Appendix Outline

The appendix is organized as follows:

- Appendix A provides the proof of Theorem 2.1.
- Appendix B presents supplementary analysis, including model prediction trends (B.1), latent space comparisons (B.2), quantification of memorization behaviors (B.3), and timestep shift illustration (B.4).
- Appendix C lists implementation details.
- Appendix D explores additional training-time practices, such as oracle-supervised training (D.1) and stage-aware training timestep sampling (D.2).
- Appendix E extends our findings to Flux.1[dev] [23], including timestep-shifted sampling (E.1) and analysis of two-stage behaviors in larger diffusion models (E.2).
- Appendix F presents additional qualitative results.

A. Proof of Theorem 2.1

The Flow Matching (FM) objective (Eq. 8) is given by:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,\, p_t(x_t)} ||v_t(x_t;\theta) - u_t(x_t)||^2. \tag{8}$$

The marginal velocity field $u_t(x_t)$ in Eq. 8 is generally intractable under unknown probability paths. However, we want to show that it admits a closed-form solution under the following conditions: (i) a Gaussian prior distribution, (ii) a finite dataset $\{x_1^{(i)}\}_{i=1}^N$ approximating p_{data}, and (iii) a linear interpolation path as in rectified flow. Formally, our goal is to obtain a closed-form expression for the following conditional expectation:

$$u_t^*(x_t, t) \coloneqq \mathbb{E}_{x_1 \sim p_{\text{data}}} \left[u_t(x_t \mid x_1) \,\middle|\, x_t \right]. \tag{9}$$

Under linear flows (i.e., probability paths constructed via linear interpolation), we have:

$$x_t = \alpha_t x_1 + \sigma_t x_0 \quad \Longrightarrow \quad x_0 = \frac{x_t - \alpha_t x_1}{\sigma_t}. \tag{10}$$

Recall that the conditional velocity in CFM is given by:

$$u_t(x_t \mid x_1) = \dot{\alpha}_t x_1 + \dot{\sigma}_t x_0. \tag{11}$$

Substituting Eq. 10 into Eq. 11 writes $u_t(x_t \mid x_1)$ as a function of x_t and x_1:

$$u_t(x_t \mid x_1) \;=\; \left(\dot{\alpha}_t - \frac{\alpha_t \dot{\sigma}_t}{\sigma_t}\right) x_1 \;+\; \frac{\dot{\sigma}_t}{\sigma_t}\, x_t. \tag{12}$$

Taking the conditional expectation given x_t yields:

$$\begin{aligned} u_t^*(x_t, t) \;&\coloneqq\; \mathbb{E}[u_t(x_t \mid x_1) \mid x_t] && (13) \\ &=\; \left(\dot{\alpha}_t - \frac{\alpha_t \dot{\sigma}_t}{\sigma_t}\right) \mathbb{E}[x_1 \mid x_t] \;+\; \frac{\dot{\sigma}_t}{\sigma_t}\, x_t, && (14) \end{aligned}$$

Given a finite dataset $\{x_1^{(i)}\}_{i=1}^N$, we essentially approximate the true data distribution $p_{\text{data}}(x_1)$ via an empirical mixture:

$$p_{\text{data}}(x_1) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(x_1 - x_1^{(i)}), \tag{15}$$

where $\delta(\cdot)$ is the Dirac delta function with $\delta(0) = \infty$ and zero elsewhere. Accordingly, the empirical probability path marginal $\tilde{p}_t(x_t)$ is given by a Gaussian mixture:

$$\tilde{p}_t(x_t) \;=\; \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}\big(x_t;\, \alpha_t x_1^{(i)},\, \sigma_t^2 I\big). \tag{16}$$

By Bayes' rule, the posterior $p(x_1^{(i)} \mid x_t)$ is proportional to $p(x_t \mid x_1^{(i)})$ up to a common normalizing factor that ensures the probabilities sum to one. Since all mixture components share the same Gaussian covariance $\sigma_t^2 I$ and uniform prior weight $1/N$, their normalization constants cancel out when computing the posterior weights. Let $\gamma_i(x_t, t)$ denote the resulting normalized weighting function that reflects the relative contribution of each data sample $x_1^{(i)}$ to the current point x_t, we have:

$$\gamma_i(x_t, t) \;=\; \frac{\exp\!\big(-\frac{\|x_t - \alpha_t x_1^{(i)}\|^2}{2\sigma_t^2}\big)}{\sum_{j=1}^{N} \exp\!\big(-\frac{\|x_t - \alpha_t x_1^{(j)}\|^2}{2\sigma_t^2}\big)}. \tag{17}$$

Hence, the posterior mean is given by:

$$\mathbb{E}[x_1 \mid x_t] \;=\; \sum_{i=1}^{N} \gamma_i(x_t, t)\, x_1^{(i)}. \tag{18}$$

This is also known as the Nadaraya-Watson estimator [30, 46]. Combining Eq. 14 and 18, we reach the closed form:

$$u_t^*(x_t, t) \;=\; A_t \sum_{i=1}^{N} \gamma_i(x_t, t)\, x_1^{(i)} \;+\; B_t\, x_t, \tag{19}$$

where $A_t = \dot{\alpha}_t - \frac{\alpha_t \dot{\sigma}_t}{\sigma_t}$, $B_t = \frac{\dot{\sigma}_t}{\sigma_t}$. We refer to this closed-form expression of the marginal velocity field under the linear probability path construction as the *oracle velocity field*. Moreover, the oracle velocity field can also be evaluated conditionally; for instance, under a class-conditional generation setting, it can be computed within each class-specific subset, denoted as $u_t^*(x_t, t \mid y)$. □

B. Supplementary Analysis

B.1. Model Prediction Trends

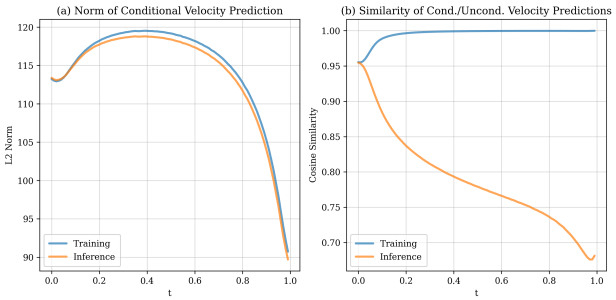

Figure 9. Analysis of model prediction trends. (a) Norm of velocity predictions peaks around t=0.4, coinciding with the timesteps where CFG most effectively enhances sample fidelity (Tab. 3). (b) Cosine similarity between conditional and unconditional predictions. Along training trajectories, the two remain aligned; during inference, x_t deviates from the supervised region, leading to diverged predictions and exhibiting generalization behaviors.

B.2. Latent Space Comparisons

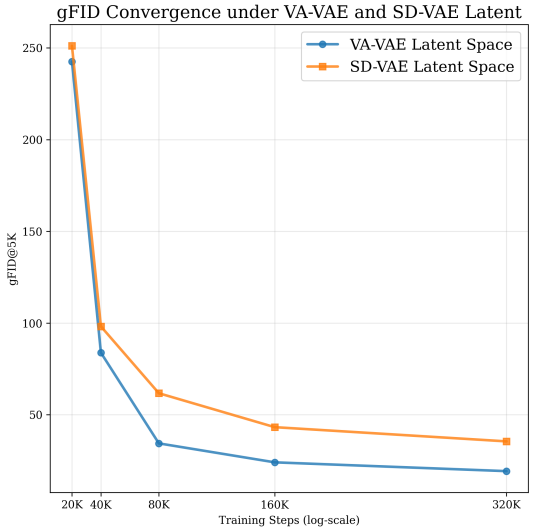

Figure 10. Convergence of gFID@5K when training rectified flow models under different latent spaces. We use LightningDiT-B/1 for VA-VAE [47] and LightningDiT-B/2 for SD-VAE [36] to align the training resolution to 16^2. The training in the VA-VAE latent space converges faster, indicating a better latent space structure.

B.3. Quantification of Memorization Behaviors

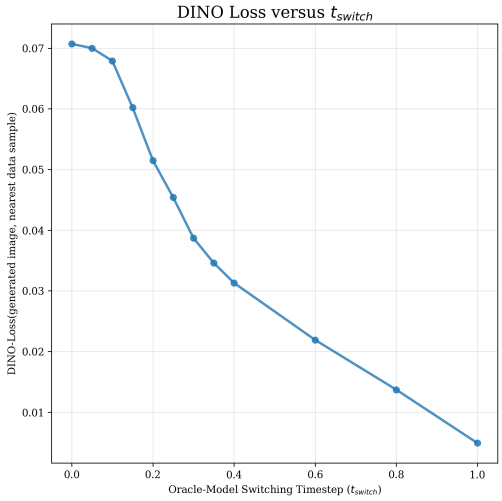

Figure 11. Quantitative results for oracle-model mixed generation. We report the DINO [33] loss (MSE of DINO self-attention maps, reflecting structural dissimilarity) between the generated images and the nearest training sample, measured across different switching timesteps (t_{switch}). Overall, we observe: (1) a sharp decline emerges after $t_{\text{switch}} \approx 0.1$ (i.e., the shift of training target); (2) when the loss falls below roughly 0.05, the generated layouts become closely aligned with those of the training samples (Fig. 16).

B.4. Timestep Shift Illustration

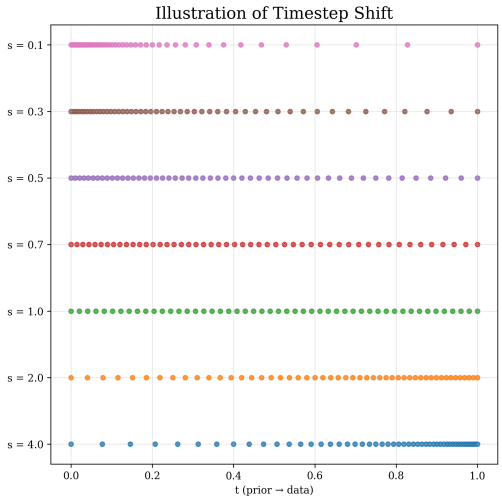

Figure 12. Illustration of timestep shift mapping $t_m = \frac{s\, t_n}{1+(s-1)t_n}$, where s is a shift factor and t_n is the uniform sampling schedule. Intuitively, with $s < 1$, we allocate more steps to the early (navigation) timesteps, while $s > 1$ biases the schedule toward later (refinement) timesteps. The best gFID is achieved with $s = 0.5$.

C. Implementation Details

architecture			
LightningDiT [47] variants	Small	Base	XL
depth	12	12	28
hidden dim	384	768	1152
heads	6	12	16
image size	256		
patch size	1 (VA-VAE), 2 (SD-VAE)		
latent size	16×16		
training			
epochs	$\{100, 800\}$		
optimizer	AdamW [29] ($\beta_1, \beta_2 = 0.9, 0.995$)		
batch size	512		
learning rate	1e-4		
learning rate schedule	constant		
weight decay	0		
max gradient norm	1.0		
ema decay	0.9999		
time sampler	Uniform[0, 1]		
class token drop (for CFG)	0.1		
sampling			
ODE solver	Euler		
ODE steps	50		
time steps	uniform / stage-wise uniform / shifted		
CFG [16] scale	$\{1.0, 2.5\}$		

Table 4. Implementation details.

D. Additional Training-Time Practices

D.1. Oracle-Supervised Training

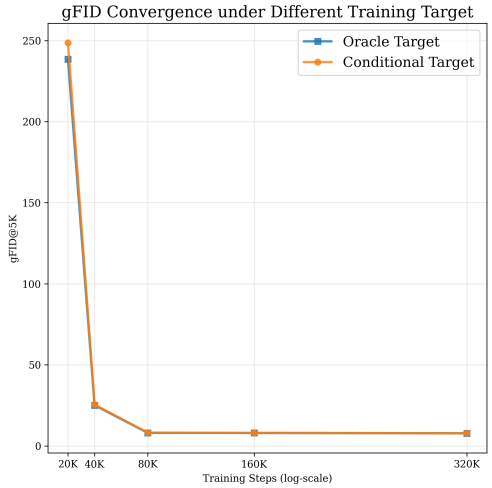

Figure 13. Convergence of gFID@5K when training rectified flow models with oracle/conditional target on a 100-class ImageNet [8] subset. Oracle supervision offers a slight advantage in early training (due to more accurate guidance on near-prior timesteps), while the later training dynamics largely overlap (as the oracle target collapses to the conditional target on most timesteps).

D.2. Stage-aware Training Timestep Sampling

Fraction of Stage 1 Training Steps	5%	10% (uniform)	20%
ImageNet gFID@5K ↓	**23.69**	24.28	25.34

Table 5. gFID convergence with stage-aware timestep sampling. We set the stage shift at $t = 0.1$ and replace the uniform sampler with a piecewise-uniform distribution that controls the fraction of training steps allocated to each stage. Under uniform sampling, 10% of timesteps fall in Stage 1 by default. Allocating more training steps to Stage 2 accelerates convergence, consistent with the observation that the benefits of additional capacity and training compute are primarily reflected in the refinement stage (Sec. 4.3). Evaluated on LightningDiT-B at 160K steps, no CFG.

E. Findings on Flux.1[dev]

E.1. Timestep-shifted Sampling on Flux.1[dev]

Based on the oracle velocity, higher-resolution models like Flux/SD3 exhibit a faster concentration of the top-1 posterior, leading to a shorter navigation interval. Hence, these models become more sensitive to timestep allocation (during both training and inference) due to the condensed navigation stage. On a Flux.1[dev] model, *a properly timestep-shifted sampling schedule (i.e., allocating slightly more navigation steps) improves both content fidelity and visual aesthetics* (Tab. 6), while an inappropriate shift can lead to corrupted content (Fig. 14).

Timestep Shift s	1.0	0.7	0.5	0.3	0.2	0.1
CLIP-Score ↑	22.02	27.69	28.84	**29.16**	27.26	21.10
LAION-Aesthetics ↑	4.96	6.37	6.90	**7.22**	7.11	4.39

Table 6. Timestep-shifted sampling in Flux.1[dev]: slightly more navigation steps (i.e., smaller s) improves image quality.

Figure 14. Flux.1[dev] generations under varying timestep shifts.

E.2. Analysis of Two-Stage Behaviors in Flux.1[dev]

Figure 15. Qualitative illustration of two-stage behavior in `Flux.1[dev]` [23]. Specifically, we first generate a reference latent z_{gt} via text-to-image sampling. Then, we re-noise it by interpolating with Gaussian noise at a chosen t_{resume} and resume sampling. Owing to Flux's higher-dimensional latent space, the stage transition appears earlier than models trained on 256^2 ImageNet data, and the model can reliably recover nearly identical images even after $\sim 90\%$ noise corruption. We also note that Flux employs a non-uniform, resolution-aware timestep schedule and a different time convention; all t_{resume} values shown here are converted to our convention for consistency.

F. Additional Qualitative Results

Figure 16. Mixed sampling results with switch point t_{switch}. Oracle u_t^* is used before t_{switch} and LightningDiT-XL/1 [47] afterward. Overall, early switching yields diverse novel outputs (generalization), while late switching reproduces training samples (memorization). Despite minor variations across sampling trajectories, the empirical stage transition (i.e., reverting to training-like layouts) emerges around $t = 0.2$, slightly lagging behind the shift in the training target. Zoom in for the best view. Better view with color.

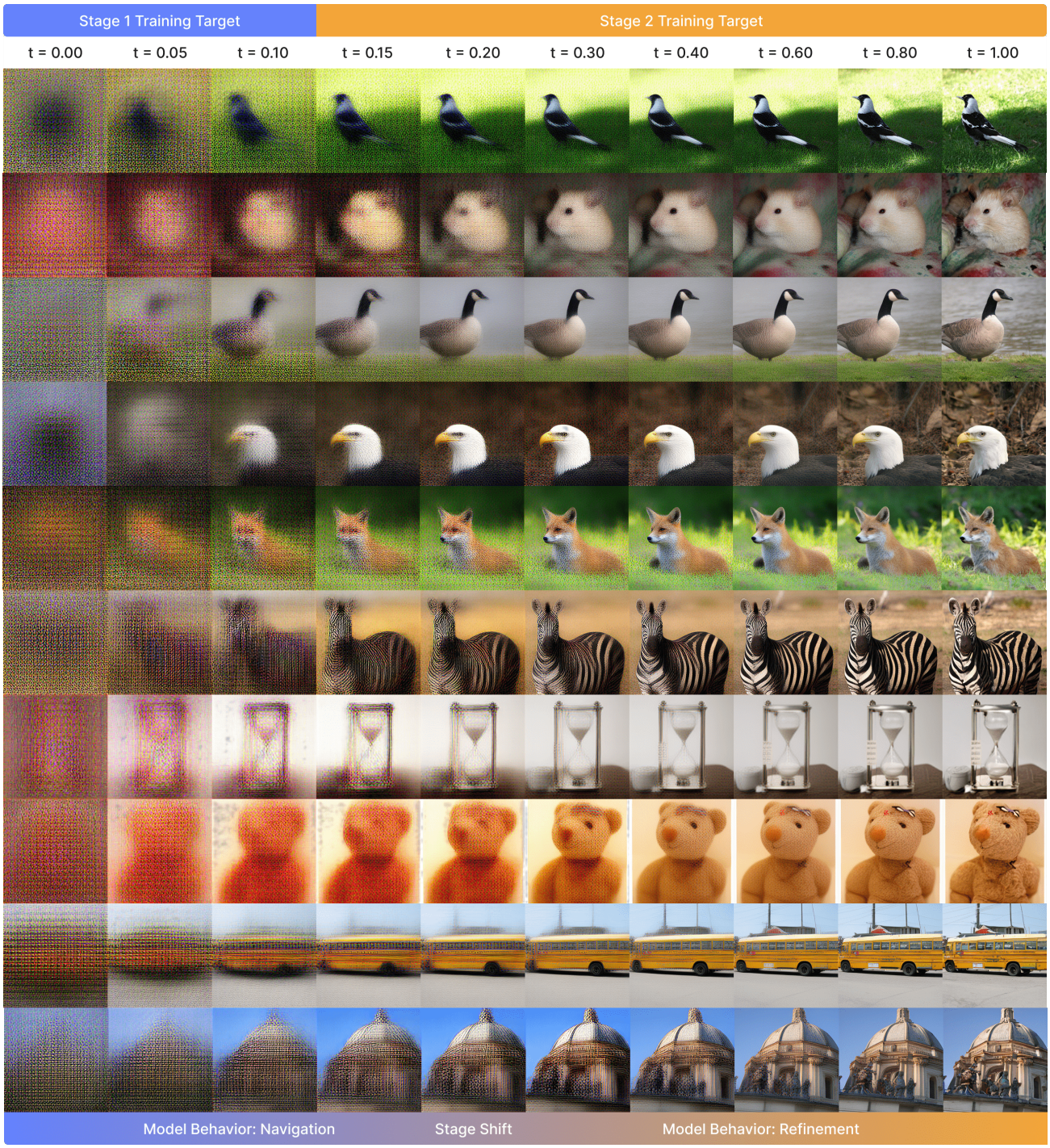

Figure 17. Intermediate predictions of a LightningDiT-XL/1 [47] model across timesteps. Overall, early stages primarily navigate global layout via smoothed predictions, while later stages refine fine-grained details. Zoom in for the best view. Better view with color.