv preprint arXiv:2503.12450*, 2025. 2, 3

[40] Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024. 2, 3

[41] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[42] Qinyu Zhao, Jaskirat Singh, Ming Xu, Akshay Asthana, Stephen Gould, and Liang Zheng. Disa: Diffusion step annealing in autoregressive image generation. *arXiv preprint arXiv:2505.20297*, 2025. 3

[43] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025. 3

.

In Section A, we provide more experimental details, while in Section B, we present theoretical explanations for some key propositions mentioned in our paper.

A. Experiment Results

Student Model Training Configuration We adhere to the teacher configuration for student training, with the exception of gradient clipping and batch size. The training configuration for the 5-step student model using regression loss is as follows: a learning rate of 10^{-4}, weight decay of 0.0, gradient clipping of 1.0, batch size of 64, 300k training iterations, and an EMA decay rate of 0.9999. The student model is initialized with the teacher's weights. Training is performed on 8 NVIDIA H100 GPUs and requires approximately 2 days to complete 300k iterations. According to our convergence analysis, the FID metric exhibits stable convergence within the first 100k iterations (equivalent to roughly 16 hours of training). The same configuration is applied to the baseline (step distillation) as well.

Discriminator Loss Configuration When incorporating an additional discriminator loss, we use the teacher network as a feature extractor and train only the discriminator heads attached to the features extracted from each transformer block. The discriminator heads predict logits on a per-token basis. We employ hinge loss and adopt the discriminator head architecture proposed in the same work. The discriminator is trained using the student model's final prediction and real data. It is trained with a learning rate of 1×10^{-3} and no weight decay. Adaptive balancing is applied between the regression loss and the discriminator loss. A batch size of 48 is used for both the student model and the discriminator.

Adversarial Fine-tuning Procedure By adding the discriminator loss and further fine-tuning a student model that was pre-trained with regression loss, we observe significant performance gains. The adversarial training component consistently improves all metrics across different model sizes. The fine-tuning process is conducted for 40k iterations, during which both the student generator and the discriminator are jointly optimized with adaptive loss balancing.

Performance Improvement Results As shown in our ablation studies (Table 2), the adversarial fine-tuning yields substantial improvements: the Base model's FID improves from 2.02 to 1.63, the Large model from 1.79 to 1.25, and the Huge model from 1.73 to 1.22. These results demonstrate the effectiveness of incorporating adversarial training into the distillation framework, with consistent enhancements observed across all model scales.

B. Theoretical Proofs of Multi-Dimensional Conditional Enhancement

B.1. Information-Theoretic Foundation of Conditional Modeling

Definition 1 (Conditional Entropy Reduction)**.** *Let target distribution be $p(\mathbf{x})$ and conditioning variable be $\mathbf{c}$. The conditional distribution $p(\mathbf{x}|\mathbf{c})$ has lower entropy than the unconditional distribution $p(\mathbf{x})$.*

Theorem 1 (Conditional Entropy Inequality)**.**

$$H(\mathbf{x}|\mathbf{c}) \leq H(\mathbf{x}) \tag{9}$$

with equality if and only if $\mathbf{x}$ and $\mathbf{c}$ are independent.

Proof. By definition of conditional entropy and non-negativity of mutual information:

$$H(\mathbf{x}|\mathbf{c}) = H(\mathbf{x}) - I(\mathbf{x};\mathbf{c}) \leq H(\mathbf{x}) \tag{10}$$

This means conditional information $\mathbf{c}$ reduces uncertainty in the target distribution. □

B.2. Theoretical Proof of Denoising-Dimension Conditional Enhancement

B.2.1. Autoregressive Trajectory as Condition

In XYZFlow, we use the complete denoising trajectory history as condition:

$$\mathbf{c}_{\text{denoise}} = \{\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots, \mathbf{x}_T\} \tag{11}$$

The generation process becomes:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_{t+1:T}) \tag{12}$$

Proposition 1 (Information Gain from Complete Trajectory)**.** *Using complete denoising trajectory as condition provides more information than using only $\mathbf{x}_t$, reducing generation uncertainty.*

Proof. Define information gain:

$$\Delta I_t = I(\mathbf{x}_{t-1}; \mathbf{x}_{t+1:T}|\mathbf{x}_t) \tag{13}$$

By chain rule:

$$I(\mathbf{x}_{t-1}; \mathbf{x}_{t+1:T}|\mathbf{x}_t) = I(\mathbf{x}_{t-1}; \mathbf{x}_{t+1}|\mathbf{x}_t) + I(\mathbf{x}_{t-1}; \mathbf{x}_{t+2:T}|\mathbf{x}_t, \mathbf{x}_{t+1}) \tag{14}$$

Since diffusion process Markovity is broken, $\mathbf{x}_{t+1:T}$ depends on $\mathbf{x}_{t-1}$ through $\mathbf{x}_0$, thus:

$$\Delta I_t > 0 \quad \text{for non-Markovian processes} \tag{15}$$

Meaning conditional distribution $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_{t+1:T})$ has lower entropy than $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$. □

B.2.2. Trajectory Straightening Proof

Theorem 2 (Trajectory Straightening Theorem)**.** *When using the complete historical denoising trajectory as condition, probability flow paths exhibit significant straightening compared to Markovian processes.*

Proof. Consider the probability flow ODE governing the diffusion process:

$$d\mathbf{x}_t = \mathbf{v}(\mathbf{x}_t, t)dt \tag{16}$$

Define the path straightness metric as the expected deviation from ideal linear interpolation:

$$S(\{\mathbf{x}_t\}_{t=0}^{1}) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[\|(\mathbf{x}_1 - \mathbf{x}_0) - \mathbf{v}_\theta(\mathbf{x}_t, t)\|^2\right] \tag{17}$$

In XYZFlow, we condition on the complete historical trajectory rather than just the current state. Let $\mathcal{H}_t = \{\mathbf{x}_\tau\}_{\tau=0}^{t-\Delta t}$ represent the historical states up to time t. The conditional velocity field becomes:

$$\mathbf{v}_\theta^{\text{conditional}}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_1 - \mathbf{x}_0 \mid \mathbf{x}_t, \mathcal{H}_t] \tag{18}$$

This contrasts with the traditional unconditional estimation:

$$\mathbf{v}_\theta^{\text{unconditional}}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_1 - \mathbf{x}_0 \mid \mathbf{x}_t] \tag{19}$$

By the smoothing property of conditional expectation and the law of total variance:

$$\text{Var}[\mathbf{v}_\theta^{\text{unconditional}}] = \mathbb{E}[\text{Var}[\mathbf{v}_\theta^{\text{unconditional}} \mid \mathcal{H}_t]] + \text{Var}[\mathbb{E}[\mathbf{v}_\theta^{\text{unconditional}} \mid \mathcal{H}_t]] \tag{20}$$

$$\text{Var}[\mathbf{v}_\theta^{\text{conditional}}] = \mathbb{E}[\text{Var}[\mathbf{v}_\theta^{\text{conditional}} \mid \mathcal{H}_t]] \tag{21}$$

Since conditioning on $\mathcal{H}_t$ provides additional information:

$$\text{Var}[\mathbf{v}_\theta^{\text{conditional}} \mid \mathcal{H}_t] \leq \text{Var}[\mathbf{v}_\theta^{\text{unconditional}} \mid \mathcal{H}_t] \tag{22}$$

Therefore, the overall variance satisfies:

$$\text{Var}[\mathbf{v}_\theta^{\text{conditional}}] \leq \text{Var}[\mathbf{v}_\theta^{\text{unconditional}}] \tag{23}$$

The straightness metric S can be decomposed as:

$$S = \mathbb{E}[\|\mathbf{v}_\theta - (\mathbf{x}_1 - \mathbf{x}_0)\|^2] = \text{Var}[\mathbf{v}_\theta] + \|\mathbb{E}[\mathbf{v}_\theta] - (\mathbf{x}_1 - \mathbf{x}_0)\|^2 \tag{24}$$

The bias term $\|\mathbb{E}[\mathbf{v}_\theta] - (\mathbf{x}_1 - \mathbf{x}_0)\|^2$ remains approximately constant under proper training, while the variance term $\text{Var}[\mathbf{v}_\theta]$ decreases with conditioning. Thus:

$$S^{\text{conditional}} \leq S^{\text{unconditional}} \tag{25}$$

This demonstrates that historical trajectory conditioning straightens probability flow paths. □

B.3. Theoretical Proof of Spatial-Dimension Conditional Enhancement

B.3.1. Structural Autoregressive Dependency Modeling

Partition image into patch sequence: $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\}$

When generating i-th patch, use all previous patches as condition:

$$\mathbf{c}_{\text{spatial}} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^{i-1}\} \tag{26}$$

Generation process:

$$p_\theta(\mathbf{x}^i | \mathbf{x}^{1:i-1}) \tag{27}$$

Proposition 2 (Structural Constraint Enhancement)**.** *Spatial conditions provide structural constraints, reducing generation distribution uncertainty.*

Proof. Consider mutual information between patches:

$$I(\mathbf{x}^i; \mathbf{x}^{1:i-1}) = H(\mathbf{x}^i) - H(\mathbf{x}^i | \mathbf{x}^{1:i-1}) \tag{28}$$

Due to spatial correlations in natural images:

$$I(\mathbf{x}^i; \mathbf{x}^{1:i-1}) > 0 \Rightarrow H(\mathbf{x}^i | \mathbf{x}^{1:i-1}) < H(\mathbf{x}^i) \tag{29}$$

Conditional entropy reduction means more certain generation distribution. □

B.3.2. Variance Reduction Effect

Theorem 3 (Spatial Conditional Variance Reduction)**.** *Spatial autoregressive conditions reduce generation process variance.*

Proof. Unconditional variance:

$$\sigma^2_{\text{uncond}} = \text{Var}[\mathbf{x}^i] \tag{30}$$

Conditional variance:

$$\sigma^2_{\text{cond}} = \mathbb{E}[\text{Var}[\mathbf{x}^i | \mathbf{x}^{1:i-1}]] \tag{31}$$

By variance decomposition:

$$\text{Var}[\mathbf{x}^i] = \mathbb{E}[\text{Var}[\mathbf{x}^i | \mathbf{x}^{1:i-1}]] + \text{Var}[\mathbb{E}[\mathbf{x}^i | \mathbf{x}^{1:i-1}]] \tag{32}$$

Thus:

$$\sigma^2_{\text{cond}} = \sigma^2_{\text{uncond}} - \text{Var}[\mathbb{E}[\mathbf{x}^i | \mathbf{x}^{1:i-1}]] \leq \sigma^2_{\text{uncond}} \tag{33}$$

Equality only when conditional expectation is constant (independent patches). □

B.4. Theoretical Proof of Next-Shortcut Prediction

B.4.1. Mechanism Definition and Formalization

Next-shortcut prediction uses denoising trajectories of preceding patches as condition:

$$\mathbf{c}_{\text{shortcut}} = \{\mathbf{x}_t^j : j = 1, \ldots, i-1, t = 0, \ldots, T\} \tag{34}$$

Generation process:

$$p_\theta(\mathbf{x}^i | \mathbf{c}_{\text{shortcut}}) = p_\theta(\mathbf{x}^i | \mathbf{x}^{1:i-1}, \{\mathbf{x}_t^j\}_{j=1,t=0}^{i-1,T}) \tag{35}$$

B.4.2. Structural Constraint Enhancement

Theorem 4 (Structural Constraint Strengthening)**.** *Next-shortcut prediction imposes strong structural constraints through cross-patch trajectory consistency.*

Proof. Define trajectory consistency metric:

$$C(\mathbf{x}^i, \mathbf{x}^j) = \mathbb{E}\left[\|\mathbf{v}(\mathbf{x}_t^i) - \mathbf{v}(\mathbf{x}_t^j)\|^2\right] \tag{36}$$

With shortcut conditions:

$$\mathbb{E}[C(\mathbf{x}^i, \mathbf{x}^j)|\mathbf{c}_{\text{shortcut}}] \leq \mathbb{E}[C(\mathbf{x}^i, \mathbf{x}^j)] \tag{37}$$

Conditional mutual information is non-negative:

$$I(\mathbf{x}^i; \{\mathbf{x}_t^j\}_{j=1}^{i-1}|\mathbf{x}^{1:i-1}) \geq 0 \tag{38}$$

Thus:

$$H(\mathbf{x}^i|\mathbf{x}^{1:i-1}, \{\mathbf{x}_t^j\}_{j=1}^{i-1}) \leq H(\mathbf{x}^i|\mathbf{x}^{1:i-1}) \tag{39}$$

Conditional entropy reduction means stronger structural constraints. □

B.4.3. Mapping Specificity Enhancement

Theorem 5 (Mapping Specificity Improvement)**.** *Next-shortcut prediction enriches condition space, making noise-to-data mapping more specific.*

Proof. Consider mapping $f_i : \mathcal{Z}^i \to \mathcal{X}^i$. Conditional mapping $f_i^{\text{shortcut}}(\mathbf{z}^i|\mathbf{c}_{\text{shortcut}})$ has richer parameterization.

Mapping specificity measured by distribution kurtosis:

$$\text{Specificity} = \mathbb{E}[(\mathbf{x}^i - \mu)^4]/\sigma^4 \tag{40}$$

With enhanced conditions, distribution becomes more peaked, kurtosis increases.

Conditional Jacobian has better condition number:

$$\kappa(J_{f_i}^{\text{conditional}}) \leq \kappa(J_{f_i}^{\text{unconditional}}) \tag{41}$$

Condition information constrains mapping directions, improving numerical stability. □

B.4.4. Variance Reduction Analysis

Theorem 6 (Shortcut Prediction Variance Reduction)**.** *Next-shortcut prediction reduces generation process variance.*

Proof. Unconditional variance: $\sigma^2_{\text{uncond}} = \text{Var}[\mathbf{x}^i]$

Spatial-only variance: $\sigma^2_{\text{spatial}} = \mathbb{E}[\text{Var}[\mathbf{x}^i|\mathbf{x}^{1:i-1}]]$

Shortcut variance: $\sigma^2_{\text{shortcut}} = \mathbb{E}[\text{Var}[\mathbf{x}^i|\mathbf{x}^{1:i-1}, \{\mathbf{x}_t^j\}_{j=1}^{i-1}]]$

By variance decomposition:

$$\sigma^2_{\text{shortcut}} = \sigma^2_{\text{spatial}} - \text{Var}[\mathbb{E}[\mathbf{x}^i|\mathbf{x}^{1:i-1}, \{\mathbf{x}_t^j\}_{j=1}^{i-1}]|\mathbf{x}^{1:i-1}] \leq \sigma^2_{\text{spatial}} \tag{42}$$

Variance reduction directly improves sampling efficiency. □

B.5. Unified Perspective: Path Straightening through Conditional Enhancement

B.5.1. Mathematical Equivalence Proof

Theorem 7 (Variance Reduction-Path Straightening Equivalence)**.** *For diffusion model probability flow paths, conditional variance reduction is mathematically equivalent to path straightening.*

Proof. Straightness metric:

$$S = \mathbb{E}_t[\|(\mathbf{x}_1 - \mathbf{x}_0) - \mathbf{v}_\theta(\mathbf{x}_t, t)\|^2] \tag{43}$$

Velocity field variance:

$$\text{Var}[\mathbf{v}_\theta] = \mathbb{E}[\|\mathbf{v}_\theta - \mathbb{E}[\mathbf{v}_\theta]\|^2] \tag{44}$$

For straight paths: $\mathbf{v}_\theta \equiv \mathbf{x}_1 - \mathbf{x}_0$, thus:

$$S = 0 \Leftrightarrow \text{Var}[\mathbf{v}_\theta] = 0 \tag{45}$$

Therefore, variance reduction directly implies path straightening. □

B.5.2. Dual Mechanisms of Path Straightening

Theorem 8 (Spatial Condition Path Constraint)**.** *Spatial autoregressive conditions $\mathbf{x}^{1:i-1}$ straighten paths through spatial consistency constraints.*

Proof. For adjacent patch paths $\{\mathbf{x}_t^i\}$ and $\{\mathbf{x}_t^{i-1}\}$:

Unconditionally: $\text{Cov}(\mathbf{v}_t^i, \mathbf{v}_t^{i-1}) \approx 0$

Conditionally: $\mathbf{v}_t^i = f(\mathbf{v}_t^{i-1}) + \text{small error}$

Spatial consistency enforces path coordination, reducing spatial bending. □

Theorem 9 (Shortcut Prediction Trajectory Alignment)**.** *Next-shortcut prediction further constrains path direction consistency through cross-patch trajectory alignment.*

Proof. Using preceding patches' trajectories $\{\mathbf{x}_t^j\}_{j=1}^{i-1}$ as condition enables trajectory interpolation.

Minimize trajectory consistency loss:

$$\mathcal{L}_{\text{trajectory}} = \sum_{j=1}^{i-1} \|\mathbf{v}_t^i - \mathbf{v}_t^j\|^2 \tag{46}$$

Minimization forces new patch paths to align with existing ones, naturally straightening paths. □

B.5.3. Synergistic Straightening Effects

Theorem 10 (Orthogonal Constraint Synergy)**.** *Spatial conditions and shortcut predictions have approximately orthogonal constraint spaces, producing multiplicative path optimization.*

Proof. Define constraint operators:

- P_S: Spatial condition projection operator

- P_T: Shortcut prediction projection operator

Path optimization:

$$\min \|(I - P_S P_T)\mathbf{v}\|^2 \tag{47}$$

With approximate orthogonality, joint optimization outperforms individual optimizations. □

Theorem 11 (Unified Path Evolution)**.**
- ***Unconditional paths***: *Random walks in high-dimensional space*
- ***Spatial conditions only***: *Spatial dimension straightened, temporal dimension still curved*
- ***Spatial + Shortcut conditions***: *Spatio-temporal simultaneous straightening*

B.6. Multi-Dimensional Shortcut Scaling Theory

B.6.1. Coordinate System Enrichment

Definition 2 (Probability Path Coordinate System)**.** *The coordinate system of probability path $\{\mathbf{x}_t\}$ is defined by the space spanned by condition variables.*

Standard diffusion: $\{\mathbf{x}_t\}$
XYZFlow: $\{\mathbf{x}_t^i, \mathbf{x}_{t+1:T}^i, \mathbf{x}^{1:i-1}, \{\mathbf{x}_t^j\}_{j=1}^{i-1}\}$

Theorem 12 (Coordinate System Enrichment)**.** *Multi-dimensional shortcut scaling enriches probability path coordinate system, improving mapping specificity.*

Proof. Consider mapping $f : \mathcal{Z} \to \mathcal{X}$. Conditional mapping $f(\mathbf{z}_t^i | \mathbf{c}_{\text{denoise}}, \mathbf{c}_{\text{spatial}}, \mathbf{c}_{\text{shortcut}})$ has richer parameterization.

Conditional Jacobian has better condition number:

$$\kappa(J_f^{\text{conditional}}) \leq \kappa(J_f^{\text{unconditional}}) \tag{48}$$

Condition information provides additional constraints, stabilizing mapping process. □

B.6.2. Cumulative Variance Reduction

Theorem 13 (Multiplicative Variance Reduction)**.** *Denoising and spatial dimension conditional enhancements have synergistic variance reduction effects.*

Proof. Consider joint conditional distribution:

$$p(\mathbf{x}_{t-1}^i | \mathbf{x}_t^i, \mathbf{x}_{t+1:T}^i, \mathbf{x}^{1:i-1}, \{\mathbf{x}_t^j\}_{j=1}^{i-1}) \tag{49}$$

By iterative variance decomposition:

$$\text{Var}[\mathbf{x}_{t-1}^i] = \mathbb{E}[\text{Var}[\mathbf{x}_{t-1}^i | \mathbf{x}_t^i]] + \text{Var}[\mathbb{E}[\mathbf{x}_{t-1}^i | \mathbf{x}_t^i]] \tag{50}$$

$$\text{Var}[\mathbf{x}_{t-1}^i | \mathbf{x}_t^i] = \mathbb{E}[\text{Var}[\mathbf{x}_{t-1}^i | \mathbf{x}_t^i, \mathbf{x}_{t+1:T}^i] | \mathbf{x}_t^i] + \cdots \tag{51}$$

Each conditioning step further reduces conditional variance, producing cumulative reduction. □

B.7. Corollaries and Implications

B.7.1. Sampling Efficiency Improvement

Corollary 1 (Sampling Efficiency Enhancement)**.** *Variance reduction enables fewer sampling steps for same generation quality.*

Proof. By numerical integration error analysis:

$$\text{Error} \propto \sum \Delta t^2 \cdot \sigma^2 \tag{52}$$

Variance reduction $\sigma^2 \downarrow$ allows larger steps $\Delta t \uparrow$ or fewer steps. □

B.7.2. Mapping Determinism Enhancement

Corollary 2 (Mapping Determinism Improvement)**.** *Rich conditional information makes probability flow paths approach deterministic mapping.*

Proof. With sufficiently rich conditions:

$$\lim_{|\mathbf{c}| \to \infty} H(\mathbf{x}_{t-1} | \mathbf{c}) = 0 \tag{53}$$

Noise-to-data mapping becomes almost deterministic, reducing sampling variability. □