# AMORTIZED MOMENT MATCHING FOR VISUAL GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We propose amortized moment matching, utilizing neural networks to learn data moments as distributional training signals. By casting diffusion denoisers through polynomial projections, we establish a general framework for moment amortization, revealing that an $n$-th degree projection explicitly identifies data moments up to order $n+1$. Derived from the tractable affine case, we instantiate the Amortized Fréchet Distance (AMFD) loss. Unlike FD-loss which relies on explicit marginal moment calculations, AMFD is able to dynamically learn conditional moments via an alternating, matrix-free optimization pipeline that effortlessly scales to high-dimensional data. When operating on representation features, AMFD serves as a powerful post-training objective; empirically, its neural formulation yields more robust training dynamics than exact statistical matching, substantially surpassing the FD-loss baseline on the FDr$^6$ metric and achieving superior one-step generation on ImageNet. Furthermore, it unlocks direct exploration within native generative spaces, suggesting that the first two moments can identify target distributions only in spaces with strong semantics. Finally, when scaled to text-to-image generation, the condition-aware nature of AMFD unlocks massive gains in instruction-following capabilities, enabling our one-step models to outperform their multi-step FLUX.2 [klein] 4B teachers on the GenEval benchmark while achieving on-par performance on PickScore.

## 1 INTRODUCTION

Distribution matching is a natural paradigm for generative modeling, aiming to directly align the generated distribution with the target distribution. Classical approaches adopt moment matching and maximum mean discrepancy (MMD) objectives (Sriperumbudur et al., 2010; Gretton et al., 2012; Li et al., 2015; Dziugaite et al., 2015; Mathiasen & Hvilshøj, 2020). However, they struggled to become competitive for visual generation, largely due to challenges in finite-sample estimation, kernel selection, and optimization stability (Li et al., 2017). In contrast, dominant paradigms such as diffusion and flow matching (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Liu et al., 2022a; Lipman et al., 2022), autoregressive models (Van Den Oord et al., 2016; Razavi et al., 2019; Esser et al., 2021; Sun et al., 2024), GANs (Goodfellow et al., 2014; Karras et al., 2019), and normalizing flows (Dinh et al., 2016; Kingma & Dhariwal, 2018; Zhai et al., 2024) circumvent explicit distribution matching by adopting particle-level generative rules, allowing distributional alignment to emerge implicitly.

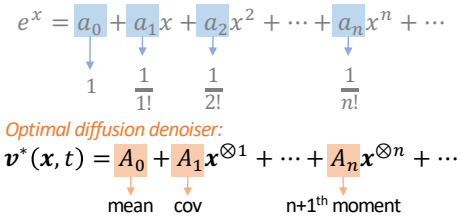


Figure 1: **Diffusion denoiser polynomial projection.** The Maclaurin series expands $e^x$ into scalar polynomial terms; analogously, expanding the optimal diffusion denoiser via tensor powers yields high-dimensional coefficients. We show that these coefficients explicitly map to the underlying distributional moments, where $A_0$ corresponds to the mean, $A_1$ to the covariance, and $A_n$ to the $(n+1)$-th moment.

Recent progress, however, reveals that the historical weakness of direct distribution matching stemmed largely from the underlying data spaces. Once shifted to semantically rich representation spaces, distribution matching emerges as an effective paradigm. For instance, Drifting models (Deng et al., 2026) utilize kernel forces to evolve generated distributions, while the FD-loss (Yang et al.,



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

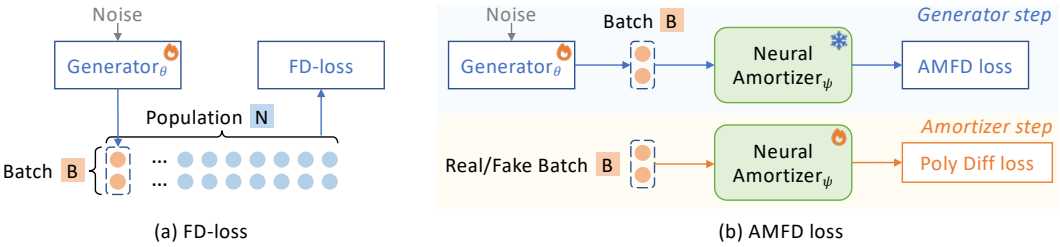


Figure 2: **Comparison between FD-loss and our AMFD loss.** (a) FD-loss explicitly computes marginal distribution statistics over a large population $N$, propagating gradients through mini-batches. (b) Our AMFD loss uses neural amortizers which can leverage neural generalization to naturally encode conditions. These amortizers dynamically maintain the conditional mean and covariance statistics, which the generator directly queries to obtain gradient signals for updating.

2026) directly optimizes mean and covariance matching with a large population. Advancing the perspective of moment matching, we propose *amortized moment matching*. Rather than estimating moments purely from empirical batches or fixed global references, we amortize the statistics themselves: a neural amortizer predicts moment operators as dynamic functions of the conditioning signal and/or the target representation space. Our theoretical motivation stems from diffusion models, where a denoising model learns to approximate the conditional expectation of the clean data or the velocity field given the noisy observation. As illustrated in Figure 1, we analyze this denoiser function through polynomial projections: the affine projection recovers mean and covariance information, while higher-degree projections correspond to amortizing higher-order moments. This formulation offers a generalized route to moment amortization.

While high-order moment tensors grow exponentially in size, we instantiate a tractable and powerful second-order case termed the *Amortized Fréchet Distance* (AMFD) loss. When operating on global representation features, AMFD serves as an effective post-training objective that can enhance generative models and transform multi-step models into one-step generators. Crucially, in contrast to FD-loss (Yang et al., 2026) which computes aggregated marginal moments, AMFD injects the conditioning signal directly into the amortizer. This aligns naturally with modern diffusion models, where denoisers operate as conditional functions of labels or text prompts, enabling dynamic, condition-aware distribution matching. As illustrated in Figure 2(b), AMFD formulates this as a fully deep-learning-native pipeline driven by an alternating optimization between the generator and the neural amortizer. To avoid the prohibitive complexity of explicit covariance matrices, we parameterize the affine function via Jacobian-Vector-Product (JVP) actions. This matrix-free efficiency effortlessly scales to high dimensions, which additionally unlocks the exploration of whether matching merely the first two moments provides sufficient generative signals within native generative spaces (*e.g.*, raw pixels, VAE/RAE latents (Rombach et al., 2022; Zheng et al., 2025)), where FD-loss is computationally intractable. Ultimately, AMFD strategically trades strict empirical exactness for superior conditional generalization and scalability.

We evaluate AMFD in diverse visual generation tasks. First, in ImageNet-256×256 post-training, its neural formulation yields more robust optimization dynamics than the FD-loss baseline, achieving state-of-the-art FDr$^6$ scores of 1.79 on JiT-H/16 and 1.75 on pMF-H/16. Second, exploration within native generative spaces suggests that the first two moments can identify target distributions only in spaces with strong semantics, such as the RAE space. Finally, when scaled to text-to-image generation with pixel-space PixelGen (Ma et al., 2026) and latent-space FLUX.2 [klein] 4B (Black Forest Labs, 2026), the condition-aware nature of AMFD unlocks massive gains in instruction-following capabilities. Notably, our one-step post-trained FLUX models significantly surpass their multi-step and few-step teachers on the GenEval benchmark, while maintaining on-par on PickScore.

Our main contributions are summarized as follows:

- For the first time, we establish a rigorous connection between diffusion denoisers and data moments by proving that an $n$-th degree polynomial projection of a denoiser explicitly identifies moments up to order $n+1$. Grounded in this theory, we introduce *amortized moment matching*, a scalable framework that learns and matches distributional moments via neural amortizers.
- We instantiate the tractable second-order case as the Amortized Fréchet Distance (AMFD) loss. Unlike FD-loss that is restricted to unconditional marginals, AMFD leverages neural generaliza-



Figure 3: One-step text-to-image samples generated by our AMFD post-trained FLUX.2 [klein] 4B.

tion to efficiently match fine-grained conditional moments—a capability crucial for scaling to complex text-to-image (T2I) generation.

- Empirically, AMFD achieves state-of-the-art FDr$^6$ for one-step generation on ImageNet-256×256, reaching 1.79 (JiT-H/16) and 1.75 (pMF-H/16). Furthermore, when extended to text-to-image generation, our one-step FLUX.2 models surpass their multi-step teachers on the GenEval benchmark (0.846 vs. 0.794) while maintaining on-par performance on PickScore.

## 2 Preliminaries

### 2.1 Fréchet Distance

Let $P_r$ and $P_g$ denote the real and generated data distributions over the image space $\mathcal{X}$. Given a frozen representation encoder $\phi : \mathcal{X} \to \mathbb{R}^D$, we aim to compare the push-forward feature distributions $\phi_\# P_r$ and $\phi_\# P_g$. We define their respective first two moments for $i \in \{r, g\}$ as:

$$\mu_i = \mathbb{E}_{X\sim P_i}[\phi(X)], \quad \Sigma_i = \mathrm{Cov}_{X\sim P_i}[\phi(X)], \tag{1}$$

where $X$ denotes the random variable in $\mathcal{X}$. The squared Fréchet distance, which measures the discrepancy between their Gaussian approximations, is defined as:

$$d^2_{\mathrm{FD}}(P_g, P_r; \phi) = \|\mu_g - \mu_r\|_2^2 + \mathrm{Tr}\left(\Sigma_g + \Sigma_r - 2(\Sigma_g^{1/2}\Sigma_r\Sigma_g^{1/2})^{1/2}\right). \tag{2}$$

When $\phi$ is Inception-v3 (Szegedy et al., 2016), this recovers the widely used Fréchet Inception Distance (FID) (Heusel et al., 2017). Recently, the FD-loss (Yang et al., 2026) proposed directly optimizing Eq. 2 across multiple representation spaces to train generative models.

### 2.2 Diffusion Models

While our theoretical framework applies broadly to diffusion models, we adopt the flow matching formulation with linear interpolants (Liu et al., 2022a; Lipman et al., 2022; Albergo et al., 2023) for clarity. Consider a base distribution $X_0 \sim P_0 = \mathcal{N}(0, I)$ and a target conditional data distribution $X_1 \sim P_r(\cdot \mid c)$. The linear interpolant is defined as $X_t = (1-t)X_0 + tX_1$ for $t \in [0, 1]$.

Flow matching trains a neural vector field $v_\theta$ by minimizing the regression objective:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}\left[\|v_\theta(X_t, t, c) - (X_1 - X_0)\|_2^2\right], \tag{3}$$

which yields the population-level optimum:

$$v^\star(x, t, c) = \mathbb{E}[X_1 - X_0 \mid X_t = x, c]. \tag{4}$$

This optimal vector field drives the probability flow ordinary differential equation (ODE)

$$\frac{dX_t}{dt} = v^\star(X_t, t, c), \tag{5}$$

which deterministically transports the prior distribution $P_0$ to the target distribution $P_r(\cdot \mid c)$.

# 3 AMORTIZED MOMENT MATCHING: THEORETICAL MOTIVATION

While the flow-matching objective Eq. 3 is conventionally interpreted as learning a transport vector field, we demonstrate from a distributional perspective that its optimal vector field (Eq. 4) fundamentally encodes conditional data statistics. Specifically, by analyzing the orthogonal projection of this denoiser onto polynomial function classes, we reveal a natural moment hierarchy. This connection allows us to isolate the affine case to derive a computationally tractable second-order matching objective. We formalize this insight below (see Appendix B for the rigorous formulation and proof).

**Theorem 3.1** (Denoising Projection Moment Hierarchy, Informal)**.** *Under the linear interpolant $X_t = tX_1 + (1-t)X_0$, fix a condition $c$ and time $t \in (0,1)$. Let $\mathcal{P}_n$ denote the space of vector-valued polynomials in $X_t$ of degree at most $n$. The optimal degree-$n$ polynomial denoiser*

$$v_n^\star = \arg\min_{v \in \mathcal{P}_n} \mathbb{E}\left[\|v(X_t,t,c) - (X_1 - X_0)\|^2 \mid c\right], \tag{6}$$

*is exactly the orthogonal projection of the unconstrained flow-matching velocity $v^\star$ onto $\mathcal{P}_n$.*

*Crucially, $v_n^\star$ satisfies the normal equations, meaning it perfectly matches the moment tensors of the target velocity up to order $n$:*

$$\mathbb{E}\left[v_n^\star(X_t,t,c) \otimes X_t^{\otimes k} \mid c\right] = \mathbb{E}\left[(X_1 - X_0) \otimes X_t^{\otimes k} \mid c\right] := G_k^t(P \mid c), \quad \forall k = 0,\dots,n. \tag{7}$$

*Writing the data moments as $M_j(P \mid c) = \mathbb{E}[X_1^{\otimes j} \mid c]$, $G_k^t$ obey a triangular recurrence relation:*

$$G_k^t(P \mid c) = t^k M_{k+1}(P \mid c) + F_k^t(M_0,\dots,M_k), \tag{8}$$

*where $F_k^t$ is a deterministic function depending solely on $t$, the Gaussian moments of $X_0$, and the lower-order data moments $\{M_j\}_{j=0}^k$. Consequently, optimizing the degree-$n$ denoising objective uniquely determines the conditional data moments up to order $n+1$.*

This theorem reveals that diffusion models inherently learn a hierarchy of statistics, where the coefficients of the degree-zero, affine, and higher-degree terms correspond to the mean, covariance, and higher-order tensors, respectively. Since matching high-order tensors is computationally prohibitive, we isolate the affine case. This provides a tractable yet effective instantiation, formalized as follows.

**Corollary 3.2** (Affine Projection Identifies Conditional Covariance)**.** *Under the setting of Theorem 3.1, assume $X_1 \mid c$ has a finite second moment. Let $\mu(c) = \mathbb{E}[X_1 \mid c]$ and $\Sigma(c) = \mathrm{Cov}(X_1 \mid c)$. Then the degree-zero projection is exactly the conditional mean:*

$$v_0^\star(z,t,c) = \mu(c), \tag{9}$$

*and the degree-one projection admits the affine form:*

$$v_1^\star(z,t,c) = \mu(c) + R_t(c)\big(z - t\mu(c)\big), \tag{10}$$

*where the affine operator is given by*

$$R_t(c) = \left[t\Sigma(c) - (1-t)I\right]\left[t^2\Sigma(c) + (1-t)^2 I\right]^{-1}. \tag{11}$$

*For a fixed $t \in (0,1)$, the mapping $\Sigma(c) \mapsto R_t(c)$ is rigorously injective. Therefore, the affine component of the denoising velocity serves as an exact, injective transformation of the covariance.*

# 4 THE AMORTIZED FRÉCHET DISTANCE LOSS

Motivated by the insight of the affine case, we propose the Amortized Fréchet Distance (AMFD) loss. We instantiate this matching process within a pre-trained representation space, as adopted by Yang et al. (2026). Below we detail the parameterization and the alternating training algorithm.

## 4.1 AMORTIZER PARAMETERIZATION

Let $\{\phi_\ell\}_{\ell=1}^L$ be a set of frozen representation encoders. For a data branch $b \in \{r, g\}$, let $X_r^\ell = \phi_\ell(X_r)$ and $X_g^\ell = \phi_\ell(G_\theta(z,c))$ denote the real and generated features in $\mathbb{R}^{D_\ell}$. To capture their

statistics, AMFD employs a neural amortizer that predicts a conditional mean $\mu_b^\ell(c) \in \mathbb{R}^{D_\ell}$ and an affine operator, which we parameterize residually as $R_{b,t}^\ell(c) = -I + tA_{b,t}^\ell(c)$ for numerical stability.

For a target feature $X_1^\ell \in \{X_r^\ell, X_g^\ell\}$ and noise $X_0^\ell \sim \mathcal{N}(0, I)$, we construct the interpolant $X_t^\ell = tX_1^\ell + (1-t)X_0^\ell$. Substituting the residual parameterization, the affine denoiser takes the form:

$$\hat{v}_b^\ell(X_t^\ell, t, c) = \mu_b^\ell(c) + R_{b,t}^\ell(c)\tilde{X}_t^\ell = \mu_b^\ell(c) - \tilde{X}_t^\ell + tA_{b,t}^\ell(c)\tilde{X}_t^\ell, \tag{12}$$

where $\tilde{X}_t^\ell = X_t^\ell - t\mu_b^\ell(c)$ is the time-centered feature.

Materializing $A_{b,t}^\ell(c)$ explicitly would incur a prohibitive $O(D_\ell^2)$ memory cost in high-dimensional representations. We circumvent this by implicitly defining the operator via its JVP action on arbitrary vectors $u \in \mathbb{R}^{D_\ell}$. Given a differentiable neural mapping $f_{b,\psi}^\ell(s, t, c) : \mathbb{R}^{D_\ell} \to \mathbb{R}^{D_\ell}$ governed by an auxiliary input $s$, the exact operator action is efficiently computed as:

$$A_{b,t}^\ell(c)u = \left.\frac{\partial f_{b,\psi}^\ell(s,t,c)}{\partial s}\right|_{s=0} u = \mathrm{JVP}_s\left[f_{b,\psi}^\ell(s,t,c)\right]_{s=0}(u). \tag{13}$$

Detailed configurations for the JVP operation are provided in Appendix D.2 and D.4.

### 4.2 Training Algorithm

Figure 2 (b) outlines the AMFD training procedure, which relies on an alternating optimization scheme (detailed alogrithm see Appendix C.1). The amortizers learn to estimate the conditional real and generated statistics, while the generator is trained to minimize the discrepancy between them. To ensure unbiased scale aggregation across diverse spaces and prevent high-dimensional features from disproportionately dominating the objective, all per-encoder mean, covariance, and amortizer losses employ the normalized inner product $\langle a, b\rangle_\ell = \frac{1}{D_\ell}\langle a, b\rangle$ and its induced norm $\|a\|_\ell^2 = \langle a, a\rangle_\ell$.

**Amortizer update.** For branch $b \in \{r, g\}$, define the target feature as

$$X_1^\ell = \begin{cases} h_r^\ell, & b = r, \\ \mathrm{sg}[h_g^\ell], & b = g, \end{cases}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operation. Given a sampled time $t \sim \mathcal{U}(0, 1)$ and noise $X_0^\ell \sim \mathcal{N}(0, I)$, we form the linear interpolant $X_t^\ell$ and train the affine denoiser (Eq. 12) via:

$$\mathcal{L}_b^\ell(\psi) = \mathbb{E}\left[\left\|\hat{v}_b^\ell(X_t^\ell, t, c) - (X_1^\ell - X_0^\ell)\right\|_\ell^2\right] + \mathbb{E}\left[\left\|\mu_b^\ell(c) - X_1^\ell\right\|_\ell^2\right]. \tag{14}$$

At the population optimum, the diffusion matching term natively identifies the conditional mean and covariance functions. The explicit mean-regression term is theoretically redundant but practically serves as a regularization factor that consistently improves empirical convergence. The total amortizer objective sums over all branches and encoders:

$$\mathcal{L}_{\mathrm{amort}}(\psi) = \sum_{\ell=1}^{L}\left(\mathcal{L}_r^\ell(\psi) + \mathcal{L}_g^\ell(\psi)\right). \tag{15}$$

**Generator mean loss.** The generator should modify its features along the direction that explicitly minimizes the mean discrepancy $\mu_g^\ell(c) - \mu_r^\ell(c)$. We optimize this using the first-variation objective:

$$\mathcal{L}_\mu^\ell(\theta) = \left\langle \mathrm{sg}\left[\mu_g^\ell(c) - \mu_r^\ell(c)\right], h_g^\ell\right\rangle_\ell, \tag{16}$$

where gradients are only routed through the generated features $h_g^\ell$.

**Generator covariance loss.** For a given encoder $\ell$, we define the centered generated feature as $u^\ell = h_g^\ell - \mathrm{sg}[\mu_g^\ell(c)]$. Ideally, the generator would directly minimize the quadratic discrepancy between the two conditional covariance matrices. Though the amortizer-predicted affine operator $A_t^\ell(c)$ maps uniquely to $\Sigma^\ell(c)$, directly substituting $A_t^\ell(c)$ for $\Sigma^\ell(c)$ is poorly conditioned.

Instead, we project the discrepancy through the local Jacobian of the mapping $A_t^\ell \mapsto \Sigma^\ell$ evaluated at the generated branch. This yields the Jacobi operator $\mathcal{J}_{g,t}^\ell(c)u = (1+t)u - t^2 A_{g,t}^\ell(c)u$, which defines the local pullback force:

$$\delta a_{\mathrm{J}}^\ell = \mathcal{J}_{g,t}^\ell(c)\left(A_{g,t}^\ell(c) - A_{r,t}^\ell(c)\right)\mathcal{J}_{g,t}^\ell(c)u^\ell. \tag{17}$$

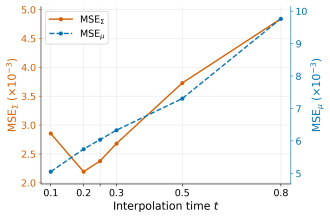


Figure 4: Estimation accuracy across interpolation time $t$.

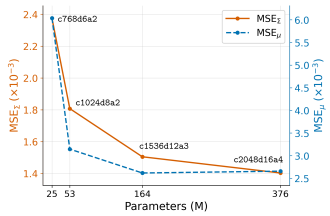


Figure 5: Estimation accuracy across amortizer MLP sizes.

Table 1: Estimation accuracy with (AMFD-C) and without (AMFD-U) class conditioning (MSE $\times 10^{-3}$).

| Variant | MSE$_\mu$ | MSE$_\Sigma$ |
|---|---|---|
| AMFD-C | 5.75 | 2.18 |
| AMFD-U | 0.0106 | 0.114 |

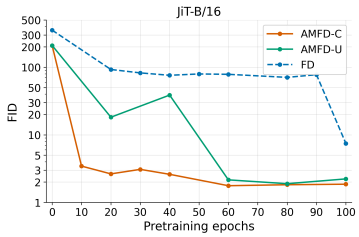


Figure 6: Impact of pretraining epochs on JiT-B/16 post-training.

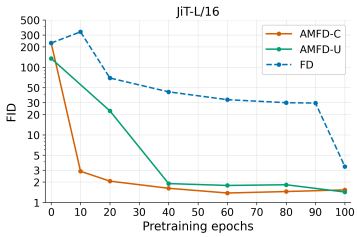


Figure 7: Impact of pretraining epochs on JiT-L/16 post-training.




Figure 8: AMFD in native generative spaces.

We then compute the generator covariance loss as:

$$\mathcal{L}_{\text{cov}}^{\ell}(\theta) = \omega_{\text{cov}}(t)\left\langle u^{\ell}, \delta a_{\text{J}}^{\ell}\right\rangle_{\ell}, \tag{18}$$

where $\omega_{\text{cov}}(t)$ is a scaling coefficient (derivation detailed in Appendix C.2) to approximately match the weighting of the FD objective (Eq. 2). During the generator update, the amortizers are frozen, but the internal JVP operator actions (Eq. 13) are tethered to $u^{\ell}$, ensuring exact gradient flow to the generator. Notably, ignoring this local geometry yields a non-Jacobi variant $\delta a_{\text{noJ}}^{\ell} = (A_{g,t}^{\ell}(c) - A_{r,t}^{\ell}(c))u^{\ell}$, which we empirically found to be significantly less effective.

**Multi-representation normalization.** Aggregating losses across multiple heterogeneous encoders often produces generator gradients with highly disproportionate magnitudes. Similar to FD-loss (Yang et al., 2026), we normalize each loss via a per-sample gradient proxy, thereby stabilizing the optimization process. For encoder $\ell$, the proxy vector is defined as:

$$g_{\text{proxy}}^{\ell} = \text{sg}\left[\mu_g^{\ell}(c) - \mu_r^{\ell}(c)\right] + 2\omega_{\text{cov}}(t)\delta a_{\text{J}}^{\ell}. \tag{19}$$

The normalized per-encoder generator loss is subsequently computed as:

$$\mathcal{L}_{\text{norm}}^{\ell}(\theta) = \mathbb{E}\left[\frac{\mathcal{L}_{\mu}^{\ell}(\theta) + \mathcal{L}_{\text{cov}}^{\ell}(\theta)}{\left(\sum_{j=1}^{D_\ell}\left(g_{\text{proxy},j}^{\ell}\right)^2\right)^{\rho} + \epsilon}\right], \tag{20}$$

where $\rho$ and $\epsilon$ serve as scaling hyperparameters. The final unified generator objective aggregates these normalized losses across all $L$ representation encoders:

$$\mathcal{L}_{\text{gen}}(\theta) = \sum_{\ell=1}^{L}\mathcal{L}_{\text{norm}}^{\ell}(\theta). \tag{21}$$

## 5 Experiments

In this section, we empirically evaluate our proposed AMFD framework. First, we isolate and ablate the specific design choices of the amortizers. Second, we assess AMFD in ImageNet post-training and extend our study to native generative spaces. Finally, we investigate the transferability of the AMFD objective to text-to-image post-training across both pixel and latent spaces.

### 5.1 Amortizer Implementation

We instantiate one amortizer for each representation encoder, yielding a total of $L$ amortizers with randomly initialized weights. For global feature representations, the amortizer is an AdaLN-conditioned MLP (Li et al., 2024); in text-to-image models, a cross-attention module is integrated

into every AdaLN block to process text conditioning. For native generative spaces, the amortizer mirrors the generator's architecture. We refer to this condition-aware configuration as AMFD-C (Conditional). For unconditional generation, denoted as AMFD-U (Unconditional), we retain the same network architecture but replace the conditioning input with a fixed dummy condition. The real and fake branches share identical amortizer parameters, relying exclusively on the time step $t$ to distinguish the specific prediction task. More details are provided in Appendix D.3 and D.4.

### 5.2 Amortizer Ablations

To isolate estimation accuracy, we train the amortizer solely on real Inception-v3 (Szegedy et al., 2016) features. We evaluate its performance using $\mathrm{MSE}_\mu$ and $\mathrm{MSE}_\Sigma$ to measure the discrepancy against the analytical ground truth (details in Appendix D.3).

**Effect of interpolation time $t$.** Eq. 11 shows that the target moments can be perfectly identified at any fixed $t \in (0, 1)$. By sweeping $t$ across this interval, we observe that smaller $t$ values yield consistently lower amortization errors for both the mean and covariance (Figure 4).

**Scaling model capacity.** We evaluate estimation accuracy across varying MLP capacities. As depicted in Figure 5, increasing model size steadily reduces approximation error. However, as $\mathrm{MSE}_\mu$ plateaus and $\mathrm{MSE}_\Sigma$ gains diminish at the `c2048d16a4` scale, we adopt it as our default.

**Effect of class conditioning.** As shown in Table 1, the unconditional setting (AMFD-U) achieves considerably higher estimation accuracy than its conditional counterpart (AMFD-C). This gap likely stems from the inherent simplicity of modeling a single global marginal distribution, compounded by the sparse per-class data density that limits statistical efficiency during conditional matching.

### 5.3 ImageNet Post-Training

**Superior dynamics and results when matching in representation spaces.** We evaluate AMFD for ImageNet post-training using both the multi-step JiT and one-step pMF models. As shown in Table 2, AMFD-U substantially improves the FDr$^6$ score over the FD-loss baseline across both architectures (see Appendix E for full FDr metrics and Appendix F for qualitative samples). The table also reveals that AMFD-C slightly underperforms AMFD-U. We attribute this to two intertwined factors. First, amortizing marginal moments is inherently easier than fitting fine-grained conditional moments, a difference in estimation accuracy explicitly corroborated in Table 1. Second, the discrete class labels in ImageNet represent a relatively simple condition; in such low-complexity conditional spaces, the benefits of conditional matching are outweighed by the estimation difficulty. We further study the robustness of AMFD against FD-loss when the base model is far from a one-step generative regime by analyzing post-training convergence across varying pre-training checkpoints. Notably, while FD-loss completely diverges on early-stage checkpoints, both AMFD variants consistently converge to competitive FIDs (Figures 6 and 7). This demonstrates that the primary driver of robustness is the neural formulation itself. Furthermore, AMFD-C achieves strong FID scores with even fewer pre-training epochs than AMFD-U. This indicates that, despite being harder to fit perfectly, precise class-aware supervision provides further training stabilization.

**Exploration in native generative space.** Empowered by the space complexity efficiency of AMFD, we further investigate whether matching solely the first and second moments within native generative spaces is sufficient to capture the target data distribution. Our evaluation spans four distinct environments with increasing semantic density: raw pixel space, SD-VAE (Rombach et al., 2022) latent space, VA-VAE (Yao et al., 2025) latent space, and RAE (Zheng et al., 2025) latent space. As detailed in Figure 8, both qualitative visualizations and quantitative metrics (FID and FDr$^6$) reveal that generative performance improves progressively as the semantic strength of the space increases. Notably, while the lower-semantic spaces struggle to produce coherent structures, the RAE latent space successfully yields reasonable visual quality with a competitive FID of 3.13. This contrast suggests that matching first- and second-order statistics becomes progressively more effective as the representation space encodes richer and more structured semantics.

### 5.4 Text-to-Image Post-Training

We evaluate AMFD on text-to-image generation at 512×512 resolution using the GenEval (Ghosh et al., 2023) (instruction following) and PickScore (Kirstain et al., 2023) (human preference) bench-

Table 2: System-level comparison for class-conditional generation on ImageNet 256×256. † indicates using CFG interval. FD baseline and AMFD employ SIM (SigLIP, Inception, MAE).

| Method | Space | NFE | Params | FDr$^6$↓ | FID↓ | IS↑ | Precision↑ | Recall↑ |
|---|---|---|---|---|---|---|---|---|
| 50k validation images | - | - | - | 1.00 | 1.68 | 232.2 | 0.75 | 0.66 |
| *Multi-step models* | | | | | | | | |
| VAR-d30 (Tian et al., 2024) | discrete | $10 \times 2$ | 2B | 6.70 | 1.97 | 304.6 | 0.82 | 0.59 |
| BAR-L (Yu et al., 2026) | discrete | $256 \times 2{\times}4$ | 1.1B | 3.57 | 1.01 | 281.9 | 0.77 | 0.68 |
| SiT-XL/2 (Ma et al., 2024) | latent | $250 \times 2$ | 675M | 8.44 | 2.12 | 256.7 | 0.81 | 0.60 |
| MAR-L (Li et al., 2024) | latent | $256 \times 2{\times}100$ | 478M | 6.68 | 1.80 | 293.4 | 0.80 | 0.60 |
| FlowAR-H (Ren et al., 2024) | latent | $50 \times 2^{\dagger}$ | 1.9B | 6.13 | 1.68 | 274.1 | 0.80 | 0.62 |
| MAR-H (Li et al., 2024) | latent | $256 \times 2{\times}100$ | 942M | 5.61 | 1.56 | 299.5 | 0.80 | 0.62 |
| MAR-L, DeTok (Yang et al., 2025) | latent | $256 \times 2{\times}100$ | 478M | 5.49 | 1.39 | 306.2 | 0.81 | 0.62 |
| REG (Wu et al., 2026) | latent | $250 \times 2^{\dagger}$ | 685M | 4.64 | 1.54 | 302.9 | 0.78 | 0.62 |
| SiT-XL/2-REPA (Yu et al., 2024) | latent | $250 \times 2^{\dagger}$ | 675M | 5.45 | 1.42 | 306.1 | 0.80 | 0.65 |
| LightningDiT (Yao et al., 2025) | latent | $250 \times 2$ | 675M | 4.57 | 1.42 | 294.3 | 0.80 | 0.64 |
| DDT-XL (Wang et al., 2026) | latent | $250 \times 2$ | 675M | 5.70 | 1.26 | 309.3 | 0.79 | 0.66 |
| REPA-E (Leng et al., 2025) | latent | $250 \times 2^{\dagger}$ | 676M | 3.04 | 1.17 | 298.3 | 0.79 | 0.66 |
| RAE-XL (Zheng et al., 2025) | latent | $50 \times 2^{\dagger}$ | 839M | 3.26 | 1.16 | 261.0 | 0.77 | 0.67 |
| PixNerd-XL (Wang et al., 2025) | pixel | $100 \times 2$ | 1.0B | 5.01 | 2.10 | 318.8 | 0.81 | 0.59 |
| JiT-B (Li & He, 2026) | pixel | $50 \times 2 \times 2^{\dagger}$ | 131M | 15.65 | 3.71 | 269.0 | 0.81 | 0.50 |
| + FD-loss (Yang et al., 2026) | pixel | 1 | 131M | 5.53 | 1.00 | 344.6 | 0.78 | 0.60 |
| + AMFD-C (Ours) | pixel | 1 | 131M | 4.75 | 0.95 | 319.4 | 0.77 | 0.67 |
| + AMFD-U (Ours) | pixel | 1 | 131M | 3.91 | 0.95 | 325.2 | 0.77 | 0.60 |
| JiT-L (Li & He, 2026) | pixel | $50 \times 2 \times 2^{\dagger}$ | 459M | 10.73 | 2.59 | 288.5 | 0.79 | 0.59 |
| + FD-loss (Yang et al., 2026) | pixel | 1 | 459M | 3.24 | 0.77 | 317.3 | 0.77 | 0.66 |
| + AMFD-C (Ours) | pixel | 1 | 459M | 2.67 | 0.87 | 325.1 | 0.76 | 0.69 |
| + AMFD-U (Ours) | pixel | 1 | 459M | 2.02 | 0.85 | 319.8 | 0.77 | 0.65 |
| JiT-H (Li & He, 2026) | pixel | $50 \times 2 \times 2^{\dagger}$ | 953M | 7.66 | 1.97 | 296.0 | 0.78 | 0.63 |
| + FD-loss (Yang et al., 2026) | pixel | 1 | 953M | 2.65 | 0.75 | 313.0 | 0.76 | 0.66 |
| + AMFD-C (Ours) | pixel | 1 | 953M | 2.15 | 0.85 | 328.1 | 0.76 | 0.69 |
| + AMFD-U (Ours) | pixel | 1 | 953M | 1.79 | 0.83 | 312.2 | 0.76 | 0.67 |
| *One-step models* | | | | | | | | |
| Drift-L (Deng et al., 2026) | latent | 1 | 463M | 10.92 | 1.53 | 257.2 | 0.79 | 0.63 |
| Drift-L (Deng et al., 2026) | pixel | 1 | 465M | 10.51 | 1.43 | 305.8 | 0.81 | 0.60 |
| iMF-XL (Geng et al., 2026) | latent | 1 | 610M | 8.39 | 1.82 | 278.9 | 0.78 | 0.63 |
| + FD-loss (Yang et al., 2026) | latent | 1 | 610M | 2.45 | 0.76 | 301.3 | 0.77 | 0.67 |
| pMF-B (Lu et al., 2026) | pixel | 1 | 118M | 13.70 | 3.31 | 254.6 | 0.81 | 0.52 |
| + FD (Yang et al., 2026) | pixel | 1 | 118M | 3.50 | 0.85 | 331.4 | 0.77 | 0.65 |
| + AMFD-C (Ours) | pixel | 1 | 118M | 3.94 | 0.95 | 315.9 | 0.77 | 0.66 |
| + AMFD-U (Ours) | pixel | 1 | 118M | 3.43 | 0.92 | 310.1 | 0.77 | 0.65 |
| pMF-L (Lu et al., 2026) | pixel | 1 | 410M | 9.09 | 2.72 | 261.7 | 0.81 | 0.56 |
| + FD-loss (Yang et al., 2026) | pixel | 1 | 410M | 2.09 | 0.78 | 309.2 | 0.76 | 0.67 |
| + AMFD-C (Ours) | pixel | 1 | 410M | 2.25 | 0.88 | 321.8 | 0.76 | 0.68 |
| + AMFD-U (Ours) | pixel | 1 | 410M | 2.01 | 0.86 | 306.7 | 0.76 | 0.68 |
| pMF-H (Lu et al., 2026) | pixel | 1 | 935M | 6.87 | 2.29 | 267.2 | 0.80 | 0.59 |
| + FD-loss (Yang et al., 2026) | pixel | 1 | 935M | 1.89 | 0.77 | 310.1 | 0.77 | 0.68 |
| + AMFD-C (Ours) | pixel | 1 | 935M | 1.93 | 0.86 | 323.0 | 0.76 | 0.69 |
| + AMFD-U (Ours) | pixel | 1 | 935M | 1.75 | 0.85 | 307.3 | 0.76 | 0.68 |

marks across diverse architectures: the pixel-space PixelGen (Ma et al., 2026), the latent-space FLUX.2 [klein] 4B Base (Black Forest Labs, 2026), and its 4-step distilled counterpart. As shown in Table 3, our AMFD establishes a new paradigm with several pivotal insights:

**Inherent stability in complex T2I.** Even without conditioning, AMFD-U-SIM consistently outperforms the FD-SIM baseline across all architectures on both GenEval and PickScore (Rows 2 vs. 3, 6 vs. 7, and 12 vs. 14). This confirms that our amortized formulation inherently provides better training dynamics in highly complex text-to-image scenarios.

**The critical role of conditioning.** When post-training using a few-step teacher (which inherently operates closer to a one-step generator and already possesses robust conditional alignment), AMFD-U-SIM performs remarkably well (Row 14). However, when using a standard multi-step diffusion model, condition-aware matching becomes absolutely crucial. For instance, in the multi-step FLUX.2 Base, AMFD-C-SIM dramatically boosts the GenEval score from 0.749 to 0.833 compared to its unconditional counterpart (Rows 7 vs. 8), unlocking similarly massive gains in PixelGen (0.713 to 0.760, Rows 3 vs. 4). This completely validates the necessity of condition-aware matching for standard diffusion distillation—a stark contrast to our ImageNet findings, driven by the vastly higher semantic complexity of text prompts compared to class labels.

Table 3: GenEval and PickScore for text-to-image post-training. Higher is better for all metrics.

| Row | Method | NFE | Single Obj. | Two Obj. | Counting | Colors | Position | Color Attr. | Overall | PickScore |
|---|---|---|---|---|---|---|---|---|---|---|
| | *Pixel space, multi-step* | | | | | | | | | |
| 4 | PixelGen | 25×2 | 0.994 | 0.886 | 0.578 | 0.902 | 0.708 | 0.698 | **0.794** | **21.02** |
| 6 | FD-SIM (Yang et al., 2026) | 1 | 0.984 | 0.795 | 0.416 | 0.753 | 0.535 | 0.415 | 0.650 | 20.33 |
| 8 | AMFD-U-SIM (Ours) | 1 | 0.975 | 0.831 | 0.469 | 0.769 | 0.658 | 0.578 | 0.713 | 20.68 |
| 10 | AMFD-C-SIM (Ours) | 1 | 0.969 | 0.864 | 0.513 | 0.846 | 0.723 | 0.645 | <u>0.760</u> | <u>20.80</u> |
| | *VAE latent space, multi-step* | | | | | | | | | |
| 13 | FLUX.2 [klein] 4B Base | 50×2 | 1.000 | 0.864 | 0.759 | 0.899 | 0.505 | 0.642 | 0.778 | **21.50** |
| 15 | FD-SIM (Yang et al., 2026) | 1 | 1.000 | 0.934 | 0.653 | 0.824 | 0.448 | 0.598 | 0.743 | 21.31 |
| 17 | AMFD-U-SIM (Ours) | 1 | 0.997 | 0.917 | 0.644 | 0.827 | 0.523 | 0.590 | 0.749 | 21.37 |
| 19 | AMFD-C-SIM (Ours) | 1 | 1.000 | 0.957 | 0.750 | 0.910 | 0.733 | 0.650 | <u>0.833</u> | 21.41 |
| 21 | AMFD-C-10 enc. (Ours) | 1 | 1.000 | 0.962 | 0.766 | 0.886 | 0.720 | 0.685 | **0.836** | <u>21.46</u> |
| | *VAE latent space, few-step* | | | | | | | | | |
| 24 | FLUX.2 [klein] 4B | 4 | 0.994 | 0.904 | 0.791 | 0.880 | 0.575 | 0.623 | 0.794 | **21.85** |
| 26 | DMD2 (Yin et al., 2025) | 1 | 0.997 | 0.894 | **0.806** | 0.864 | 0.603 | 0.660 | 0.804 | - |
| 28 | FD-SIM (Yang et al., 2026) | 1 | **1.000** | 0.944 | 0.716 | 0.878 | 0.618 | 0.658 | 0.802 | 21.62 |
| 30 | iRDM (Feng et al., 2026) | 1 | 0.994 | 0.924 | 0.756 | **0.923** | 0.650 | 0.708 | 0.826 | <u>21.82</u> |
| 32 | AMFD-U-SIM (Ours) | 1 | 0.994 | 0.929 | 0.741 | 0.902 | 0.638 | 0.678 | 0.813 | 21.77 |
| 34 | AMFD-C-SIM (Ours) | 1 | 0.997 | 0.955 | 0.791 | **0.923** | 0.670 | 0.740 | **0.846** | **21.85** |
| 36 | AMFD-C-10 enc. (Ours) | 1 | **1.000** | **0.957** | 0.778 | 0.920 | **0.680** | **0.733** | <u>0.845</u> | <u>21.82</u> |

**Sufficiency of the first two moments.** FD-loss (Yang et al., 2026) has empirically established that matching the first two moments yields strong generative performance. We further corroborate this by comparing AMFD (Rows 15 & 16) with the recent iRDM (Feng et al., 2026) (Row 13), which theoretically matches *all* statistical moments via an MMD criterion. By achieving superior GenEval scores and equal or better PickScores, our results further demonstrate that, when empowered by neural amortization, the first two moments already provide abundant supervisory signals.

**Effectiveness of the compact SIM set.** Expanding the representation set from SIM to 10 encoders (Table 4, similar to Feng et al. (2026)) yields comparable performance across both FLUX.2 [klein] models (Rows 8 vs. 9; 15 vs. 16). While standard metrics may not capture all real-world visual nuances, this suggests the lightweight SIM ensemble already provides sufficient features for strong text-to-image alignment, offering a practical balance of efficiency and quality.

Ultimately, these advantages allow our one-step AMFD models to significantly surpass both the multi-step FLUX.2 [klein] 4B Base teacher and its 4-step distilled variant on GenEval, while achieving highly competitive, on-par performance on PickScore. This demonstrates the formidable capability of our method. Qualitative results are in Figure 3 and Appendix Figure 11.

## 6 Limitations

While effective, our study presents several limitations. First, although our theory accommodates arbitrary-order moments, AMFD is restricted to matching only the first two moments, leaving higher-order statistics uncaptured. Second, our reliance on off-the-shelf representation encoders, which are typically pretrained at low resolutions, bottlenecks the direct applicability to high-resolution generation. Finally, AMFD serves as a post-training objective, while it is currently unable to train generative models entirely from scratch.

## 7 Conclusion

In this work, we introduced *amortized moment matching*, a principled framework that leverages neural amortizers to provide scalable distributional training signals for generative models. By analyzing diffusion denoisers through the lens of polynomial projections, we established that they inherently learn a hierarchy of conditional statistics. We instantiated the tractable second-order case as AMFD, a post-training paradigm designed to elevate the synthesis quality and convert multi-step models into one-step generators. AMFD yields superior one-step generation performance in ImageNet post-training and scales seamlessly to text-to-image generation across both pixel and VAE latent spaces, demonstrating strong instruction-following proficiencies and high-fidelity generation.

### AI USE STATEMENT

In this work, we used generative AI tools for language polishing and assisting in writing and debugging source code. We have not used generative AI tools for generating novel research ideas, deriving theoretical proofs, designing the core methodology, or fabricating experimental results, and the rest of the required disclosure tasks are not applicable to this work. We have thoroughly reviewed all AI-assisted work. Specifically, all LLM-assisted text was manually edited to ensure scientific accuracy, and all LLM-generated code was strictly verified and tested for correctness. We take full responsibility for the final content of this work.

### ETHICS STATEMENT

While our primary contribution is algorithmic, we recognize the dual-use nature of highly efficient generative models, which lower the computational barrier for creating malicious content such as deepfakes or misinformation. Furthermore, because our post-training methodology is applied to foundational models, the resulting models may inherit or faithfully reproduce biases present in the original pre-training datasets. We strongly encourage practitioners to integrate safety measures, such as origin watermarking and post-hoc safety filters, when deploying these models.

### REPRODUCIBILITY STATEMENT

We are committed to the reproducibility of both our theoretical and empirical claims. Complete mathematical proofs establishing the connection between diffusion denoisers and explicit moment identification are provided in the Appendix. For empirical reproducibility, all dataset processing steps, network architectures, optimization hyperparameters, and exact evaluation protocols for metrics are detailed in the experimental section and the Appendix. We will release our code and checkpoints upon publication.

## REFERENCES

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence. `https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence`, 2026. Model weights: `https://huggingface.co/collections/black-forest-labs/flux2`.

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

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, et al. Perception encoder: The best visual embeddings are not at the output of the network. *Advances in Neural Information Processing Systems*, 38:60884–60937, 2026.

Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and O Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *arXiv preprint arXiv:2310.06721*, 2023.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025a.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025b.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

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*, pp. 12873–12883, 2021.

David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 370–382, 2025.

Lan Feng, Wuyang Li, Eloi Zablocki, Matthieu Cord, and Alexandre Alahi. Representation distribution matching for one-step visual generation. *arXiv preprint arXiv:2607.02375*, 2026.

Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor G Turrisi da Costa, Louis Béthune, Zhe Gan, et al. Multimodal autoregressive pre-training of large vision encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9641–9654, 2025.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30467–30476, 2026.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

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.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International conference on machine learning*, pp. 1718–1727. PMLR, 2015.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

Shanchuan Lin, Ceyuan Yang, Zhijie Lin, Hao Chen, and Haoqi Fan. Continuous adversarial flow models. *arXiv preprint arXiv:2604.11521*, 2026.

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

Wenze Liu and Xiangyu Yue. Learning to integrate diffusion odes by averaging the derivatives. *arXiv preprint arXiv:2505.14502*, 2025.

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*, 2022a.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Pixel diffusion beats latent diffusion with perceptual loss. *arXiv preprint arXiv:2602.02493*, 2026.

Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *International Conference on Learning Representations*, volume 2024, pp. 49233–49263, 2024.

Alexander Mathiasen and Frederik Hvilshøj. Backpropagating through fr\'echet inception distance. *arXiv preprint arXiv:2009.14075*, 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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.

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.

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*, pp. 10684–10695, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Peng Sun, Zhenglin Cheng, Deyuan Liu, Jun Xie, Xinyi Shang, and Tao Lin. Three-body scattering for generative modeling. *arXiv preprint arXiv:2607.18198*, 2026.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

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, 2024.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40633–40642, 2026.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36:8406–8441, 2023.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Ming-Ming Cheng, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *Advances in Neural Information Processing Systems*, 38:7714–7743, 2026.

Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good visual tokenizers. *arXiv preprint arXiv:2507.15856*, 2025.

Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation fr\'echet loss for visual generation. *arXiv preprint arXiv:2604.28190*, 2026.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15703–15712, 2025.

Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, et al. Echo-4o: Harnessing the power of gpt-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in Neural Information Processing Systems*, 37:47455–47487, 2025.

Qihang Yu, Qihao Liu, Ju He, Xinyang Zhang, Yang Liu, Liang-Chieh Chen, and Xi Chen. Autoregressive image generation with masked bit modeling. *arXiv preprint arXiv:2602.09024*, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *arXiv preprint arXiv:2412.06329*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025a.

Linqi Zhou, Mathias Parger, Ayaan Haque, and Jiaming Song. Terminal velocity matching. *arXiv preprint arXiv:2511.19797*, 2025b.

# APPENDIX

## A RELATED WORK

### A.1 DIFFUSION AND FLOW MATCHING

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) and flow matching (Liu et al., 2022a; Lipman et al., 2022) typically learn unrestricted denoising functions or velocity fields to iteratively transport samples from a prior to the data distribution. From a statistical perspective, classical Tweedie identities connect the optimal Gaussian denoiser to posterior moments conditioned on a particular noisy observation (Manor & Michaeli, 2024; Boys et al., 2023). Our perspective differs fundamentally: by restricting the denoising regression to polynomial function classes, we study its projection coefficients. Crucially, these coefficients depend on the conditional data distribution, rather than on an individual noised sample, and form a hierarchy that identifies increasingly high-order data moments. Accordingly, we do not use the learned denoiser for generative sampling. Instead, we utilize these amortized coefficients as explicit distributional training signals for a separate generator.

### A.2 MOMENT MATCHING IN REPRESENTATION SPACES

Moment matching provides a direct approach to generative modeling by comparing statistics of real and generated distributions. Traditional methods rely on nonparametric frameworks, optimizing empirical metrics such as kernel mean embeddings and maximum mean discrepancy (MMD) from finite sample sets (Sriperumbudur et al., 2010; Gretton et al., 2012; Li et al., 2015; Dziugaite et al., 2015; Li et al., 2017). A complementary line of work demonstrates that pretrained visual encoders provide highly effective spaces for generative supervision. Perceptual objectives (Zhang et al., 2018) compare paired images using frozen deep features, projected GANs (Sauer et al., 2021) construct discriminators on top of pretrained representations, representation alignment methods (Yu et al., 2024) supervise internal generative features, and representation encoders (Zheng et al., 2025) train generative models natively in representation spaces. Recent work has begun to directly optimize discrepancies between real and generated distributions in representation spaces. Drifting Models (Deng et al., 2026) construct a kernel-induced field that evolves the generated distribution during training. FD-loss (Yang et al., 2026) demonstrates that representation-space means and covariances can be explicitly optimized by decoupling population-level moment estimation from generator minibatches. Several concurrent works have also explored representation-space distribution matching to enable strong one-step generation. Specifically, Representation Distribution Matching (RDM) (Feng et al., 2026) utilizes MMD estimators, while Three-Body Scattering Modeling (TBSM) (Sun et al., 2026) employs an online tracker to learn an energy-distance descent field. Rather than estimating empirical marginal statistics or learning sample-dependent transport fields, AMFD amortizes condition-level moment coefficients derived from polynomial projections of a diffusion denoiser.

### A.3 DISTILLATION AND POST-TRAINING FOR ONE-STEP GENERATION

State-of-the-art generative paradigms, such as diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) and flow matching (Liu et al., 2022a; Lipman et al., 2022), achieve unprecedented synthesis quality but typically rely on slow, iterative generation processes. To address this, generative distillation and post-training techniques have been extensively developed to compress these iterative processes into efficient one-step generators. Existing approaches can be broadly categorized into several directions. Direct distillation methods (Luhman & Luhman, 2021; Salimans & Ho, 2022) explicitly matches the input-output or intermediate mappings of a multi-step teacher. Trajectory consistency methods (Song et al., 2023; Boffi et al., 2024; Liu & Yue, 2025; Geng et al., 2025; Zhou et al., 2025a) enforce consistency along the teacher's deterministic ODE trajectory. GAN-based post-training methods (Sauer et al., 2024; Lin et al., 2025) leverage adversarial objectives to directly align the one-step generated distribution with the real data distribution. Furthermore, variational score distillation (VSD) (Wang et al., 2023; Yin et al., 2024) optimize the one-step generator by estimating and matching the score functions of the target and generated distributions. Although AMFD shares the alternating optimization structure of adversarial and VSD-based post-training, its amortizers serve a fundamentally different statistical role. A

GAN discriminator learns to classify real versus generated samples, and VSD-style networks estimate the full score fields of intermediate noisy distributions. In contrast, AMFD neural amortizers directly regress specific statistical targets: conditional means and covariance-dependent operators. Consequently, AMFD bypasses traditional adversarial dynamics and the necessity to reproduce a fixed teacher trajectory, offering a principled, condition-aware approach to one-step generation.

## B PROOFS

In this section, we present the formal statement and proof of Theorem 3.1, alongside the proof of Corollary 3.2.

**Theorem B.1** (Denoising Projection Moment Hierarchy, Formal Statement)**.** *Fix a condition $c$ and $t \in (0,1)$. Let $X_1 \sim P(\cdot \mid c)$ and $X_0 \sim \mathcal{N}(0, I)$ be independent variables. Define the interpolant $Z = tX_1 + (1-t)X_0$ and the target velocity $V = X_1 - X_0$, assuming $\mathbb{E}[\|V\|^2 \mid c] < \infty$. Let $\mathcal{H} = L^2(P_{Z|c}; \mathbb{R}^D)$ be the Hilbert space of square-integrable vector fields, wherein the unconstrained flow-matching solution is given by $v^\star = \mathbb{E}[V \mid Z, c]$.*

*For any integer $n \geq 0$, let $\mathcal{H}_n \subset \mathcal{H}$ be the subspace of vector-valued polynomials in $Z$ up to degree $n$. The optimal degree-$n$ polynomial denoiser,*

$$v_n^\star = \arg\min_{v \in \mathcal{H}_n} \mathbb{E}\left[\|v(Z) - V\|^2 \mid c\right], \tag{22}$$

*satisfies $v_n^\star = \Pi_n v^\star$, where $\Pi_n$ is the orthogonal projection onto $\mathcal{H}_n$. If the polynomial space $\bigcup_{n\geq 0} \mathcal{H}_n$ is dense in $\mathcal{H}$, these projections converge strongly to $v^\star$, admitting the orthogonal decomposition $v^\star = \sum_{n=0}^{\infty} \Delta_n$ in $L^2(P_{Z|c})$, where $\Delta_0 = v_0^\star$ and $\Delta_n = v_n^\star - v_{n-1}^\star$.*

*Furthermore, by the first-order optimality over $\mathcal{H}_n$, $v_n^\star$ satisfies the normal equations, thereby perfectly matching the target moment tensors of the unconstrained velocity:*

$$\mathbb{E}\left[v_n^\star(Z) \otimes Z^{\otimes k} \mid c\right] = G_k^t(P \mid c) := \mathbb{E}\left[V \otimes Z^{\otimes k} \mid c\right], \qquad k = 0, \ldots, n. \tag{23}$$

*Writing the $j$-th conditional data moment as $M_j(P \mid c) = \mathbb{E}[X_1^{\otimes j} \mid c]$, the target tensors $G_k^t$ obey the triangular recurrence relation:*

$$G_k^t(P \mid c) = t^k M_{k+1}(P \mid c) + F_k^t(M_0, \ldots, M_k), \tag{24}$$

*where $F_k^t$ is a deterministic tensor-valued function depending solely on $t$, the known Gaussian moments of $X_0$, and the lower-order data moments. Consequently, the optimal degree-$n$ polynomial denoiser uniquely determines the conditional data moments up to order $n+1$.*

*Proof.* We begin by establishing the orthogonal projection property and the resulting decomposition.

For an arbitrary $v \in \mathcal{H}$, we decompose the residual as:

$$v(Z) - V = \big(v(Z) - v^\star(Z)\big) + \big(v^\star(Z) - V\big). \tag{25}$$

By definition, $v^\star(Z) = \mathbb{E}[V \mid Z, c]$, which immediately implies $\mathbb{E}[V - v^\star(Z) \mid Z, c] = 0$. Consequently, the cross term vanishes in expectation:

$$\mathbb{E}\left[\big(v(Z) - v^\star(Z)\big)^\top \big(v^\star(Z) - V\big) \mid c\right] = 0. \tag{26}$$

This decouples the mean squared error into:

$$\mathbb{E}\left[\|v(Z) - V\|^2 \mid c\right] = \mathbb{E}\left[\|v(Z) - v^\star(Z)\|^2 \mid c\right] + \mathbb{E}\left[\|v^\star(Z) - V\|^2 \mid c\right]. \tag{27}$$

Since the second term is independent of $v$, minimizing the flow-matching objective over $\mathcal{H}_n$ is mathematically equivalent to minimizing the distance $\|v - v^\star\|_{\mathcal{H}}^2$ over $v \in \mathcal{H}_n$. Thus, the minimizer is exactly the orthogonal projection: $v_n^\star = \Pi_n v^\star$.

To establish the orthogonality of the increments $\Delta_n = v_n^\star - v_{n-1}^\star$, consider any $h \in \mathcal{H}_{n-1}$. Since $v_n^\star = \Pi_n v^\star$ and $\mathcal{H}_{n-1} \subset \mathcal{H}_n$, the projection residual $v^\star - v_n^\star$ is orthogonal to $h$. Similarly, $v^\star - v_{n-1}^\star \perp h$. Subtracting these equations yields $\langle v_n^\star - v_{n-1}^\star, h\rangle_{\mathcal{H}} = 0$, implying $\Delta_n \perp \mathcal{H}_{n-1}$. Because $\Delta_m \in \mathcal{H}_m \subset \mathcal{H}_{n-1}$ for any $m < n$, we obtain $\langle \Delta_n, \Delta_m\rangle_{\mathcal{H}} = 0$. Under the assumption

that the polynomial space is dense in $\mathcal{H}$, the orthogonal projections converge strongly, yielding the valid infinite series $v^\star = \sum_{n=0}^{\infty} \Delta_n$ in $L^2(P_{Z|c})$.

Having established the projection properties, we now turn to the moment hierarchy and the triangular recurrence relation.

Since $\mathcal{H}_n$ is spanned by the tensor polynomials $\{Z^{\otimes k}\}_{k=0}^n$, the condition that the residual $(v_n^\star - V)$ is orthogonal to $\mathcal{H}_n$ in $L^2(P_{Z|c})$ is equivalent to testing against the basis elements. Thus, for any $k \in \{0, \dots, n\}$:

$$\langle v_n^\star - V, z \mapsto z^{\otimes k} \rangle_{\mathcal{H}} = 0 \implies \mathbb{E}\left[\big(v_n^\star(Z) - V\big) \otimes Z^{\otimes k} \mid c\right] = 0. \tag{28}$$

Rearranging this yields the moment matching property: $\mathbb{E}[v_n^\star(Z) \otimes Z^{\otimes k} \mid c] = \mathbb{E}[V \otimes Z^{\otimes k} \mid c] = G_k^t(P \mid c)$.

To establish the triangular recurrence, we explicitly expand $G_k^t(P \mid c)$. Recalling $V = X_1 - X_0$ and $Z = tX_1 + (1-t)X_0$, we write:

$$G_k^t(P \mid c) = \mathbb{E}\left[(X_1 - X_0) \otimes \big(tX_1 + (1-t)X_0\big)^{\otimes k} \mid c\right]. \tag{29}$$

By the non-commutative binomial expansion of the tensor power, the highest-order term of $X_1$ inside the parentheses is $t^k X_1^{\otimes k}$. Distributing $(X_1 - X_0)$ into this expansion, the term with the highest tensor power of $X_1$ is generated solely by $X_1 \otimes (t^k X_1^{\otimes k}) = t^k X_1^{\otimes (k+1)}$.

All other terms in the expansion consist of tensor products mixing at most $k$ copies of $X_1$ with various copies of $X_0$. Because $X_0 \sim \mathcal{N}(0, I)$ is independent of $X_1$ given $c$, the expectations of these mixed terms factorize. The expectations of the $X_0$ components yield standard Gaussian moments (which are known constants), and the expectations of the $X_1$ components yield lower-order data moments $M_j(P \mid c)$ for $j \le k$. Gathering all these lower-order terms into a single deterministic tensor-valued function $F_k^t(M_0, \dots, M_k)$, we obtain:

$$G_k^t(P \mid c) = t^k \mathbb{E}\left[X_1^{\otimes (k+1)} \mid c\right] + F_k^t(M_0, \dots, M_k) = t^k M_{k+1}(P \mid c) + F_k^t(M_0, \dots, M_k). \tag{30}$$

Since $t \in (0, 1)$, the coefficient $t^k$ is strictly positive. Therefore, knowing $G_k^t(P \mid c)$ (which is perfectly matched by the degree-$n$ projection for $k \le n$) and the previously determined lower-order moments allows iterative back-substitution to uniquely solve for $M_{k+1}(P \mid c)$. This concludes the proof. □

**Corollary B.2** (Affine Component Identifies a Covariance Operator)**.** *Under the setting of Theorem 3.1, assume $X_1 \mid c$ has a finite second moment. Let $\mu(c) = \mathbb{E}[X_1 \mid c]$ and $\Sigma(c) = \mathrm{Cov}(X_1 \mid c)$. Then the degree-zero projection is exactly the conditional mean:*

$$v_0^\star(z, t, c) = \mu(c), \tag{31}$$

*and the degree-one projection admits the affine form:*

$$v_1^\star(z, t, c) = \mu(c) + R_t(c)\big(z - t\mu(c)\big), \tag{32}$$

*where the affine operator is given by*

$$R_t(c) = \big[t\Sigma(c) - (1-t)I\big]\big[t^2\Sigma(c) + (1-t)^2 I\big]^{-1}. \tag{33}$$

*For a fixed $t \in (0, 1)$, the mapping $\Sigma(c) \mapsto R_t(c)$ is rigorously injective. Therefore, the affine component of the denoising velocity serves as an exact, injective transformation of the covariance.*

*Proof.* By standard least-squares theory, the optimal degree-zero predictor is simply the conditional expectation of the target:

$$v_0^\star(z, t, c) = \mathbb{E}[V \mid c] = \mu(c). \tag{34}$$

Similarly, the optimal degree-one (affine) predictor is given by the exact population ordinary least squares (OLS) solution:

$$v_1^\star(z, t, c) = \mathbb{E}[V \mid c] + \mathrm{Cov}(V, Z \mid c)\,\mathrm{Cov}(Z \mid c)^{-1}\big(z - \mathbb{E}[Z \mid c]\big). \tag{35}$$

**Algorithm 1** Alternating Training Pipeline for AMFD

**Require:** Generator $G_\theta$, frozen encoders $\{\phi_\ell\}_{\ell=1}^{L}$, amortizers $\{\psi_\ell\}_{\ell=1}^{L}$
1: **repeat**
2: Sample real data $(x_r, c)$ and noise $z$; generate $x_g = G_\theta(z, c)$
3: Compute feature representations $h_r^\ell = \phi_\ell(x_r)$ and $h_g^\ell = \phi_\ell(x_g)$ for all $\ell = 1, \dots, L$
4: Update amortizers by minimizing $\mathcal{L}_{\text{amort}}(\psi)$ (Eq. 15), with generated features $h_g^\ell$ detached
5: Freeze amortizers and update generator $G_\theta$ by minimizing $\mathcal{L}_{\text{gen}}(\theta)$ (Eq. 21)
6: **until** convergence

Recall that $Z = tX_1 + (1-t)X_0$ and $V = X_1 - X_0$. Taking conditional expectations yields $\mathbb{E}[Z \mid c] = t\mu(c)$ and $\mathbb{E}[V \mid c] = \mu(c)$. Utilizing the independence of $X_1$ and $X_0 \sim \mathcal{N}(0, I)$, we directly evaluate the covariance matrices:

$$\mathrm{Cov}(Z \mid c) = t^2\Sigma(c) + (1-t)^2 I, \qquad \mathrm{Cov}(V, Z \mid c) = t\Sigma(c) - (1-t)I. \tag{36}$$

Substituting these moments into the OLS formula directly yields $v_1^\star(z, t, c) = \mu(c) + R_t(c)\big(z - t\mu(c)\big)$, where $R_t(c) = \mathrm{Cov}(V, Z \mid c)\,\mathrm{Cov}(Z \mid c)^{-1}$.

Finally, to establish injectivity, we isolate $\Sigma$ from $R_t$. Right-multiplying $R_t$ by $(t^2\Sigma + (1-t)^2 I)$ and grouping the terms involving $\Sigma$ yields:

$$t(I - tR_t)\Sigma = (1-t)\big(I + (1-t)R_t\big). \tag{37}$$

Observe that the matrix factor $I - tR_t = (1-t)(t^2\Sigma + (1-t)^2 I)^{-1}$ is strictly positive definite, hence invertible. Left-multiplying by its inverse yields the explicit closed-form expression for $\Sigma(c)$:

$$\Sigma(c) = \frac{(1-t)}{t}\big[I - tR_t(c)\big]^{-1}\big[I + (1-t)R_t(c)\big]. \tag{38}$$

Thus, $R_t$ uniquely determines the covariance matrix. □

# C Additional Details on AMFD Training

## C.1 AMFD Training Algorithm

Algorithm 1 outlines the complete alternating optimization procedure for the AMFD framework.

## C.2 Derivation of the Jacobi Covariance Weighting

In this section, we derive the weighting factor $\omega_{\text{cov}}(t)$ introduced in Section 4.2. Our objective is to calibrate the covariance loss such that its gradient approximately matches the standard Fréchet Distance force, $\frac{1}{2}d\Sigma$, at the isotropic equilibrium point $\Sigma = I$.

Crucially, at $\Sigma = I$, the operators $R_t$, $A_t$, and the Jacobi operator $\mathcal{J}_t = I - tR_t$ all reduce to scalar multiples of the identity matrix. Because these operators commute at this specific point, we can compute their differentials using standard scalar calculus.

By the proof of Corollary B.2, the covariance is expressed as:

$$\Sigma = \frac{1-t}{t}(I - tR_t)^{-1}(I + (1-t)R_t). \tag{39}$$

Evaluating the differential $d\Sigma$ with respect to $R_t$ around the commuting point yields:

$$d\Sigma = \frac{1-t}{t}(I - tR_t)^{-2}\big[(1-t)(I - tR_t) - (-t)(I + (1-t)R_t)\big]dR_t = \frac{1-t}{t}\mathcal{J}_t^{-2}dR_t. \tag{40}$$

Using our residual parameterization $R_t = -I + tA_t$, we substitute $dR_t = t\,dA_t$ to obtain:

$$d\Sigma = (1-t)\mathcal{J}_t^{-2}dA_t. \tag{41}$$

Our applied generator force is defined as $\delta a_{\mathrm{J}} = \mathcal{J}_t dA_t \mathcal{J}_t$. Since all matrices commute at $\Sigma = I$, this simplifies to $\delta a_{\mathrm{J}} = \mathcal{J}_t^2 dA_t$, which implies $dA_t = \mathcal{J}_t^{-2} \delta a_{\mathrm{J}}$. Substituting this back into the $d\Sigma$ equation provides the direct relationship:

$$d\Sigma = (1-t)\mathcal{J}_t^{-4} \delta a_{\mathrm{J}}. \tag{42}$$

To compute the explicit scalar value, we evaluate $\mathcal{J}_t(I) = I - tR_t(I)$ using Eq. 11:

$$\mathcal{J}_t(I) = I - t\left(\frac{2t-1}{t^2+(1-t)^2} I\right) = \frac{1-t}{t^2+(1-t)^2} I. \tag{43}$$

Plugging the scalar value of $\mathcal{J}_t(I)^{-4}$ into Eq. 42 yields:

$$d\Sigma = \frac{\left(t^2+(1-t)^2\right)^4}{(1-t)^3} \delta a_{\mathrm{J}}. \tag{44}$$

To precisely recover the target FD force $\frac{1}{2} d\Sigma$, the loss weighting must be:

$$\omega_{\mathrm{cov}}(t) = \frac{1}{2} \frac{\left(t^2+(1-t)^2\right)^4}{(1-t)^3}. \tag{45}$$

# D Implementation Details

In this section, we provide comprehensive implementation details for our experiments. Specifically, we specify the representation encoders utilized (§D.1), introduce a custom analytical JVP implementation designed to accelerate amortizer training (§D.2), outline the experimental settings for our amortizer ablation studies (§D.3), and provide the default amortizer settings for generation tasks (§D.4). Furthermore, we supply explicit data curation procedures and hyperparameter configurations for both ImageNet (§D.5) and text-to-image (§D.6) post-training environments.

## D.1 Representation Encoders

Table 4 summarizes the representation encoders utilized across our experimental settings. For the pooling operations, `cls` denotes the class token, `avg` indicates global average pooling over spatial or patch tokens, and `attn` refers to the attention-based pooling head. During amortizer training, we apply per-dimension normalization to all extracted features. This ensures that the inputs to the amortizer maintain zero mean and unit variance, facilitating stable neural network optimization.

Table 4: Detailed configuration of the representation encoders. We list the model names and architectures, the used weight checkpoints, the feature extraction pooling operations, the input image resolutions, and the resulting feature dimensions.

| Model | Checkpoint | Arch. | Pooling | Resolution | Dim |
|---|---|---|---|---|---|
| Inception-v3 (Szegedy et al., 2016) | torch-fidelity `inception_v3` | CNN | `avg` | 299 | 2048 |
| ConvNeXt-v2 (Liu et al., 2022b) | `convnextv2_base.fcmae_ft_in22k_in1k` | CNN | `avg` | 224 | 1024 |
| SigLIP2 (Tschannen et al., 2025) | `vit_so400m_patch16_siglip_256.v2_webli` | ViT | `cls/attn` | 224 | 1152 |
| MAE (He et al., 2022) | `vit_large_patch16_224.mae` | ViT | `cls` | 224 | 1024 |
| CLIP (Radford et al., 2021) | `vit_large_patch14_clip_224.openai` | ViT | `cls` | 256 | 1024 |
| DINOv2 (Oquab et al., 2023) | `vit_large_patch14_dinov2.lvd142m` | ViT | `cls` | 224 | 1024 |
| PE-Core (Bolya et al., 2026) | `vit_pe_core_large_patch14_336.fb` | ViT | `attn` | 224 | 1024 |
| AIM-v2 (Fini et al., 2025) | `aimv2_huge_patch14_224.apple_pt` | ViT | `avg` | 224 | 1536 |
| Web-SSL (Fan et al., 2025) | `webssl-dino1b-full2b-224` | ViT | `cls` | 224 | 1536 |
| DreamSim (Fu et al., 2023) | DINO + CLIP + OpenCLIP ensemble | ViT | `cls` | 224 | 1792 |

## D.2 Custom JVP Implementation for MLP Amortizer

Evaluating the AMFD objective in Eq. 18 requires frequent JVP computations through the amortization network. In practice, relying on PyTorch's generic forward-mode automatic differentiation (AD) for our MLP amortizer resulted in severe GPU underutilization, emerging as a primary training

**Algorithm 2** Manual JVP Forward Pass for MLP

**Require:** Tangent vector $v$, condition label $c$, time $t$, amortizer $g_\theta$.

**Ensure:** Exact input-direction JVP $\dot{x}_{\text{out}} = \left.\frac{\partial g_\theta(s,t,c)}{\partial s}\right|_{s=0} v$.

$x \leftarrow \mathbf{0}$
$\dot{x} \leftarrow v$
$(x, \dot{x}) \leftarrow \text{LinearJVP}(x, \dot{x}; W_{\text{in}}, b_{\text{in}})$ ▷ $x = W_{\text{in}}x + b_{\text{in}}$, $\dot{x} = W_{\text{in}}\dot{x}$
*% Conditioning path has zero tangent w.r.t. $s$ and is evaluated once.*
$y \leftarrow \text{SiLU}\big(\text{TimeEmbed}(t) + \text{ClassEmbed}(c)\big)$
$(\gamma, \beta, \alpha) \leftarrow \text{AdaLN}_0(y)$
**for** $i = 0, \ldots, N-1$ **do**
  **if** $i > 0$ **and** $i \bmod K = 0$ **then**
    $(\gamma, \beta, \alpha) \leftarrow \text{AdaLN}_{i/K}(y)$ ▷ $K$ is the AdaLN switch frequency
  **end if**
  *% RMSNorm JVP, sharing the primal normalization statistics.*
  $m \leftarrow \text{Mean}(x^2)$
  $r \leftarrow (m + \epsilon)^{-1/2}$
  $x_{\text{n}} \leftarrow x \cdot r$
  $\dot{r} \leftarrow -r^3 \cdot \text{Mean}(x \odot \dot{x})$
  $\dot{x}_{\text{n}} \leftarrow \dot{x} \cdot r + x \cdot \dot{r}$
  **if** RMSNorm has affine weight $w_{\text{n}}$ **then**
    $x_{\text{n}} \leftarrow x_{\text{n}} \odot w_{\text{n}}$
    $\dot{x}_{\text{n}} \leftarrow \dot{x}_{\text{n}} \odot w_{\text{n}}$
  **end if**
  *% AdaLN modulation. Shift has no input tangent.*
  $h \leftarrow x_{\text{n}} \odot (1 + \gamma) + \beta$
  $\dot{h} \leftarrow \dot{x}_{\text{n}} \odot (1 + \gamma)$
  *% SwiGLU MLP JVP.*
  $(u, \dot{u}) \leftarrow \text{LinearJVP}(h, \dot{h}; W_1, b_1)$
  $(u_1, u_2) \leftarrow \text{Chunk}(u)$
  $(\dot{u}_1, \dot{u}_2) \leftarrow \text{Chunk}(\dot{u})$
  $\sigma \leftarrow \text{Sigmoid}(u_1)$
  $s \leftarrow \text{SiLU}(u_1) = u_1 \odot \sigma$
  $\dot{s} \leftarrow \dot{u}_1 \odot \sigma \odot \big(1 + u_1 \odot (1 - \sigma)\big)$
  $z \leftarrow s \odot u_2$
  $\dot{z} \leftarrow \dot{s} \odot u_2 + s \odot \dot{u}_2$
  $(o, \dot{o}) \leftarrow \text{LinearJVP}(z, \dot{z}; W_2, b_2)$
  *% Residual gated update.*
  $x \leftarrow x + o \odot \alpha$
  $\dot{x} \leftarrow \dot{x} + \dot{o} \odot \alpha$
**end for**
*% Final AdaLN + RMSNorm + linear projection.*
$(\gamma_f, \beta_f) \leftarrow \text{FinalAdaLN}(y)$
$(x_{\text{n}}, \dot{x}_{\text{n}}) \leftarrow \text{RMSNormJVP}(x, \dot{x})$
$x \leftarrow x_{\text{n}} \odot (1 + \gamma_f) + \beta_f$
$\dot{x} \leftarrow \dot{x}_{\text{n}} \odot (1 + \gamma_f)$
$(_, \dot{x}_{\text{out}}) \leftarrow \text{LinearJVP}(x, \dot{x}; W_{\text{out}}, b_{\text{out}})$
**return** $\dot{x}_{\text{out}}$

bottleneck. To circumvent this, we implement an exact analytical JVP forward pass tailored specifically to our architecture, as outlined in Algorithm 2. By jointly propagating primal activations and tangent vectors across linear layers, RMSNorm, SiLU/SwiGLU activations, and AdaLN-modulated residual blocks, our custom implementation maximally reuses shared intermediate tensors. This approach strictly preserves the generic AD outputs up to numerical precision while yielding substantial gains in GPU utilization and overall training throughput.

### D.3 AMORTIZER ABLATION DETAILS

**Amortizer architecture details.** The amortizer is parameterized as an AdaLN-conditioned MLP (Li et al., 2024). A grouped AdaLN mechanism is used to minimize conditioning overhead: rather than computing AdaLN parameters at every layer, residual blocks are grouped to share a single modulation projection. Our notation `c[channels]d[depth]a[AdaLN-blocks]` reflects this structure; for instance, `c2048d16a4` applies AdaLN only 4 times across its 16 layers.

**Evaluation metrics and protocol.** To quantify the amortization accuracy, we evaluate its estimates of the conditional mean and the full covariance. Let $\mu_c \in \mathbb{R}^D$ and $\Sigma_c \in \mathbb{R}^{D\times D}$ denote the empirical mean and covariance under condition $c$, and let $\hat{\mu}_c$ and $\hat{\Sigma}_c$ be their respective estimates. We define the mean estimation error as

$$\mathrm{MSE}_\mu = \frac{1}{|\mathcal{C}|D} \sum_{c\in\mathcal{C}} \|\hat{\mu}_c - \mu_c\|_2^2 . \tag{46}$$

To evaluate the covariance estimates, we materialize the complete covariance operator by applying it to the canonical basis vectors. Because the learned operator is not explicitly constrained to be symmetric, we extract its symmetric component

$$\hat{\Sigma}_c^{\mathrm{sym}} = \frac{1}{2}\left(\hat{\Sigma}_c + \hat{\Sigma}_c^\top\right), \tag{47}$$

and define the covariance estimation error as

$$\mathrm{MSE}_\Sigma = \frac{1}{|\mathcal{C}|D^2} \sum_{c\in\mathcal{C}} \left\|\hat{\Sigma}_c^{\mathrm{sym}} - \Sigma_c\right\|_F^2 . \tag{48}$$

For our ImageNet experiments, $\mathcal{C}$ denotes the set of evaluated classes. We compute the mean estimation error $\mathrm{MSE}_\mu$ across all 1,000 classes (i.e., $|\mathcal{C}| = 1000$). However, materializing and evaluating the full $D \times D$ covariance operators across all classes is computationally demanding. Therefore, we evaluate $\mathrm{MSE}_\Sigma$ on a subset of the 100 most populated classes ($|\mathcal{C}| = 100$), each containing exactly 1,300 images. This subset corresponds to class indices 0–42, 44–50, 52–61, 63–97, 99–102, and 104. For both metrics, we report the macro-average over their respective evaluated classes. In contrast, for the unconditional (marginal) experiments, we compute both $\mathrm{MSE}_\mu$ and $\mathrm{MSE}_\Sigma$ directly over the global data distribution, treating the entire dataset as a single entity without class-wise partitioning.

The symmetric covariance error is inherently better aligned with our generator objective than the MSE of the raw, potentially asymmetric operator. Specifically, the covariance term in the generator loss is evaluated through quadratic forms of the type $u^\top(\hat{\Sigma}_g - \hat{\Sigma}_r)u$. For any skew-symmetric matrix $K$, $u^\top K u = 0$; hence, the antisymmetric component of the estimated operator has no impact on this generator objective. Consequently, $\mathrm{MSE}_\Sigma$ precisely measures the covariance component that can influence the generator update, properly avoiding penalties on antisymmetric estimation artifacts that are invisible to the generator loss.

**Training configurations.** Table 5 details the default hyperparameter configurations for these ablation studies.

Table 5: Default hyperparameters for the amortizer ablation studies.

| | |
|---|---|
| representation encoder | Inception-v3 `avg` |
| amortizer architecture | AdaLN MLP |
| amortization variant | AMFD-C (conditional) |
| task conditioning time ($t$) | real: $t = 2$ for mean, $t \in (0, 1)$ for cov; generated: $-t$ |
| optimizer | AdamW, $\beta_1 = 0.9, \beta_2 = 0.95$ |
| learning rate | 1e-4 |
| weight decay | 0 |
| precision | bf16 |
| batch Size | 1024 |
| training steps | 20,000 |
| evaluation metrics | $\mathrm{MSE}_\mu$, $\mathrm{MSE}_\Sigma$ |

### D.4 Default Amortizer Configurations for Image Generation

Unless otherwise specified, the following settings apply universally to all amortizer training procedures:

- **Architecture & initialization:** Parameterized by a `c2048d16a4` MLP with random initialization.
- **Condition injection:** The external conditioning signal is explicitly injected into the amortizer network, corresponding to the AMFD-C configuration.
- **Optimization:** We adopt the same optimizer and global batch size as the generator, applying a constant learning rate of 1e-4 without weight decay.
- **Diffusion parameters:** The interpolation time is fixed at $t = 0.25$. Given the lightweight architecture of the MLP amortizer, we independently draw 4 noise samples per data input to expand the effective batch size and fully utilize hardware computational capacity.
- **Task-conditioning time:** For the real branch, it is set to $t = 2$ for mean prediction and $t \in (0, 1)$ for covariance prediction. The generated branch utilizes the corresponding negative time steps ($-t$).
- **JVP calculation:** We employ a custom analytical implementation for the Jacobian-vector product (JVP), as detailed in Appendix D.2.
- **Multi-representation normalization:** To compute the normalized generator loss (Eq. 20), we consistently fix $\rho = 1$ and $\epsilon = 0.01$, aligning with the multi-representation FD-loss formulation in Yang et al. (2026). Notably, this normalization is bypassed during single-representation training.

For our default MLP amortizers, we utilize the custom analytical JVP implementation. For experiments conducted in native generative spaces where the amortizers adopt Transformer architectures, we opt to employ PyTorch's functional API (`torch.func.jvp` and `vmap`). While our analytical MLP JVP inherently avoids distributed synchronization issues, the PyTorch JVP engine can conflict with the asynchronous communication hooks of Distributed Data Parallel (DDP). To ensure a unified, robust codebase compatible across all amortizer architectures, we systematically structure our distributed training pipeline by wrapping the JVP operation inside the DDP module (*i.e.*, `DDP(jvp(Amortizer))`). This uniform architectural arrangement natively circumvents synchronization failures during forward-mode differentiation, similar to the strategy noted in Lin et al. (2026). Moreover, we empirically observed that computing JVPs through `LayerNorm` leads to severe training instability; `RMSNorm` provides significantly better stability under such operations. While the majority of our adopted amortizer architectures natively employ `RMSNorm`, the SiT (Ma et al., 2024) model used in our native space experiments defaults to `LayerNorm`. To ensure stable training, we explicitly replace all `LayerNorm` layers with `RMSNorm` exclusively within the SiT *amortizer*, while leaving the SiT *generator* entirely unmodified to preserve its exact pretrained formulation. This issue has also been reported by Zhou et al. (2025b) and Lin et al. (2026).

### D.5 ImageNet Post-Training

**Training settings.** Table 6 details the comprehensive hyperparameter configurations for ImageNet post-training, covering both the representation-feature approaches (employing JiT and pMF) and the investigation across various native generative spaces.

**Evaluation.** We measure the model performance using the standard FID alongside the recently introduced FDr$^6$ metric (Yang et al., 2026).

### D.6 Text-to-Image Post-Training

**Training settings.** We evaluate AMFD in both the pixel space using PixelGen (Ma et al., 2026) and the VAE latent space using FLUX.2 [klein] 4B Base (Black Forest Labs, 2026) and FLUX.2 [klein] 4B (Black Forest Labs, 2026). i) To construct the training corpus for the PixelGen experiments, we aggregate subsets from BLIP3o-60k (Chen et al., 2025a) ($58,859$ samples), ShareGPT-4o-Image (Chen et al., 2025b) ($45,717$ samples), and Echo-4o-Image (Ye et al., 2025) (excluding

Table 6: Configurations for ImageNet-256×256 post-training. The top section details post-training using representation features, while the bottom section specifies settings for four native generative spaces (pixel, SD-VAE (Rombach et al., 2022), VA-VAE (Yao et al., 2025), and RAE (Zheng et al., 2025)). For native spaces, we adopt JiT (Li & He, 2026), SiT (Ma et al., 2024), LightningDiT (Yao et al., 2025), and DiT$^{\text{DH}}$ as architectures for both the generator and the amortizer.

| *Representation feature spaces* | **JiT** | **pMF** |
|---|---|---|
| model sizes | B, L, H | B, L, H |
| initialization | official pretrained weights | |
| AMFD-Incep. | Inception-v3 `avg` | |
| AMFD-SIM | SigLIP2 `cls` + Inception-v3 `avg` + MAE `cls` | |
| batch size | 1024 | |
| optimizer | AdamW (Loshchilov & Hutter, 2017), $\beta_1 = 0.9, \beta_2 = 0.95$ | |
| weight decay | 0 | |
| learning rate | 1e-5 | 1e-6 |
| lr schedule | cosine | |
| total epochs (Table 2) | 100 | |
| total epochs (Figure 6 & 7) | 20 | |
| warmup epochs | 1 | |
| precision | bf16 mixed | |
| dropout | 0 | |
| augmentation | center crop, horizontal flip | |
| EMA decay | EDM2-style (Karras et al., 2024) | |

| *Native generative spaces* | **Pixel** | **SD-VAE** | **VA-VAE** | **RAE** |
|---|---|---|---|---|
| generator model | JiT-L/16 | SiT-XL/2 | LightningDiT-XL/1 | DiT$^{\text{DH}}$-XL/2 |
| amortizer model | JiT-B/16 | SiT-B/2 | LightningDiT-B/1 | DiT$^{\text{DH}}$-B/2 |
| initialization | official pretrained weights | | | |
| batch size | 1024 | | | |
| optimizer | AdamW (Loshchilov & Hutter, 2017), $\beta_1 = 0.9, \beta_2 = 0.95$ | | | |
| weight decay | 0 | | | |
| learning rate | 1e-5 | | | |
| lr schedule | cosine | | | |
| total epochs (Figure 8) | 20 | | | |
| warmup epochs | 1 | | | |
| precision | bf16 mixed | | | |
| dropout | 0 | | | |
| augmentation | center crop, horizontal flip | | | |
| EMA decay | EDM2-style (Karras et al., 2024) | | | |
| amortizer noise samples | 1 | | | |
| amortizer JVP calculation | `torch.func.jvp` | | | |

the multi-reference split, $105,506$ samples). This curation yields a total of $210,082$ training samples. During post-training, the additional transformer blocks introduced for text conditioning remain frozen. ii) For the FLUX.2 [klein] 4B models, our data preparation protocol and the selection of the ten representation encoders generally follow the settings introduced by iRDM (Feng et al., 2026). However, as DINOv3 is currently inaccessible, we substitute DINOv3-L with DINOv2-L. The reference dataset is partitioned into two distinct subsets: a *perception block* based on natural COCO captions, and a *composition block* based on detector-verified GenEval prompts. All reference images are generated offline at a $512 \times 512$ resolution using the four-step FLUX.2 [klein] 4B teacher. Unlike iRDM, which truncates captions to $48$ tokens, we preserve the original $512$ text token length of FLUX.2 [klein]. For the perception block, we generate $24$ candidates for each of the $82,783$ COCO `train2014` captions and retain the top three scored by PickScore. This exactly matches the iRDM dataset scale, yielding $248,349$ samples. For the composition block, given the $553$ GenEval prompts, we sample $150$ seeds (expanding up to $1,000$ seeds if necessary) to retain up to $100$ detector-verified generations per prompt. Notably, two specific prompts (one targeting position and one color attribute) yielded zero valid samples even after exploring $1,000$ seeds. Consequently, our final composition block comprises $53,504$ samples, closely approximating iRDM's $53,800$. In total, the FLUX.2 [klein] training corpus contains $301,853$ samples. The complete training configurations for PixelGen and FLUX.2 [klein] experiments are summarized in Table 7.

Table 7: Hyperparameter configurations for text-to-image post-training. The top section details post-training in pixel space (PixelGen), while the bottom section specifies settings for VAE space (FLUX.2 [klein]).

| *Pixel-space generation* | **PixelGen** | |
|---|---|---|
| original sampling steps | $25 \times 2 \times 2$ | |
| initialization | official checkpoint | |
| base resolution | $512 \times 512$ | |
| trainable parameters | image generator only | |
| text encoder (frozen) | Qwen3-1.7B | |
| text tokens | 128 | |
| AMFD-SIM | SIM (Inception `avg`, SigLIP2 `cls`, MAE `cls`) | |
| training data | public datasets (§D.6) | |
| batch size | 1024 | |
| optimizer | AdamW (Loshchilov & Hutter, 2017), $\beta_1 = 0.9, \beta_2 = 0.95$ | |
| learning rate | 1e-5 | |
| total steps | 5000 | |
| warmup steps | 500 | |
| precision | bf16 | |
| EMA decay | none | |
| amortizer architecture | `c2048d16a4` MLP w/ 8-head cross-attn per adaLN | |
| *Latent-space generation* | **FLUX.2 [klein] 4B Base** | **FLUX.2 [klein] 4B** |
| original sampling steps | $50 \times 2$ | 4 |
| initialization | official pretrained weights | |
| base resolution | $512 \times 512$ | |
| trainable parameters | image generator only | |
| text encoder (frozen) | Qwen3-4B | |
| text tokens | 512 | |
| AMFD-SIM | SIM (Inception `avg`, SigLIP2 `cls`, MAE `cls`) | |
| AMFD-10 enc. | 10 encoders in Table 4, SigLIP2 using `attn` | |
| training data | prepared following Feng et al. (2026) | |
| batch size | 1024 | |
| optimizer | AdamW (Loshchilov & Hutter, 2017), $\beta_1 = 0.9, \beta_2 = 0.95$ | |
| learning rate | 5e-6 | |
| total steps | 1500 | |
| warmup Steps | 150 | |
| precision | bf16 | |
| EMA decay | none | |
| amortizer architecture | `c2048d16a4` MLP w/ 8-head cross-attn per adaLN | |

**Evaluation.** We evaluate our models on the GenEval (Ghosh et al., 2023) and PickScore (Kirstain et al., 2023) benchmarks. For PickScore, we report the mean score over the 499 prompts from the Pick-a-Pic test set. Regarding the baseline results in Table 3, the GenEval metrics for FLUX.2 [klein] 4B and its post-training methods DMD2 and iRDM are directly cited from the iRDM paper (Feng et al., 2026). For PickScore, since the specific prompt set evaluated in their paper is not publicly available, we directly cite the baseline scores reported in their official codebase, which are based on the Pick-a-Pic test set. The PickScore for DMD2 is left blank, as it is not available in the repository.

## E Additional Metrics on ImageNet Post-training

Table 8 reports the FDr (Yang et al., 2026) metrics evaluated across six diverse pre-trained representation encoders for ImageNet post-training.

## F Additional Visualizations

**Class-conditional generation on ImageNet.** We provide additional generated samples on ImageNet 256×256 using post-trained JiT-H/16 and pMF-H/16 in Figure 9 and Figure 10, respectively. In each figure, the columns from left to right display samples from base models post-trained with

Table 8: Additional FDr metrics for one-step post-training on ImageNet-256×256. Lower is better for all metrics.

| Method | Repr. | FDr ↓ Inception | ConvNeXt | DINOv2 | MAE | SigLIP | CLIP | FID ↓ | $FDr^6$ ↓ |
|---|---|---|---|---|---|---|---|---|---|
| 50K validation images | | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.68 | 1.00 |
| JiT-B/16 | - | - | - | - | - | - | - | 3.71 | 15.65 |
| + FD-loss | SIM | - | - | - | - | - | - | 1.00 | 5.53 |
| + AMFD-C | SIM | 0.56 | 1.14 | 5.38 | 2.16 | 5.77 | 13.47 | 0.95 | 4.75 |
| + AMFD-U | SIM | 0.56 | 1.19 | 4.64 | 1.28 | 3.26 | 12.54 | 0.95 | 3.91 |
| JiT-L/16 | - | 1.54 | 3.49 | 6.10 | 8.07 | 19.37 | 25.82 | 2.59 | 10.73 |
| + FD-loss | SIM | - | - | - | - | - | - | 0.77 | 3.24 |
| + AMFD-C | SIM | 0.52 | 0.88 | 2.73 | 0.62 | 3.33 | 7.94 | 0.87 | 2.67 |
| + AMFD-U | SIM | 0.51 | 0.92 | 2.42 | 0.31 | 1.94 | 6.03 | 0.85 | 2.02 |
| JiT-H/16 | - | 1.18 | 2.52 | 4.28 | 5.65 | 11.91 | 20.40 | 1.97 | 7.66 |
| + FD-loss | SIM | **0.45** | 0.86 | 2.10 | 0.43 | **1.68** | 10.37 | **0.75** | 2.65 |
| + AMFD-C | SIM | 0.51 | 0.84 | 2.12 | 0.38 | 2.73 | 6.30 | 0.85 | 2.15 |
| + AMFD-U | SIM | 0.49 | **0.79** | **1.88** | **0.21** | 1.73 | **5.65** | 0.83 | **1.79** |
| pMF-B/16 | - | - | - | - | - | - | - | 3.31 | 13.70 |
| + FD-loss | SIM | - | - | - | - | - | - | 0.85 | 3.50 |
| + AMFD-C | SIM | 0.57 | 0.81 | 4.34 | 2.20 | 6.98 | 8.78 | 0.95 | 3.94 |
| + AMFD-U | SIM | 0.55 | 0.83 | 3.87 | 1.69 | 5.80 | 7.86 | 0.92 | 3.43 |
| pMF-L/16 | - | 1.62 | 1.36 | 6.70 | 9.72 | 20.34 | 14.81 | 2.72 | 9.09 |
| + FD-loss | SIM | **0.47** | 0.57 | 2.21 | 0.56 | **3.03** | 5.68 | **0.78** | 2.09 |
| + AMFD-C | SIM | 0.52 | **0.53** | 2.38 | 0.67 | 4.07 | 5.35 | 0.88 | 2.25 |
| + AMFD-U | SIM | 0.51 | 0.55 | **2.14** | **0.47** | 3.36 | **5.03** | 0.86 | **2.01** |
| pMF-H/16 | - | 1.37 | 1.15 | 5.43 | 6.25 | 15.33 | 11.68 | 2.29 | 6.87 |
| + FD-loss | SIM | **0.46** | **0.57** | **1.74** | 0.35 | **2.46** | 5.77 | **0.77** | 1.89 |
| + AMFD-C | SIM | 0.51 | 0.58 | 1.91 | 0.41 | 3.47 | 4.71 | 0.86 | 1.93 |
| + AMFD-U | SIM | 0.50 | 0.60 | **1.74** | **0.27** | 2.86 | **4.54** | 0.85 | **1.75** |

FD-loss, our conditional AMFD (AMFD-C), and our unconditional AMFD (AMFD-U). To ensure a direct visual comparison, identically positioned samples across the three columns are generated using the exact same initial noise.

**Text-to-image generation.** Figure 11 showcases additional text-to-image samples produced by the FLUX.2 [klein] 4B model post-trained with our AMFD loss. Ordered column by column from left to right (and top to bottom within each column), the corresponding text prompts are as follows:

(1) A striking close-up portrait of a young woman with vitiligo wearing a simple cobalt silk blouse, natural curls framing her face, confident calm expression, pale peach studio background, soft beauty lighting, truthful unretouched skin detail.

(2) A large bold word AMFD painted in glowing anime bubble letters across the center of a night festival stage, fireworks behind it, colorful spotlights, dynamic celebratory composition.

(3) A silver electric concept coupe parked on a mirror-like salt flat at blue hour, seamless aerodynamic body, thin white running lights, distant mountains doubled in shallow water, minimal luxury automotive photography.

(4) A poetic close-up portrait of a young adult Chinese woman standing among tall silver grass on a coastal hill, cream scarf, hair lifted by sea wind, overcast horizon softened behind, introspective expression, muted cinematic color and fine film grain.

(5) A sculptural ceramic vase filled with one branch of white magnolia, placed on a travertine pedestal against a muted terracotta wall, clean side light and subtle shadows, refined design editorial.

(6) A macro still life of a cracked blue ceramic bowl repaired with gold kintsugi, placed on dark linen with white flowers.

(7) A delicate watercolor-style scene of a sailboat crossing a lake at sunrise, reeds, ducks, pale orange sky, calm composition.

(8) A tiny hummingbird hovering beside a red trumpet flower, iridescent throat flashing emerald and magenta, wings rendered as delicate motion, dew-covered garden in soft morning bokeh, macro wildlife photograph.

(9) A curious alpaca standing at the edge of a high Andean village, woven red tassel on its halter, terraced green mountains and low clouds behind, bright documentary travel photograph.

(10) A secluded tropical cove viewed through limestone arches, luminous aquamarine water, a narrow white beach and one wooden boat, humid morning haze, realistic Southeast Asian travel photograph.

(11) A jewel-toned fruit tart beneath a glass bakery dome, concentric raspberries, kiwi and mango with a mirror glaze, soft reflections on a walnut counter, elegant close food photography.

(12) A proud Friesian horse galloping along a wide Atlantic beach, black mane streaming, hooves scattering silver surf, overcast sky and distant dunes, dramatic equine photography.

(13) A massive green word FRESH placed above a farm stand with fruit crates, morning sunlight, clean commercial photography.

(14) A red electric motorcycle parked in a narrow alley after rain, neon signs reflected in wet pavement, steam rising from vents, realistic urban night photography, low angle lens.

(15) A delicate close-up portrait of a young Chinese ink painter in a bright studio, subtle charcoal marks on one fingertip and cheek, long dark hair loosely tied, rice paper bokeh, introspective gaze, soft high-key natural photography.

(16) A small blue delivery van parked beside a quiet seaside road under a bright but soft afternoon sky. The van has rounded corners, clean windows, slightly worn tires, and a simple roof rack carrying two neatly tied wooden crates. The road curves gently along the coast, with a low stone barrier, pale grass, and calm water visible beyond it. A few distant sailboats appear near the horizon, and the sunlight gives the van a mild highlight along its side panel. The composition should feel clear and pleasant, like a travel photograph from a slow coastal route. Focus on the van's shape, paint surface, glass reflections, tire texture, road markings, stone barrier, sea color, and open air. The mood should be relaxed, bright, and realistic, with simple forms and a clean sense of place. The camera frames the van from a relaxed roadside angle, giving the rounded body, roof rack, crates, tires, glass, road curve, stone barrier, grass, sea, and horizon a clear readable structure. Colors stay bright and gentle, with blue paint, pale road, warm grass, soft sky, and calm water creating a clean coastal mood.

(17) A cinematic portrait of a weathered polar expedition pilot inside an icy cockpit, frost on the glass, amber instrument lights reflecting on the face, shallow depth of field, realistic skin texture, dramatic cold atmosphere.

(18) A green sea turtle drifting above a colorful coral garden, small reef fish surrounding its shell, clear sunbeams descending from the surface, richly detailed but natural underwater photography.

(19) A black cat sitting on the hood of a vintage cream convertible parked beside a lavender field, sunset sky, chrome reflections, playful fashion campaign mood.

(20) The Namib Desert at sunrise seen from a high dune, sweeping orange ridges casting blue shadows, a line of gemsbok tracks crossing untouched sand, minimal and precise fine-art landscape photograph.

(21) A sunlit close-up portrait of a weathered Greek fisherman at a harbor, white beard, knitted navy cap, deep smile lines, turquoise boat paint softly visible behind him, direct honest gaze, Mediterranean color, high-detail environmental portrait.

(22) A colorful cereal box on a breakfast table, the front label clearly reads PIXEL CRUNCH in playful letters, morning sunlight, realistic product photography.




Figure 9: Uncurated samples from JiT-H post-trained with FD-loss (left), our AMFD-C (middle), and AMFD-U (right), generated using identical initial noise.

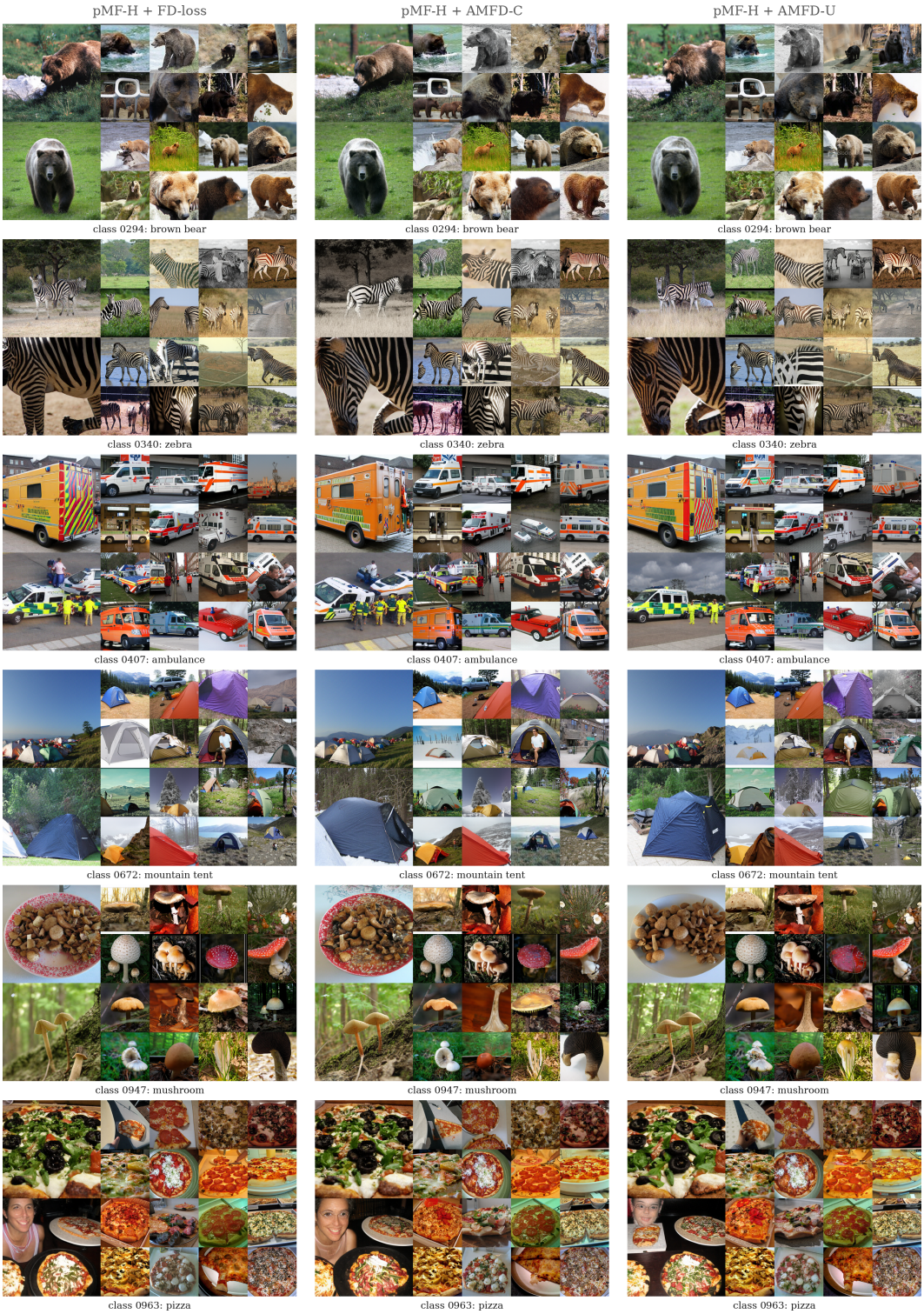


Figure 10: Uncurated samples from pMF-H post-trained with FD-loss (left), our AMFD-C (middle), and AMFD-U (right), generated using identical initial noise.

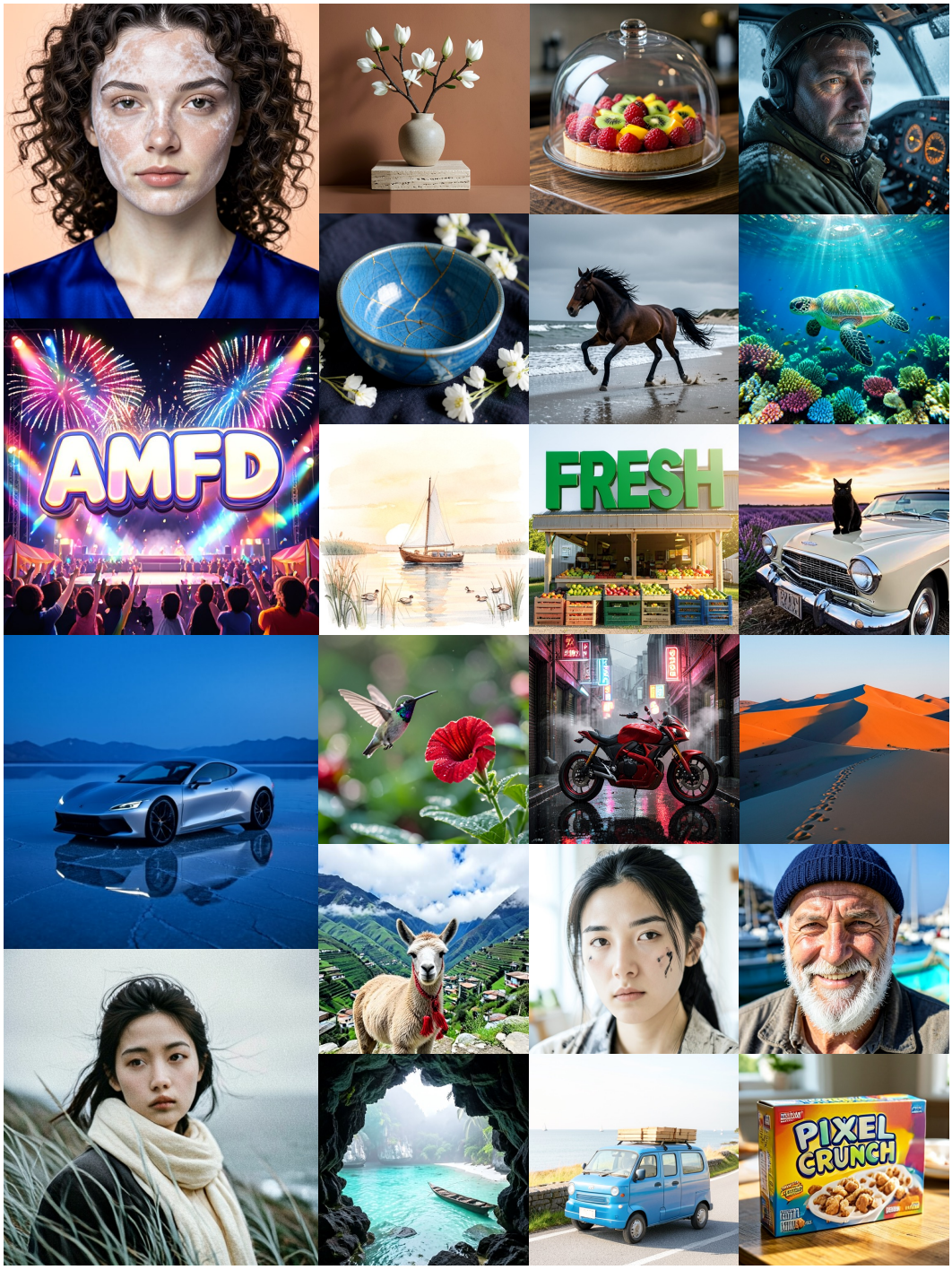


Figure 11: Additional text-to-image samples generated by the FLUX.2 [klein] 4B model post-trained with AMFD.