UNIFYING DISTRIBUTIONAL TRAINING FOR ONE-STEP VISUAL GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: **One-step text-to-image samples.** FLUX.2 [klein] 4B post-trained with MGFlow.

ABSTRACT

Distributional training provides collective supervision for one-step visual generation by matching real and generated features in frozen representation spaces. We introduce *a unified theoretical framework* that separates distribution modeling from matching discrepancy and connects global objectives to pointwise feature updates through Wasserstein gradient flow. Under this framework, FD-Loss and Gaussian-kernel Drifting are recovered through Gaussian optimal transport and kernel-density-based KL matching respectively. The framework motivates **MGFlow**, which models feature distributions with Gaussian mixtures at an adjustable granularity between global moments and sample-based representations. MGFlow supports both optimal transport and score based matching, and couples mass-constrained sample assignment with paired component updates to address mode collapse that mixture expressivity alone does not resolve. On ImageNet 256×256, MGFlow substantially surpasses the FD-Loss baseline, achieving state-of-the-art results with **1.45** FDr^6 on pMF-H and **1.64** on JiT-H. For text-to-image generation, MGFlow post-trains FLUX.2 [klein] 4B into a one-step generator that outperforms the original four-step model on both GenEval and PickScore. Project page: `https://mgflow-repo.github.io`.

1 INTRODUCTION

One-step visual generation aims to synthesize high-quality images with a single network evaluation, avoiding iterative refinement at inference. Existing approaches include learning consistent trajectory

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.

endpoints (Song et al., 2023; Song & Dhariwal, 2024; Kim et al., 2024; Luo et al., 2023; Geng et al., 2025b; Lu & Song, 2025), average velocities (Frans et al., 2025; Geng et al., 2025a; 2026; Lu et al., 2026), or distilling pretrained diffusion models (Salimans & Ho, 2022; Yin et al., 2024b; Sauer et al., 2024; Zhou et al., 2024; Yin et al., 2024a). A prominent recent direction is *distributional training* in frozen representation spaces (Yang et al., 2026a; Deng et al., 2026; Feng et al., 2026). Rather than assigning a fixed target image to each output, these methods compare collections of real and generated features and obtain *collective supervision* from their distributional mismatch. Each feature's gradient depends on shared statistics or interactions with other features, rather than only on an individual reconstruction target. This approach raises two fundamental questions: *how should feature distributions be modeled, and how should their mismatch guide learning?*

We introduce *a unified theoretical framework for distributional training* that separates the distribution model from the matching discrepancy. Sampling and frozen encoders determine the feature populations being compared; the distribution model and matching discrepancy determine what distributional information is recorded and how it is aligned. Wasserstein gradient flow (WGF) (Jordan et al., 1998; Peyré & Cuturi, 2019) act as the theoretical bridge that relates the distributional discrepancy to a pointwise velocity. This establishes a common formulation for global moment losses and local sample interactions, *with existing methods recovered as specific model–discrepancy choices.*

For example, a Gaussian distribution model with an optimal transport (OT) based W_2 distance recovers *FD-Loss* (Yang et al., 2026a). Let p_{G} and q_{G} be Gaussians with the respective means and covariances of the real and generated features. FD-Loss optimizes:

$$\mathcal{D}_{\mathrm{FD}}(q_{\mathrm{G}}, p_{\mathrm{G}}) = W_2^2(q_{\mathrm{G}}, p_{\mathrm{G}}) = \|\mu_q - \mu_p\|_2^2 + \mathrm{Tr}\left(\Sigma_q + \Sigma_p - 2\left(\Sigma_q^{1/2}\Sigma_p\Sigma_q^{1/2}\right)^{1/2}\right). \quad (1)$$

Thus, the Fréchet discrepancy between global feature statistics is a Gaussian transport cost (Dowson & Landau, 1982; Peyré & Cuturi, 2019). At the population level, differentiating through the moments moves features along the associated affine transport field. The Gaussian model determines which statistics are retained, while the OT discrepancy determines how they guide the movement of generated features.

A sample-based density model with Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951) instead recovers *Gaussian-kernel Drifting*. Drifting (Deng et al., 2026) is specified through kernel-weighted attraction and repulsion. For real and generated feature distributions p and q_θ, a positive kernel k, and $r \in \{p, q_\theta\}$, its mean-shift and update fields are defined respectively as

$$a_r(z) = \frac{\mathbb{E}_{y\sim r}[k(z,y)(y-z)]}{\mathbb{E}_{y\sim r}[k(z,y)]}, \quad v_{\text{drift}}(z) = a_p(z) - a_{q_\theta}(z). \quad (2)$$

The generator regresses samples toward detached targets shifted by this field v_{drift}. For a Gaussian kernel of bandwidth h, each mean shift satisfies $a_r(z) = h^2\nabla_z \log r_h(z)$, where r_h is the Gaussian kernel density estimate (KDE) of r (Lai et al., 2026). Their difference is therefore the density-level KL velocity evaluated at those KDEs, up to a bandwidth-dependent scale (Cao et al., 2026).

Built on top this theoretical framework, we propose **Mixture Gradient Flow (MGFlow)**. To refine the modeling granularity while capturing global correlations, MGFlow uses Gaussian mixtures (GMs) (Delon & Desolneux, 2020), which retain componentwise means and covariances at an adjustable resolution between global moment matching and sample-centered KDE. We develop both OT-based and score-based matching for this representation. Crucially, we note that an expressive modeling does not itself ensure mode coverage or correct transportation (Wenliang & Kanagawa, 2020): naive posterior assignment and transport may suffer from weight mismatch and mode collapse. MGFlow therefore couples mass-constrained sample assignment with paired component matching, using reference weights to control the mass of each generated component and a shared component correspondence to guide features toward the matched reference components.

We evaluate MGFlow under the same sampling and feature encoder settings as FD-Loss. On ImageNet (Russakovsky et al., 2015) 256×256, MGFlow achieves state-of-the-art $\mathbf{1.45}$ FDr^6 (Yang et al., 2026a) on pMF-H (Lu et al., 2026) and $\mathbf{1.64}$ on JiT-H (Li & He, 2026), improving over the FD-Loss baseline with **23%** and **38%** margins, respectively. For text-to-image generation, MGFlow post-trains FLUX.2 [klein] 4B (Black Forest Labs, 2026) for one-step inference, reaching $\mathbf{0.900}$ GenEval (Ghosh et al., 2023) and $\mathbf{21.98}$ PickScore (Kirstain et al., 2023), outperforming all previous methods. Together with comparisons across model–discrepancy combinations, these results show how our novel formulation leads to effective alternatives to existing training prescriptions.

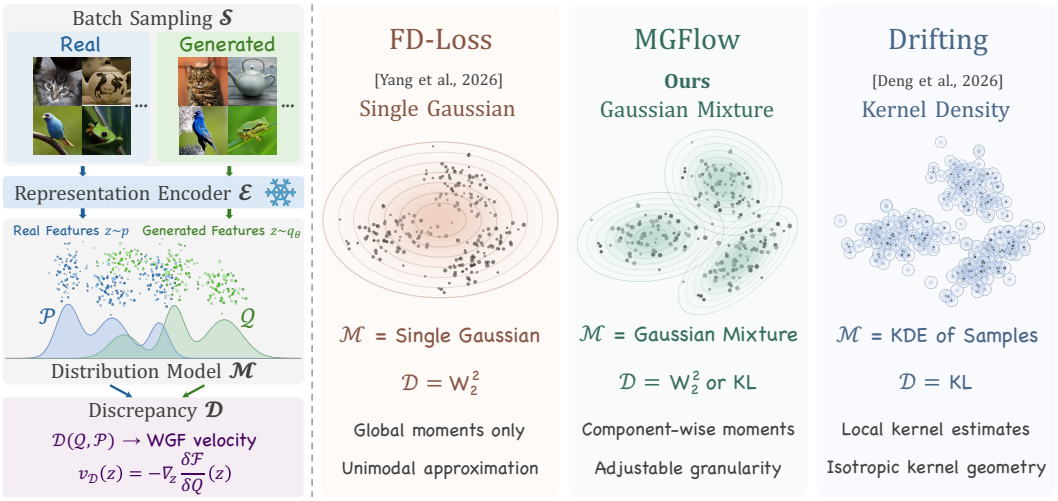


Figure 2: **A unified view of distributional training.** Left: sampling $\mathcal{S}$, encoding $\mathcal{E}$, distribution modeling $\mathcal{M}$, and matching discrepancy $\mathcal{D}$ define the paradigm. Right: FD-Loss, MGFlow, and Gaussian-kernel Drifting use Gaussian, GM, and KDE representations, respectively. Their matching objectives induce feature-space velocity fields through the Wasserstein gradient flow of their respective distributional objectives.

2 A Unified View of Distributional Training

A collective feature-space distributional training paradigm. We formulate *distributional training* by $(\mathcal{S}, \mathcal{E}, \mathcal{M}, \mathcal{D})$ (Figure 2): the sampling scheme $\mathcal{S}$ supplies sample collections that provide collective supervision, the frozen encoder $\mathcal{E}$ maps samples to representation spaces where they are compared. For real and generated feature distributions p and q_θ, the distribution model $\mathcal{M}$ constructs parametric approximations P and Q, and the discrepancy $\mathcal{D}$ measures the mismatch. This construction provides distributional supervision: each feature update depends on the entire sampled population through the distribution model, rather than only on an individual target.

Table 1: **Distributional training** with distribution models and discrepancies. All methods train pMF-B with Inception features for 10 epochs. All methods use the aligned recipe in Appendix E.1.

Model	Method	FDr^6↓
OT-based (W_2)		
Gaussian	FD-Loss (Yang et al., 2026a)	13.05
GM $K = 4$	MGFlow-W_2	**11.33**
Sample-based	W-Flow (Han et al., 2026)	12.58
Score-based (KL)		
Gaussian	Gaussian KL	12.73
GM $K = 16$	MGFlow-KL	**11.32**
Sample-based	Gaussian-kernel Drifting (Deng et al., 2026)	12.29

FD-Loss and Drifting provide collective supervision through shared moments or sample interactions rather than independent reconstruction targets (Yang et al., 2026a; Deng et al., 2026). FD-Loss is trained in the feature space of three independent encoders, while drifting with a latent MAE encoder. By default, we adopt FD-Loss's sampling scheme $\mathcal{S}$ and encoders $\mathcal{E}$, and focus on the distribution model and discrepancy $(\mathcal{M}, \mathcal{D})$.

A global discrepancy induces a pointwise descent field. To compare loss-based and field-based training prescriptions, we need to connect a scalar distributional mismatch to an update direction for each generated feature. The Wasserstein gradient flow (Jordan et al., 1998; Peyré & Cuturi, 2019) provides this connection by describing how feature locations should move to decrease the chosen discrepancy. Fix P and let $\mathcal{F}(Q) = \mathcal{D}(Q, P)$, where Q_t is the evolving generated distribution through training time t. Let $\delta\mathcal{F}/\delta Q$ denote the first variation of the objective, describing its sensitivity to infinitesimal density changes. The 2-WGF defines descent $v_{\mathcal{D},t}$ using mean-squared point displacement. Under suitable regularity, the velocity and energy dissipation satisfy:

$$v_{\mathcal{D},t}(z) = -\nabla_z \frac{\delta \mathcal{F}}{\delta Q}(z)\Big|_{Q=Q_t}, \quad \frac{\mathrm{d}z_t}{\mathrm{d}t} = v_{\mathcal{D},t}(z_t), \quad \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}(Q_t) = -\mathbb{E}_{z\sim Q_t}\left[\|v_{\mathcal{D},t}(z)\|_2^2\right] \le 0. \quad (3)$$

Thus *moving points along the induced field decreases global discrepancy*. Details in Appendix B.5.

FD-Loss and Drifting are special cases of distributional training. We now instantiate this construction with specific distribution models and discrepancies to recover the feature-update fields of FD-Loss and Gaussian-kernel Drifting. For $\mathcal{M}$, let $p_{\mathrm{G}} = \mathcal{N}(\mu_p, \Sigma_p)$ and $q_{\mathrm{G}} = \mathcal{N}(\mu_q, \Sigma_q)$ be moment-matched Gaussian models with positive-definite covariances. Alternatively, let p_h, q_h be Gaussian kernel density estimates (KDEs) of p, q_θ using samples centered kernels with covariance $h^2 I$ (Parzen, 1962). For $\mathcal{D}$, we consider an optimal transport (OT) approach using the 2-Wasserstein distance W_2 and a score based approach with the Kullback-Leibler divergence D_{KL}; these choices recover the FD-Loss and Gaussian-kernel Drifting fields (Lai et al., 2026; Cao et al., 2026):

$$\begin{aligned} \tfrac{1}{2} W_2^2(q_{\mathrm{G}}, p_{\mathrm{G}}) &\xrightarrow{\text{WGF}} T_{q_{\mathrm{G}} \to p_{\mathrm{G}}}(z) - z = v_{\text{FD}}(z), \\ D_{\mathrm{KL}}(q_h \| p_h) &\xrightarrow{\text{WGF}} \nabla_z \log p_h(z) - \nabla_z \log q_h(z) = h^{-2} v_{\text{drift}}(z). \end{aligned} \tag{4}$$

where $T_{q_{\mathrm{G}} \to p_{\mathrm{G}}}$ is the closed-form optimal transport map between Gaussians (Peyré & Cuturi, 2019). The Fréchet distance equals W_2^2, and differentiating it through feature moments moves samples along the first descent field. Gaussian-kernel Drifting instead evaluates the KL velocity on the smoothed densities. The two methods occupy global-moment and sample-local ends of the modeling spectrum, with different discrepancies. Appendices B.1–B.2 give proofs and derivations.

Other methods could also fit within this framework. For instance, calculating an OT based discrepancy on empirical measures gives W-Flow (Han et al., 2026), which construct optimal transport between batches with Sinkhorn (Cuturi, 2013; Feydy et al., 2019) approximation. Moreover, the framework can inspire new distributional training algorithms. Keeping the single Gaussian surrogate but replacing W_2 with KL divergence gives the following closed-form velocity field:

$$v_{\mathrm{KL}}(z) = \Sigma_p^{-1}(\mu_p - z) - \Sigma_q^{-1}(\mu_q - z) \tag{5}$$

which we denote as Gaussian KL. We conduct experiments with all methods mentioned above under the same settings, using the same sampling scheme with FD-Loss and frozen Inception (Szegedy et al., 2016) encoder. Results are in Table 1. All alternatives outperform FD-Loss on FDr^6, *even without direct optimization of the Fréchet distance*, demonstrating the versatility of our formulation.

3 MGFLOW

As shown in Section 2, FD-Loss and Gaussian-kernel Drifting take two extremes on the distribution model spectrum: A single Gaussian models only global first and second moments, whereas Gaussian KDE uses sample-centered components and overlooks global covariance structures. In order to improve modeling granularity while capturing global correlations in feature spaces, we introduce **Mixture Gradient Flow (MGFlow)**, a distributional training algorithm that adopts an intermediate distribution modeling $\mathcal{M}$ by a Gaussian Mixture (GM). We first introduce Gaussian-mixture representations and direct OT- and KL-based matching constructions. We then explain why representing multiple modes is not sufficient, and develop a coupled allocation-and-update procedure that preserves correspondence with the reference components, while still bounding the global discrepancies.

3.1 FROM A SINGLE GAUSSIAN TO GAUSSIAN MIXTURES

For the distribution models P, Q, consider using Gaussian Mixtures with K Gaussian components:

$$P(z) = \sum_{k=1}^{K} \pi_k p_k(z), \quad Q(z) = \sum_{k=1}^{K} \omega_k q_k(z), \tag{6}$$

where $p_k = \mathcal{N}(\mu_{p,k}, \Sigma_{p,k})$ and $q_k = \mathcal{N}(\mu_{q,k}, \Sigma_{q,k})$. P is fit offline by the EM algorithm (Dempster et al., 1977) and remains fixed, while Q summarizes the changing generated feature distribution, which requires updating through training. *The case $K = 1$ recovers a single Gaussian; sample-centered components with a common isotropic covariance ($K = B$) recover Gaussian KDE at the representation level.* Table 1 summarizes the model–discrepancy combinations.

Both discrepancies, W_2 and KL, extend to this representation. Though Wasserstein Distances between GMs are intractable, restricting transport to Gaussian component pairs gives a tractable upper bound, namely the Mixed Wasserstein distance(Delon & Desolneux, 2020):

$$W_2^2(Q, P) \le MW_2^2(Q, P) := \min_{\Gamma \in U(\omega, \pi)} \sum_{i,j} \Gamma_{ij} W_2^2(q_i, p_j). \tag{7}$$

Here $U(\omega, \pi)$ is the set of component couplings with marginals ω and π, and each pair cost is the Gaussian FD. For KL, the marginal mixture field is the difference of the score functions:

$$v_{\text{global}}(z) = \nabla_z \log P(z) - \nabla_z \log Q(z) = \sum_k \gamma_{P,k}(z) s_{p,k}(z) - \sum_k \gamma_{Q,k}(z) s_{q,k}(z), \quad (8)$$

where $s_{r,k}(z) = \nabla_z \log r_k(z) = \Sigma_{r,k}^{-1}(\mu_{r,k} - z)$ for $r \in \{p, q\}$, $\gamma_{P,k} = \pi_k p_k / P$, and $\gamma_{Q,k} = \omega_k q_k / Q$. Each marginal score combines component scores using its posterior responsibilities.

However, representing multiple modes does not ensure correct mode proportions. Marginal KL updates can be weak between well-separated modes. Consider $P = \sum_k \pi_k r_k$ and $Q = \sum_k \omega_k r_k$ with the same separated components r_k but different positive weights. Correcting this weight mismatch requires movement between modes, not merely local refinement within each mode. However, in a region dominated by component k, the density ratio is approximately the constant ω_k/π_k, the score difference can be small even when the component weights differ substantially:

$$\log \frac{Q(z)}{P(z)} \approx \log \frac{\omega_k}{\pi_k}, \quad v_{\text{global}}(z) = -\nabla_z \log \frac{Q(z)}{P(z)} \approx 0 \quad (9)$$

Thus, a mismatch can remain visible in the density ratio while producing only a weak feature-update signal in the high-density regions. This is the score-blindness mechanism studied by Wenliang & Kanagawa (2020), and does not contradict the descent interpretation in Section 2: a decreasing objective need not provide strong inter-mode movement. A demonstration is shown in Figure 3. Same issue can arise for Gaussian-KDE fields when smoothing leaves modes well separated. For score-based matching, *mixture expressivity does not by itself provide an effective mechanism for correcting mode proportions.* We therefore connect mass allocation explicitly to the feature updates.

3.2 Mass Constrained Allocation and Paired Transport

LP based component allocation. The reference mixture P is fitted offline and kept fixed, whereas Q must track the changing generator throughout training. For a single Gaussian, each generated batch contributes to one set of global moments. A GM instead requires component-specific moments, so updating Q requires allocating new features to components. A natural choice is posterior soft assignment: each generated sample z_n contributes to component k with weight $\gamma_{Q,k}(z_n) = \omega_k q_k(z_n)/Q(z_n)$, its posterior responsibility under the current Q. However, due to dynamical reasons, posterior soft assignment cannot transport the correct amount of mass to each component, as shown in Figure 3.

Table 2: **Ablation on component assignment and transportation.** JiT-B trains in Inception feature space for 10 epochs with $K = 4$. Timing uses 8 H200 GPUs.

	KL		W_2	
	FDr$^6\downarrow$	s/step	FDr$^6\downarrow$	s/step
Posterior	173.56	**0.197**	26.78	0.699
LP-global	146.92	0.264	23.73	0.825
LP-paired	**23.27**	0.263	**23.42**	**0.392**

We instead assign generated features to the fixed reference components p_k while constraining the mass allocated to component k to $B\pi_k$ for a batch of B features. To achieve this, we solve the capacity-constrained linear program (LP) for batch-to-component allocation:

$$R^* \in \arg\min_{R \geq 0} \sum_{n,k} R_{nk}[-\log(\pi_k p_k(z_n))], \quad \text{subject to} \quad R\mathbf{1}_K = \mathbf{1}_B, \quad R^\top \mathbf{1}_B = B\pi. \quad (10)$$

Here $R \in \mathbb{R}^{B \times K}$, where R_{nk} denote the assignment from sample n to component k, which may be fractional. The likelihood cost favors compatible reference components, while the capacity constraints enforce their prescribed masses. Using detached R^*, we compute the weighted mean and raw second moment for each generated component, maintain these statistics across training batches with EMA as in FD-Loss (Yang et al., 2026a), and recover the component covariances. We set $\omega = \pi$.

Paired OT matching. With matched weights, $\Gamma = \text{diag}(\pi)$ is feasible in Eq. (7). Empirically, in a pMF-H training run with this objective, *the optimal component transport matrix was diagonal at every recorded plan refresh* (Appendix H). Motivated by this observation, we fix this correspondence and calculate only the diagonal terms in the Mixed Wasserstein distance, optimizing:

$$\mathcal{L}_{\text{pair-}W_2} = \sum_{k=1}^{K} \pi_k W_2^2(q_k, p_k) \geq M W_2^2(Q, P) \geq W_2^2(Q, P). \quad (11)$$

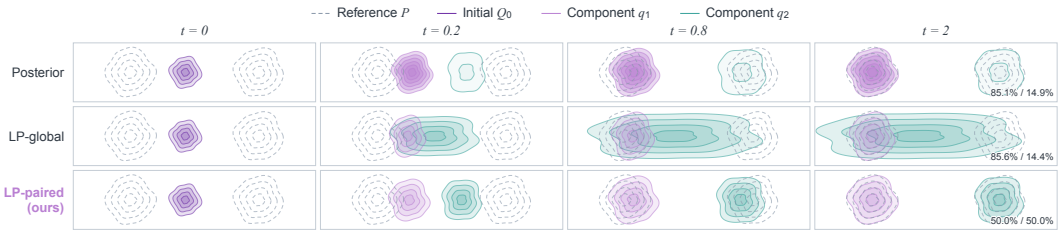


Figure 3: **Toy experiment with component assignment and transport.** Score matching with $K = 2$ moves 2048 particles. Dashed contours show the reference P. Shading and coutour counts indicate weights. LP fixes component weights, but global velocity collapses modes. Only LP-paired achieves correct transport.

which provide an upper-bound for W_2^2. A sufficient condition for diagonal optimality is given in Appendix B.4. Fixed pairing reduces the number of Gaussian pair costs from K^2 to K. We differentiate this objective through the current batch's contribution to the EMA statistics of the generated components, keeping historical statistics and the LP assignments R^* detached.

Paired score matching. For the KL discrepancy, pairing is more than a matter of efficiency, as constraining the component statistics is not sufficient to constrain the feature updates. The global field in Eq. (8) still weights reference and generated scores by separate mixture responsibilities, rather than the LP assignments. Demonstrated in the second row of Figure 3, the global velocity cannot distinguish the target component for each sample, resulting in mode collapse. To preserve the assignment in the update direction, we instead match corresponding components using

$$\mathcal{F}_{\text{pair}} = \sum_k \pi_k D_{\text{KL}}(q_k \| p_k) \geq D_{\text{KL}}(Q \| P) \tag{12}$$

which also upper-bounds the marginal KL. Appendix B.4 gives the proof and further analysis. Calculating closed-form KL between Gaussians require heavy computation, so instead we use the WGF-induced velocity. Reusing R^* for these fields gives the regularized update

$$v_{\text{pair},n} = \sum_{k=1}^{K} R^*_{nk}[s^{\lambda}_{p,k}(z_n) - s^{\lambda}_{q,k}(z_n)], \quad s^{\lambda}_{r,k}(z) = (\Sigma_{r,k} + \lambda I)^{-1}(\mu_{r,k} - z). \tag{13}$$

Here $\lambda > 0$ stabilizes covariance inversion. The same assignment now weights both scores, so the update retains the correspondence used to estimate q_k. We evaluate the field using detached, pre-update statistics and train the generator by detached-target regression:

$$\mathcal{L}_{\text{pair-KL}} = \frac{1}{2B} \sum_{n=1}^{B} \|z_n - \text{sg}[z_n + \eta v_{\text{pair},n}]\|_2^2, \qquad \eta > 0. \tag{14}$$

The entire target is detached, and the EMA state is committed after the generator step. The third row in Figure 3 exhibits that this formulation correctly transport samples from initial to the reference distribution. An ablation study in Table 2 proves that LP-based component allocation uniformly achieves better results than posterior assignments. Paired transport is crucial especially when training multi-step JiT models with score based discrepancies. This is possibly due to the large gap between initial one-step generated distribution and the target distribution induce abnormal component allocation. Detailed training and evaluation settings are given in Appendix H.

3.3 Training MGFlow.

As discussed above, we train two variants of MGFlow: *MGFlow-W_2* with the paired OT-based objective in Equation 11, and *MGFlow-KL* with paired score-based updates in Equation 14. For multiple encoders, we use fixed, discrepancy-specific normalizers for the encoder losses:

$$\mathcal{L}_{\text{MGFlow-}W_2} = \sum_e \frac{\mathcal{L}^e_{\text{pair-}W_2}}{W_2^2(R_e, V_e)}, \quad \mathcal{L}_{\text{MGFlow-KL}} = \sum_e \frac{\mathcal{L}^e_{\text{pair-KL}}}{D_{\text{KL}}(R_e \| V_e)}. \tag{15}$$

Here $\mathcal{L}^e_{\text{pair-}W_2}$ and $\mathcal{L}^e_{\text{pair-KL}}$ sum the respective branch losses for encoder e, while R_e and V_e are single-Gaussian fits to its real training and validation features. For KL calibration, the same score

ridge is added to both covariances. Unlike FD-Loss's real–generated (R, G) normalization, these fixed (R, V) scales avoid downweighting harder-to-match feature spaces merely because their current discrepancies are large (Appendix C.2). For text-to-image generation, we concatenate image features with frozen SigLIP2 text features and match their joint distributions (Feng et al., 2026) (Appendices B.6 and C.3.2).

4 Experiments

4.1 Increasing Gaussian Components

We first examine how distributional granularity affects ImageNet post-training. We compare individual Gaussian mixture models with $K \in \{1, 4, 16\}$ components and their combinations. Beyond selecting a single GM, we consider *hierarchical resolution*: for a set of component counts $\mathcal{K}$, we sum the corresponding training losses with equal weights,

$$\mathcal{L}_{\mathcal{K}} = \sum_{K \in \mathcal{K}} \mathcal{L}_K, \tag{16}$$

where $\mathcal{L}_K$ denotes the MGFlow loss using K-component distribution models. Thus, $K = 1 + 4 + 16$ combines global moment matching with progressively finer componentwise matching, forming a coarse-to-fine hierarchy of objectives.

Table 3: **Ablation on component counts.** Using pMF-B. All models train in Inception feature space for 10 epochs. FDr^5 evaluates five held-out encoders.

Components	$\text{FDr}^6\downarrow$	$\text{FDr}^5\downarrow$	FID↓
$K = 1$	12.73	15.04	1.95
$K = 4$	12.28	14.54	1.65
$K = 16$	11.32	13.41	**1.46**
$K = 1 + 4$	11.51	13.59	1.89
$K = 1 + 4 + 16$	**11.05**	**13.08**	1.51

We train pMF-B in Inception feature space for 10 epochs with different component counts. Among the same 256×256 resolution configurations, increasing K from 1 to 16 reduces FDr^6 from 12.73 to 11.32 and FID from 1.95 to 1.46, demonstrating that distributional models with finer granularity enhance generation. Hierarchical resolution further improves generalization, lowering FDr^6. These results suggest that coarse and fine distributional constraints provide complementary supervision, motivating multi-resolution matching in the subsequent experiments. By default we use $K = 1 + 4 + 16$ for MGFlow-KL and $K = 1 + 4$ for MGFlow-W_2. Larger mixtures are limited by the cost of Gaussian Wasserstein matching for MGFlow-W_2 and the samples available to accurately estimate each component's mean and covariance for MGFlow-KL (see Appendix I.2).

4.2 Class-Conditioned ImageNet Generation

We post-train the B, L, and H variants of JiT (Li & He, 2026) and pMF (Lu et al., 2026) from their official pretrained weights on ImageNet (Russakovsky et al., 2015) 256×256. All use frozen SigLIP (Tschannen et al., 2025), Inception (Szegedy et al., 2016), and MAE (He et al., 2022) encoders (SIM), a global batch of 1,024, and are trained for 100 epochs. We evaluate post-trained models with 50,000 one-step generated samples. FD-Loss (Yang et al., 2026a) motivates FDr^6 as a more robust metric for ImageNet generation than FID alone, reducing the blind spots of a single representation. We report FDr^6 across six representation spaces and FDr^3 across the three encoders not used for training, together with FID (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016). Optimization, sampling, and reference-fitting details are given in Appendices C and F.

Table 4 compares MGFlow with FD-Loss (Yang et al., 2026a), AdvFD (Gao et al., 2026), and AMFD (Liu et al., 2026). Both MGFlow-W_2 and MGFlow-KL surpasses the FD-Loss baseline on every model size. MGFlow-KL achieves *state-of-the-art* $\mathbf{1.45}$ FDr^6 on pMF-H and $\mathbf{1.64}$ on JiT-H, improving over the FD-Loss baseline with **23%** and **38%** margins, respectively. In particular, its gains on the three held-out encoders indicate better generalization beyond the representations used for training, demonstrating MGFlow can truly match distributions in feature spaces, not just overfitting on the first and second moments. Additionally, MGFlow-KL uses KL-based matching without any FD loss, yet outperforms all the FD-based baselines on both FDr^6 and held-out FDr^3, showing that its gains do not rely on directly optimizing these evaluation metrics. See Appendix J for per-encoder results. Qualitative results are demonstrated in Appendix K

Table 4: **Post-training JiT and pMF on ImageNet 256 × 256.** Backbones: Li & He (2026); Lu et al. (2026). All models train in the SIM encoder spaces for 100 epochs. 'Model' denotes feature-distribution modeling. FDr^6 evaluate the average Fréchet distance of 6 encoders while FDr^3 evaluates the three held-out encoders. Baseline results are from FD-Loss (Yang et al., 2026a), AdvFD (Gao et al., 2026), and AMFD (Liu et al., 2026). † denotes the full-CFG NFE upper bound for interval CFG; dashes denote unavailable results. Best and second-best results per backbone are bold and underlined, respectively. More baselines are in Table 10.

Method	Model	Discrepancy	NFE	#Params	$FDr^6\downarrow$	$FDr^3\downarrow$	FID↓	IS↑
Pixel-space models, multi-step backbones								
JiT-B	–	–	50×2×2†	131M	15.65	15.06	3.71	269.0
+ FD-Loss	Gaussian	W_2	1	131M	5.53	8.45	1.00	**344.6**
+ AdvFD	Gaussian	W_2	1	131M	3.92	<u>6.03</u>	**0.79**	–
+ AMFD-C	Gaussian	Amortized W_2	1	131M	4.75	6.71	<u>0.95</u>	319.4
+ AMFD-U	Gaussian	Amortized W_2	1	131M	<u>3.91</u>	6.13	<u>0.95</u>	<u>325.2</u>
+ MGFlow-W_2	GM	W_2	1	131M	4.43	7.36	1.45	313.9
+ MGFlow-KL	GM	KL	1	131M	**3.70**	**5.70**	1.28	308.5
JiT-L	–	–	50×2×2†	459M	10.73	10.27	2.59	288.5
+ FD-Loss	Gaussian	W_2	1	459M	3.24	5.46	<u>0.77</u>	317.3
+ AdvFD	Gaussian	W_2	1	459M	<u>2.01</u>	3.20	**0.73**	–
+ AMFD-C	Gaussian	Amortized W_2	1	459M	2.67	3.86	0.87	**325.1**
+ AMFD-U	Gaussian	Amortized W_2	1	459M	2.02	<u>3.12</u>	0.85	<u>319.8</u>
+ MGFlow-W_2	GM	W_2	1	459M	2.68	4.54	1.07	314.9
+ MGFlow-KL	GM	KL	1	459M	**1.92**	**2.96**	1.00	304.7
JiT-H	–	–	50×2×2†	953M	7.66	9.07	1.97	296.0
+ FD-Loss	Gaussian	W_2	1	953M	2.65	4.44	<u>0.75</u>	<u>313.0</u>
+ AdvFD	Gaussian	W_2	1	953M	1.80	2.93	**0.72**	–
+ AMFD-C	Gaussian	Amortized W_2	1	953M	2.15	3.10	0.85	**328.1**
+ AMFD-U	Gaussian	Amortized W_2	1	953M	<u>1.79</u>	<u>2.78</u>	0.83	312.2
+ MGFlow-W_2	GM	W_2	1	953M	2.55	4.42	0.99	307.8
+ MGFlow-KL	GM	KL	1	953M	**1.64**	**2.57**	0.94	301.9
Pixel-space models, one-step								
pMF-B	–	–	1	118M	13.70	11.82	3.31	254.6
+ FD-Loss	Gaussian	W_2	1	118M	3.50	4.49	<u>0.85</u>	<u>331.4</u>
+ AdvFD	Gaussian	W_2	1	118M	3.32	4.22	**0.81**	–
+ AMFD-C	Gaussian	Amortized W_2	1	118M	3.94	4.66	0.95	315.9
+ AMFD-U	Gaussian	Amortized W_2	1	118M	3.43	4.20	0.92	310.1
+ MGFlow-W_2	GM	W_2	1	118M	**3.03**	<u>4.02</u>	1.65	**332.1**
+ MGFlow-KL	GM	KL	1	118M	<u>3.09</u>	**3.76**	1.56	323.0
pMF-L	–	–	1	410M	9.09	7.62	2.72	261.7
+ FD-Loss	Gaussian	W_2	1	410M	2.09	2.72	<u>0.78</u>	309.2
+ AdvFD	Gaussian	W_2	1	410M	1.89	2.57	**0.77**	–
+ AMFD-C	Gaussian	Amortized W_2	1	410M	2.25	2.75	0.88	<u>321.8</u>
+ AMFD-U	Gaussian	Amortized W_2	1	410M	2.01	2.57	0.86	306.7
+ MGFlow-W_2	GM	W_2	1	410M	<u>1.80</u>	<u>2.47</u>	1.20	**322.4**
+ MGFlow-KL	GM	KL	1	410M	**1.74**	**2.24**	1.16	314.8
pMF-H	–	–	1	935M	6.87	6.09	2.29	267.2
+ FD-Loss	Gaussian	W_2	1	935M	1.89	2.69	<u>0.77</u>	310.1
+ AdvFD	Gaussian	W_2	1	935M	1.74	2.50	**0.74**	–
+ AMFD-C	Gaussian	Amortized W_2	1	935M	1.93	2.43	0.86	**323.0**
+ AMFD-U	Gaussian	Amortized W_2	1	935M	1.75	2.30	0.85	307.3
+ MGFlow-W_2	GM	W_2	1	935M	<u>1.50</u>	<u>2.05</u>	1.09	<u>316.1</u>
+ MGFlow-KL	GM	KL	1	935M	**1.45**	**1.88**	1.07	311.0

4.3 Text-to-Image Generation

We initialize from the distilled FLUX.2 [klein] 4B model (Black Forest Labs, 2026) and train MGFlow-KL with $K = 1+4$ for 1,000 steps with a batch size of 1,024. Following the exact protocol adopted in prior works (Liu et al., 2026; Feng et al., 2026), we use reconstructed COCO (Lin et al., 2014) reference images augmented with GenEval images. We use two versions of MGFlow. For

Table 5: **Text-to-image generation with FLUX.2 [klein] 4B** (Black Forest Labs, 2026) at $\mathbf{512 \times 512}$. We report GenEval (Ghosh et al., 2023) and PickScore (Kirstain et al., 2023) on 499 Pick-a-Pic prompts; higher is better. Bold and underlining mark the best and second-best scores, respectively. MGFlow-KL uses $K = 1 + 4$ and 1,000 training steps. Both use reconstructed COCO references supplemented with images generated from GenEval prompts. Joint matching concatenates image and SigLIP2 text features. Baselines are from AMFD (Liu et al., 2026); dashes denote unavailable results. Reference construction is in Appendix C.3.2.

Method	NFE	Single	Two	Count	Colors	Position	Color attr.	Overall	PickScore
FLUX.2 [klein] 4B	4	0.994	0.904	0.791	0.880	0.575	0.623	0.794	21.85
DMD2 (Yin et al., 2024a)	1	<u>0.997</u>	0.894	0.806	0.864	0.603	0.660	0.804	–
FD-SIM (Yang et al., 2026a)	1	**1.000**	0.944	0.716	0.878	0.618	0.658	0.802	21.62
iRDM (Feng et al., 2026)	1	0.994	0.924	0.756	<u>0.923</u>	0.650	0.708	0.826	21.82
AMFD-U-SIM (Liu et al., 2026)	1	0.994	0.929	0.741	0.902	0.638	0.678	0.813	21.77
AMFD-C-SIM (Liu et al., 2026)	1	<u>0.997</u>	0.955	0.791	<u>0.923</u>	0.670	0.740	0.846	21.85
AMFD-C-10 enc. (Liu et al., 2026)	1	**1.000**	0.957	0.778	0.920	<u>0.680</u>	0.733	0.845	21.82
MGFlow (image-only)	1	<u>0.997</u>	<u>0.967</u>	<u>0.844</u>	**0.926**	0.633	<u>0.770</u>	<u>0.856</u>	<u>21.86</u>
MGFlow (joint)	1	**1.000**	**0.982**	**0.891**	**0.926**	**0.783**	**0.818**	**0.900**	**21.98**

the joint version, we concatenate image features with frozen SigLIP2 (Tschannen et al., 2025) text features (Feng et al., 2026), matching joint distribution $p(x, c)$ (and thus, from the Bayes' theorem, the conditional distribution) instead of marginal distribution $p(x)$ for better text-image alignment. By contrast, the image-only variant uses image features alone, matching the marginal distribution. We use the MGFlow-KL variant with $K = 1 + 4$ hierarchical resolution. The model is trained to generate 512×512 images in one step. We report GenEval (Ghosh et al., 2023) and PickScore on Pick-a-Pic (Kirstain et al., 2023). Reference construction and training details are in Appendix C.3.2.

With the COCO reference, joint matching reaches a GenEval score of $\mathbf{0.900}$ in Table 5, compared with 0.794 for the four-step backbone, 0.826 for iRDM, and 0.846 for AMFD-C-SIM. It also achieves the highest PickScore of $\mathbf{21.98}$ in the table. iRDM (Feng et al., 2026) uses a batch size of 10,240 for 180 steps, whereas AMFD (Liu et al., 2026) uses 1,024 for 1,500 steps. MGFlow achieves higher GenEval and PickScore scores with **44%** and **33%** fewer generated training samples, respectively. Qualitative text-to-image examples are provided in Appendix L.

5 Related Work

One-step generation can be learned through trajectory consistency, average velocities, or diffusion distillation (Song et al., 2023; Geng et al., 2025a; Yin et al., 2024a). Recent methods instead supervises generated populations through feature space global discrepancies (Yang et al., 2026a; Feng et al., 2026) or construct updates from attraction–repulsion or transport between batches (Deng et al., 2026; Han et al., 2026). We propose the unified theoretical framework of distributional training that recovers all these works. MGFlow uses a novel Gaussian Mixture Model for distribution approximation that differs from all above. For detailed Related Works see Appendix A.

6 Conclusion

We presented a unified view of distributional training that separates sampling, encoding, distribution modeling, and matching discrepancy. Wasserstein gradient flow connects global objectives to feature updates, recovering FD-Loss and Gaussian-kernel Drifting as specific choices. MGFlow combines Gaussian mixtures with mass-constrained allocation and paired OT or KL updates. Experiments on ImageNet and text-to-image generation show improvements across training and held-out representations, with KL-based matching improving Fréchet metrics without directly optimizing them. Both modeling granularity and component correspondence matter for one-step generation. More broadly, our findings point toward richer distribution modeling and principled distribution matching as promising directions for advancing one-step generative models.

AI USE STATEMENT

We used AI tools for language polishing

ETHICS STATEMENT

Our models could be misused to generate deceptive images, including deepfakes. Such misuse may facilitate misinformation or violate individuals' privacy.

REPRODUCIBILITY STATEMENT

We provide implementation details, experimental settings, and evaluation protocols in the appendix. We will publicly release our full code and trained model weights to support reproducibility.

REFERENCES

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence, 2026.

Jiarui Cao, Zixuan Wei, and Yuxin Liu. Gradient flow drifting: Generative modeling via wasserstein gradient flows of KDE-approximated divergences. *arXiv preprint arXiv:2603.10592*, 2026.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://papers.nips.cc/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

Julie Delon and Agnès Desolneux. A Wasserstein-type distance in the space of Gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020. doi: 10.1137/19M1301047. URL https://arxiv.org/abs/1907.05254.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x.

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

D. C. Dowson and B. V. Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. doi: 10.1016/0047-259X(82)90077-X.

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. URL https://arxiv.org/abs/2607.02375.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2681–2690, 2019. URL https://proceedings.mlr.press/v89/feydy19a.html.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OlzB6LnXcS.

Leonard T. Franz, Sebastian Hoffmann, Tim Weiland, Bernhard Schölkopf, and Georg Martius. Drifting fields are not conservative. *arXiv preprint arXiv:2604.06333*, 2026.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pp. 316–321. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.repl4nlp-1.31/.

Mingju Gao, Jingkai Zhou, Kun Gai, Changqian Yu, and Hao Tang. AdvFD: Boosting visual generation via adversarial fréchet distance loss. *arXiv preprint arXiv:2608.11205*, 2026. URL https://arxiv.org/abs/2608.11205.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 75460–75482, 2025a. doi: 10.52202/085713-2534. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6d13e085b79d454da5910e4ca82a3d9d-Paper-Conference.pdf.

Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and J. Zico Kolter. Consistency models made easy. In *International Conference on Learning Representations*, 2025b. URL https://arxiv.org/abs/2406.14548.

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*, 2026. URL https://arxiv.org/abs/2512.02012.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.11513.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL https://www.jmlr.org/papers/v13/gretton12a.html.

Jiaqi Han, Puheng Li, Qiushan Guo, Renyuan Xu, Stefano Ermon, and Emmanuel J. Candès. One-step generative modeling via Wasserstein gradient flows. *arXiv preprint arXiv:2605.11755*, 2026. URL https://arxiv.org/abs/2605.11755.

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. URL https://arxiv.org/abs/2111.06377.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998. doi: 10.1137/S0036141096303359.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.02279.

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. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.01569.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. doi: 10.1214/aoms/1177729694.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 122458–122483, 2024. doi: 10.52202/079017-3892. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/dd540e1c8d26687d56d296e64d35949f-Paper-Conference.pdf.

Chieh-Hsin Lai, Bac Nguyen, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, Stefano Ermon, and Molei Tao. A unified view of score-based and drifting models. *arXiv preprint arXiv:2603.07514*, 2026.

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, October 2025. URL https://arxiv.org/abs/2504.10483.

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*, 2026. URL https://arxiv.org/abs/2511.13720.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.11838.

Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1718–1727, 2015. URL https://proceedings.mlr.press/v37/li15.html.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Wenze Liu, Xintao Wang, Pengfei Wan, and Xiangyu Yue. Amortized moment matching for visual generation. *arXiv preprint arXiv:2607.26860*, 2026. URL https://arxiv.org/abs/2607.26860.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

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

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. URL https://arxiv.org/abs/2601.22158.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. URL https://arxiv.org/abs/2310.04378.

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*, 2024. URL https://arxiv.org/abs/2401.08740.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. URL https://arxiv.org/abs/2304.07193.

Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. doi: 10.1214/aoms/1177704472.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. FlowAR: Scale-wise autoregressive image generation meets flow matching. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2412.15205.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://arxiv.org/abs/1409.0575.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.00512.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://arxiv.org/abs/1606.03498.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024. URL https://arxiv.org/abs/2403.12015.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/41bd71e7bf7f9fe68f1c936940fd06bd-Abstract-Conference.html.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

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. URL https://arxiv.org/abs/1512.00567.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.02905.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Erkan Turan, Nicolas Dufour, and Maks Ovsjanikov. Generative drifting is secretly score matching: A spectral and variational perspective. *arXiv preprint arXiv:2603.09936*, 2026.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion. In *International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2507.23268.

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*, 2026b. URL https://arxiv.org/abs/2504.05741.

Li K. Wenliang and Heishiro Kanagawa. Blindness of score-based methods to isolated components and mixing proportions. *arXiv preprint arXiv:2008.10087*, 2020. URL https://arxiv.org/abs/2008.10087.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2301.00808.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2507.01467.

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

Jiawei Yang, Tianhong Li, Lijie Fan, Yonglong Tian, and Yue Wang. Latent denoising makes good tokenizers. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2507.15856.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2501.01423.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a. URL https://arxiv.org/abs/2405.14867.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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*, 2024b. URL https://arxiv.org/abs/2311.18828.

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. URL https://arxiv.org/abs/2602.09024.

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. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.06940.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.11690.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62307–62331, 2024. URL https://proceedings.mlr.press/v235/zhou24x.html.

APPENDIX CONTENTS

Page

A Related Work . . . 16
A.1 One-step generation and distribution matching . . . 16
A.2 Drifting and its gradient-flow interpretations . . . 16
A.3 Representation-based distributional post-training . . . 17
A.4 Mixture transport and componentwise scores . . . 17

B Derivations and Proofs . . . 17
B.1 Derivation of the KDE–KL interpretation of Drifting . . . 17
B.2 Derivation of the Gaussian OT interpretation of FD-Loss . . . 18
B.3 Global KL Velocity for Gaussian Mixtures . . . 20
B.4 Component Matching and Paired Updates . . . 20
B.5 Energy descent and detached-target regression . . . 21
B.6 Conditional image scores in a joint Gaussian . . . 22

C Implementation . . . 23
C.1 Algorithms . . . 23
C.2 Weighting multiple representation losses . . . 26
C.3 Training configurations . . . 27
C.4 Statistics EMA schedule . . . 29

D Additional ImageNet Results . . . 29

E Evaluation and Comparison Protocols . . . 30
E.1 The 10-epoch distribution-model comparison . . . 30
E.2 Evaluation in training and held-out representations . . . 30

F Reference GMs and Component Structure . . . 31

G Score Ridge Settings . . . 33

H Component Matching . . . 34

I Computation and Training Efficiency . . . 34
I.1 Training-time comparison and protocol . . . 34
I.2 Scaling the number of components . . . 35

J Results in Individual Evaluation Representations . . . 38

K Qualitative ImageNet Results . . . 39

L Additional Text-to-Image Examples . . . 41

A Related Work

A.1 One-step generation and distribution matching

One-step generation can be learned by approximating a sampling trajectory or by directly matching the output distribution. Consistency models (Song et al., 2023) map points on the same probability-flow trajectory to a common endpoint and support both distillation and training without a teacher. MeanFlow (Geng et al., 2025a) instead learns an average velocity over a time interval, using its relation to the instantaneous velocity as a training target. Improved Mean Flows (Geng et al., 2026) and pixel Mean Flows (Lu et al., 2026) further develop this approach. These methods determine how a generator produces an image in one step. Distributional post-training optimizes the distribution of those images and can be applied to an already trained one-step model. Our experiments use both pMF (Lu et al., 2026) and a one-step initialization from JiT (Li & He, 2026).

Distribution Matching Distillation (DMD) (Yin et al., 2024b) expresses a reverse-KL gradient through the difference between real and generated scores at noisy image distributions. A pre-trained diffusion model provides the real score, while a separate diffusion model estimates the generated score. Its original formulation also uses a regression loss on teacher-generated pairs. DMD2 (Yin et al., 2024a) removes this regression requirement, uses two-timescale updates to improve the generated-score estimate, and incorporates an adversarial loss on real images. MGFlow also uses a difference of scores for KL matching, but computes these scores from explicit Gaussian mixtures in frozen representation spaces. It does not train diffusion score networks for the matching loss.

Kernel distribution matching predates recent one-step diffusion models. Generative Moment Matching Networks (Li et al., 2015) train a generator with maximum mean discrepancy (MMD), a kernel two-sample criterion (Gretton et al., 2012). A characteristic kernel can distinguish distributions beyond their first two moments, whereas a Gaussian Fréchet objective depends only on means and covariances. Our GM representation retains componentwise moments and assignments, allowing distributional training to distinguish modes that a single Gaussian cannot separate.

A.2 Drifting and its gradient-flow interpretations

Drifting (Deng et al., 2026) constructs a field from attraction to real features and repulsion from generated features. The generator is trained to regress its features toward detached field-shifted targets. The iterative distribution update takes place during training; inference uses a single generator evaluation. This separates training-time movement of a distribution from the denoising trajectory used by a diffusion sampler.

Several works analyze the relation between these fields and scores. Lai et al. (2026) connect Gaussian-kernel mean shifts to scores of smoothed densities and analyze the residual for more general radial kernels. Turan et al. (2026) develop a spectral and variational view, relating Gaussian-kernel Drifting to score differences and studying the effect of bandwidth. Cao et al. (2026) formulate Drifting through Wasserstein gradient flows of KDE-approximated divergences, including extensions beyond KL. The choice of kernel and normalization matters: Franz et al. (2026) show that general normalized Drifting fields need not be conservative, with the Gaussian kernel providing an exception. Our KDE–KL connection uses this Gaussian-kernel setting; it does not identify every Drifting variant with the same KL gradient flow.

Wasserstein gradient flows describe steepest descent of a distributional energy under transport geometry (Jordan et al., 1998; Peyré & Cuturi, 2019). W-Flow (Han et al., 2026) applies this perspective to one-step generation using the Sinkhorn divergence between empirical measures. Its field subtracts the self-transport barycentric projection from the cross-distribution projection. W-Flow and Gaussian-kernel Drifting thus use different discrepancies even though both construct fields from sample interactions. Table 1 groups them as sample-based methods, with empirical measures for W-Flow and KDE for Gaussian-kernel Drifting. MGFlow instead estimates a finite collection of component statistics and evaluates transport or score fields from them.

A.3 Representation-based distributional post-training

FD-Loss (Yang et al., 2026a) directly optimizes the Fréchet distance between real and generated feature statistics. The Gaussian distance has a closed form (Dowson & Landau, 1982), so training requires no learned critic. Real statistics can be precomputed, while queues or exponential moving averages provide generated statistics beyond a single batch. Gradients pass through the current batch's contribution. Using several frozen encoders extends the objective beyond a single representation. Our Gaussian OT case recovers this moment-based objective. Replacing OT by KL gives a different field with the same Gaussian representation; using a GM changes the representation of the distribution itself.

Representation Distribution Matching (RDM) (Feng et al., 2026) organizes visual generation around discrepancies between feature pushforward distributions. Its iRDM implementation uses a Nyström MMD estimator, fixed reference statistics, and joint image–text matching for conditional generation. This work makes clear that the representation, reference, estimator, and discrepancy all affect training. Our study focuses on the distribution model and its update: we compare Gaussian, GM, and sample-based constructions, then address component assignment and correspondence for GM training. For text-to-image generation, we follow iRDM in concatenating image and text features for joint matching.

AdvFD (Gao et al., 2026) augments frozen representations with a learned representation that maximizes the Fréchet discrepancy while the generator minimizes it. Whitening real features constrains the learned representation and prevents a trivial increase through feature scaling. Its adversarial update adapts the representation to the current generator. MGFlow keeps the encoders fixed and instead refines the distribution model within each representation.

Amortized Moment Matching (AMFD) (Liu et al., 2026) represents moment matching through affine denoising operators and amortizes these operators with neural networks. Its formulation supports representation spaces and native generative spaces without explicitly maintaining all covariance matrices in the loss. MGFlow uses explicit full-covariance components and closed-form Gaussian scores or pair costs. The main additional problem is then to assign new generated samples to components while keeping them matched to the reference mixture.

Frozen visual representations are also used inside generative models. REPA (Yu et al., 2025) aligns diffusion-transformer hidden states with pretrained features, while representation autoencoders (Zheng et al., 2026) use pretrained representations as a generative latent space. These uses differ from matching the distribution of generated outputs in several feature spaces. The latter lets us change the training objective without changing the generator's sampling architecture.

A.4 Mixture transport and componentwise scores

The Wasserstein distance between two Gaussians has a closed form, but the same is not true for arbitrary Gaussian mixtures. Delon & Desolneux (2020) define MW_2 by restricting the coupling to Gaussian component pairs. The squared cost upper-bounds W_2^2 between the mixtures. MGFlow builds on this discrepancy, but its LP acts at the sample level: it allocates generated features to components with prescribed reference masses. This differs from transporting mass between two already fitted mixtures.

For KL, mixture scores can be insensitive to mixing proportions when components are well separated (Wenliang & Kanagawa, 2020). MGFlow combines explicit mass constraints with paired component scores, rather than relying on independently weighted marginal scores. Appendix B.4 relates this update to labelled KL.

B Derivations and Proofs

B.1 Derivation of the KDE–KL interpretation of Drifting

We derive the mean-shift–score identity and its KL interpretation in Eq. (4).

From the mean-shift field to the KDE score. For a fixed bandwidth $h > 0$, let $k_h(z, y) = \mathcal{N}(z; y, h^2 I)$ and define

$$r_h(z) = \int k_h(z, y)\, \mathrm{d}r(y), \qquad r \in \{p, q_\theta\}. \tag{17}$$

Differentiating with respect to the query z, while holding r fixed, gives

$$\begin{aligned} \nabla_z k_h(z, y) &= \frac{y - z}{h^2} k_h(z, y), \\ \nabla_z \log r_h(z) &= \frac{\int \nabla_z k_h(z, y)\, \mathrm{d}r(y)}{\int k_h(z, y)\, \mathrm{d}r(y)} \\ &= \frac{1}{h^2} \frac{\mathbb{E}_{y\sim r}[k_h(z, y)(y - z)]}{\mathbb{E}_{y\sim r}[k_h(z, y)]} = \frac{a_r(z)}{h^2}. \end{aligned} \tag{18}$$

The Gaussian kernel and its first derivatives are bounded, so the derivative can pass through the integral. The Gaussian-kernel drifting field $v_{\mathrm{drift}} = a_p - a_{q_\theta}$ therefore satisfies

$$v_{\mathrm{drift}}(z) = h^2 \big[\nabla_z \log p_h(z) - \nabla_z \log q_h(z)\big] \tag{19}$$

(Lai et al., 2026; Turan et al., 2026; Franz et al., 2026).

From KL to the score difference. For smooth positive densities ρ and fixed p_h, write $\mathcal{E}(\rho) = \int \rho \log(\rho/p_h)\, \mathrm{d}z$. A mass-preserving perturbation $\rho + \varepsilon\eta$, with $\int \eta\, \mathrm{d}z = 0$, gives

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \mathcal{E}(\rho + \varepsilon\eta)\Big|_{\varepsilon=0} &= \int \left(\log \frac{\rho(z)}{p_h(z)} + 1\right) \eta(z)\, \mathrm{d}z, \\ \frac{\delta \mathcal{E}}{\delta \rho}(z) &= \log \rho(z) - \log p_h(z) + 1. \end{aligned} \tag{20}$$

Taking the negative spatial gradient gives the Wasserstein velocity

$$v_\rho(z) = -\nabla_z \frac{\delta \mathcal{E}}{\delta \rho}(z) = \nabla_z \log p_h(z) - \nabla_z \log \rho(z) \tag{21}$$

(Jordan et al., 1998). Evaluating at $\rho = q_h$ proves $v_{\mathrm{drift}} = h^2 v_{\mathrm{KL}}$ (Cao et al., 2026).

This is the KL field evaluated at the smoothed densities. For comparison, varying the composite functional $\mathcal{G}(q) = D_{\mathrm{KL}}(k_h * q \| p_h)$ with respect to q gives

$$\frac{\delta \mathcal{G}}{\delta q}(x) = \int k_h(z, x) \left(\log \frac{q_h(z)}{p_h(z)} + 1\right) \mathrm{d}z. \tag{22}$$

Here $\delta q_h(z) = \int k_h(z, x)\, \delta q(x)\, \mathrm{d}x$ introduces an additional smoothing operation; its negative gradient is not generally the field in Eq. (21).

Detached-target regression. Drifting uses the regression loss (Deng et al., 2026)

$$\mathcal{L}_{\mathrm{drift}} = \frac{1}{B} \sum_i \|z_i - \mathrm{sg}[z_i + v_{\mathrm{drift}}(z_i)]\|_2^2 . \tag{23}$$

The general calculation in Appendix B.5 gives $\nabla_{z_i} \mathcal{L}_{\mathrm{drift}} = -2 v_{\mathrm{drift}}(z_i)/B$.

B.2 Derivation of the Gaussian OT interpretation of FD-Loss

We derive the Gaussian transport cost, its feature gradient, and the finite-sample factors underlying the FD field in Eq. (4).

The Gaussian optimal transport cost and map. Let $q_{\mathrm{G}} = \mathcal{N}(\mu_q, \Sigma_q)$ and $p_{\mathrm{G}} = \mathcal{N}(\mu_p, \Sigma_p)$, with positive definite covariances. All matrix square roots below are symmetric positive definite. The optimal Gaussian transport map is

$$T(z) = \mu_p + A(z - \mu_q), \qquad A = \Sigma_q^{-1/2}(\Sigma_q^{1/2}\Sigma_p\Sigma_q^{1/2})^{1/2}\Sigma_q^{-1/2}. \tag{24}$$

Indeed, $A\Sigma_q A = \Sigma_p$, so T sends q_{G} to p_{G}. Since A is positive definite, T is the gradient of a convex quadratic and is optimal for quadratic transport (Peyré & Cuturi, 2019). Writing $z - T(z) = (\mu_q - \mu_p) + (I - A)(z - \mu_q)$, we obtain

$$\begin{aligned} W_2^2(q_{\mathrm{G}}, p_{\mathrm{G}}) &= \mathbb{E}_{q_{\mathrm{G}}}\|z - T(z)\|_2^2 \\ &= \|\mu_q - \mu_p\|_2^2 + \operatorname{tr}\big((I - A)\Sigma_q(I - A)\big) \\ &= \|\mu_q - \mu_p\|_2^2 + \operatorname{tr}(\Sigma_q + \Sigma_p - 2A\Sigma_q) \\ &= \|\mu_q - \mu_p\|_2^2 + \operatorname{tr}\Big(\Sigma_q + \Sigma_p - 2(\Sigma_q^{1/2}\Sigma_p\Sigma_q^{1/2})^{1/2}\Big) = \mathcal{D}_{\mathrm{FD}}. \end{aligned} \tag{25}$$

The third line uses $A\Sigma_q A = \Sigma_p$; the last uses cyclicity of the trace. This is the Gaussian Fréchet formula (Dowson & Landau, 1982).

Differentiating the mean and covariance. Set $F = \frac{1}{2}\mathcal{D}_{\mathrm{FD}}$, keeping the reference moments fixed. The mean term gives $\nabla_{\mu_q} F = \mu_q - \mu_p$. For the covariance term, use the equivalent cross term $\operatorname{tr}(M^{1/2})$, where $M = \Sigma_p^{1/2}\Sigma_q\Sigma_p^{1/2}$. The two cross-term matrices are BB^{T} and $B^{\mathsf{T}}B$ for $B = \Sigma_q^{1/2}\Sigma_p^{1/2}$, so they have the same eigenvalues. To differentiate its trace, write $S = M^{1/2}$ and differentiate $S^2 = M$:

$$S\,\mathrm{d}S + \mathrm{d}S\,S = \mathrm{d}M \quad \Longrightarrow \quad \mathrm{d}\operatorname{tr}(M^{1/2}) = \tfrac{1}{2}\operatorname{tr}(M^{-1/2}\mathrm{d}M). \tag{26}$$

The implication follows by multiplying by S^{-1} and taking the trace; it does not require M and $\mathrm{d}M$ to commute. Therefore

$$\begin{aligned} \mathrm{d}_{\Sigma_q} F &= \tfrac{1}{2}\operatorname{tr}(\mathrm{d}\Sigma_q) - \tfrac{1}{2}\operatorname{tr}(M^{-1/2}\Sigma_p^{1/2}\,\mathrm{d}\Sigma_q\,\Sigma_p^{1/2}) \\ &= \tfrac{1}{2}\operatorname{tr}\big((I - A)\,\mathrm{d}\Sigma_q\big). \end{aligned} \tag{27}$$

Here $\Sigma_p^{1/2}M^{-1/2}\Sigma_p^{1/2} = A$: both are positive definite solutions of $X\Sigma_q X = \Sigma_p$, whose unique solution follows by squaring $\Sigma_q^{1/2}X\Sigma_q^{1/2}$. Thus

$$\nabla_{\mu_q} F = \mu_q - \mu_p, \qquad \nabla_{\Sigma_q} F = \tfrac{1}{2}(I - A). \tag{28}$$

From moment derivatives to feature movement. Let q be any generated distribution with finite second moments and positive definite covariance. Perturb its features by $z_\varepsilon = z + \varepsilon u(z)$, and denote their distribution by q_ε. Differentiating the mean and $\Sigma_q = \mathbb{E}_q[zz^{\mathsf{T}}] - \mu_q\mu_q^{\mathsf{T}}$ at $\varepsilon = 0$ gives

$$\begin{aligned} \dot{\mu}_q &= \mathbb{E}_q[u(z)], \\ \dot{\Sigma}_q &= \mathbb{E}_q[u(z)z^{\mathsf{T}} + zu(z)^{\mathsf{T}}] - \dot{\mu}_q\mu_q^{\mathsf{T}} - \mu_q\dot{\mu}_q^{\mathsf{T}} \\ &= \mathbb{E}_q[u(z)(z - \mu_q)^{\mathsf{T}} + (z - \mu_q)u(z)^{\mathsf{T}}]. \end{aligned} \tag{29}$$

Combining these with Eq. (28) yields

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\varepsilon}F(\mu_{q_\varepsilon}, \Sigma_{q_\varepsilon})\bigg|_0 &= (\mu_q - \mu_p)^{\mathsf{T}}\dot{\mu}_q + \tfrac{1}{2}\operatorname{tr}((I - A)\dot{\Sigma}_q) \\ &= \mathbb{E}_q\Big[\big(\mu_q - \mu_p + (I - A)(z - \mu_q)\big)^{\mathsf{T}}u(z)\Big] \\ &= \mathbb{E}_q[(z - T(z))^{\mathsf{T}}u(z)]. \end{aligned} \tag{30}$$

The two covariance terms combine because $I - A$ is symmetric. Hence the moment objective has descent field $T(z) - z$. When $q = q_{\mathrm{G}}$, this is the Wasserstein velocity of $\frac{1}{2}W_2^2(q, p_{\mathrm{G}})$:

$$v_{\mathrm{FD},t}(z) = T_{q_{\mathrm{G},t} \to p_{\mathrm{G}}}(z) - z, \qquad \partial_t q_{\mathrm{G},t} + \nabla \cdot (q_{\mathrm{G},t}v_{\mathrm{FD},t}) = 0. \tag{31}$$

The affine field preserves Gaussianity. For non-Gaussian q, the same feature gradient differentiates the moment objective, but T need only match the reference mean and covariance, not the full distribution.

Setting $v = v_{\mathrm{FD}}$ and $\eta = 1$ in the regression formula of Appendix B.5 gives the target $\operatorname{sg}[T(z_i)]$ and gradient $(z_i - T(z_i))/B$. This recovers the same field; using $\mathcal{D}_{\mathrm{FD}} = 2F$ instead of F multiplies its gradient by two.

Finite-sample and EMA factors. For a feature z_i contributing weight c_i to the estimated mean and raw second moment, while all other contributions are fixed,

$$\mathrm{d}\mu_q = c_i\,\mathrm{d}z_i, \qquad \mathrm{d}\Sigma_q = c_i\big[\mathrm{d}z_i(z_i-\mu_q)^\mathsf{T} + (z_i-\mu_q)\mathrm{d}z_i^\mathsf{T}\big]. \tag{32}$$

Substitution into Eq. (28) gives

$$\nabla_{z_i} F = c_i\big[\mu_q - \mu_p + (I-A)(z_i-\mu_q)\big] = c_i(z_i - T(z_i)). \tag{33}$$

For moments averaged over N features, $c_i = 1/N$; detached queue entries affect the moments but receive no gradient. For EMA updates of the mean and raw second moment with decay β and current batch size B, $c_i = (1-\beta)/B$, with T evaluated at the updated moments (Yang et al., 2026a).

If the empirical covariance instead uses $\Sigma_q = (N-1)^{-1}\sum_j (z_j-\mu_q)(z_j-\mu_q)^\mathsf{T}$, its derivative has coefficient $1/(N-1)$, while the mean retains $1/N$:

$$\nabla_{z_i} F = \frac{\mu_q - \mu_p}{N} + \frac{(I-A)(z_i-\mu_q)}{N-1}. \tag{34}$$

Here the terms from differentiating the centering cancel because $\sum_j (z_j - \mu_q) = 0$.

B.3 Global KL Velocity for Gaussian Mixtures

Let P and Q have positive mixture weights and positive definite component covariances. Applying Eq. (21) to these densities gives $v_{\mathrm{global}} = \nabla \log P - \nabla \log Q$. To evaluate each mixture score, differentiate $Q(z) = \sum_k \omega_k q_k(z)$:

$$\begin{aligned}\nabla_z \log Q(z) &= \frac{\sum_k \omega_k \nabla_z q_k(z)}{Q(z)} \\ &= \sum_k \frac{\omega_k q_k(z)}{Q(z)} \nabla_z \log q_k(z) = \sum_k \gamma_{Q,k}(z) s_{q,k}(z).\end{aligned} \tag{35}$$

For $q_k = \mathcal{N}(\mu_{q,k}, \Sigma_{q,k})$,

$$s_{q,k}(z) = -\tfrac{1}{2}\nabla_z\big[(z-\mu_{q,k})^\mathsf{T}\Sigma_{q,k}^{-1}(z-\mu_{q,k})\big] = \Sigma_{q,k}^{-1}(\mu_{q,k} - z). \tag{36}$$

Applying the same calculation to P yields Eq. (8). The two scores use their respective posterior responsibilities $\gamma_{Q,k} = \omega_k q_k / Q$ and $\gamma_{P,k} = \pi_k p_k / P$.

B.4 Component Matching and Paired Updates

Mixture transport and its paired upper bound. For each component pair, let γ_{ij} be an optimal coupling of q_i and p_j, with cost $C_{ij} = W_2^2(q_i, p_j)$. For any $\Gamma \in U(\omega, \pi)$, $\gamma = \sum_{i,j} \Gamma_{ij}\gamma_{ij}$ has marginals $\sum_i \omega_i q_i = Q$ and $\sum_j \pi_j p_j = P$. Its transport cost is $\sum_{i,j}\Gamma_{ij}C_{ij}$, so minimizing over Γ gives $W_2^2(Q,P) \le MW_2^2(Q,P)$ (Delon & Desolneux, 2020). When $\omega = \pi$, the feasible coupling $\mathrm{diag}(\pi)$ gives

$$W_2^2(Q,P) \le MW_2^2(Q,P) \le \sum_k \pi_k C_{kk}, \tag{37}$$

which proves the two bounds in Eq. (11).

When is diagonal component transport optimal? Suppose $C_{ii} \le C_{ij}$ for every i, j and both mixture weights are π. For any feasible Γ, its row sums give

$$\sum_{i,j}\Gamma_{ij}C_{ij} - \sum_i \pi_i C_{ii} = \sum_{i,j}\Gamma_{ij}(C_{ij} - C_{ii}) \ge 0. \tag{38}$$

The diagonal coupling attains equality and is therefore optimal. It is the unique optimum if $C_{ii} < C_{ij}$ for all $j \ne i$. Matched weights alone do not imply this condition: swapping the labels of two equally weighted, distinct Gaussians makes off-diagonal pairing optimal. The training observation is reported in Appendix H.

Paired KL and the marginal KL bound. For fixed positive weights π_k summing to one, introduce the labelled distributions $\widetilde{Q}(k,z) = \pi_k q_k(z)$ and $\widetilde{P}(k,z) = \pi_k p_k(z)$. Since the weights cancel inside the ratio,

$$D_{\mathrm{KL}}(\widetilde{Q}\|\widetilde{P}) = \sum_k \int \pi_k q_k(z) \log \frac{\pi_k q_k(z)}{\pi_k p_k(z)} \,\mathrm{d}z = \sum_k \pi_k D_{\mathrm{KL}}(q_k\|p_k) = \mathcal{F}_{\mathrm{pair}}. \tag{39}$$

To relate this to marginal KL, factor $\widetilde{Q}(k,z) = Q(z)\gamma_{Q,k}(z)$ and $\widetilde{P}(k,z) = P(z)\gamma_{P,k}(z)$. Then

$$\begin{aligned}
\mathcal{F}_{\mathrm{pair}} &= \sum_k \int Q(z)\gamma_{Q,k}(z) \left[\log \frac{Q(z)}{P(z)} + \log \frac{\gamma_{Q,k}(z)}{\gamma_{P,k}(z)}\right] \mathrm{d}z \\
&= D_{\mathrm{KL}}(Q\|P) + \mathbb{E}_{z\sim Q}\left[\sum_k \gamma_{Q,k}(z) \log \frac{\gamma_{Q,k}(z)}{\gamma_{P,k}(z)}\right] \\
&= D_{\mathrm{KL}}(Q\|P) + \mathbb{E}_{z\sim Q}[D_{\mathrm{KL}}(\gamma_Q(\cdot \mid z)\|\gamma_P(\cdot \mid z))] \geq D_{\mathrm{KL}}(Q\|P).
\end{aligned} \tag{40}$$

The second line uses $\sum_k \gamma_{Q,k} = 1$. Equality holds exactly when the posterior label distributions agree Q-almost everywhere.

From the paired energy to component fields. Let u_k move component q_k, and set $g_k = \nabla_z \log(q_k/p_k)$. Using the KL variation in Eq. (20) and $\partial_t q_k = -\nabla \cdot (q_k u_k)$ gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}_{\mathrm{pair}} = \sum_k \pi_k \mathbb{E}_{q_k}[g_k^{\mathsf{T}} u_k], \tag{41}$$

with vanishing boundary terms. The squared product Wasserstein distance $\sum_k \pi_k W_2^2(q_k, q_k')$ corresponds to squared speed $\sum_k \pi_k \mathbb{E}_{q_k}\|u_k\|_2^2$. Its steepest-descent velocity minimizes

$$\sum_k \pi_k \mathbb{E}_{q_k}\left[g_k^{\mathsf{T}} u_k + \tfrac{1}{2}\|u_k\|_2^2\right] = \frac{1}{2}\sum_k \pi_k \mathbb{E}_{q_k}\left[\|u_k + g_k\|_2^2 - \|g_k\|_2^2\right]. \tag{42}$$

The minimum is attained at

$$v_k(z) = -g_k(z) = \nabla_z \log p_k(z) - \nabla_z \log q_k(z). \tag{43}$$

Thus the common weights in the energy and metric do not introduce an extra factor π_k into the component velocity (Jordan et al., 1998; Peyré & Cuturi, 2019).

The regularized scores in Eq. (13) are those of $r_k^\lambda = \mathcal{N}(\mu_{r,k}, \Sigma_{r,k} + \lambda I)$. Training combines their differences with the detached LP assignments R_{nk}^* and applies the regression gradient derived in Appendix B.5. Unlike the marginal KL velocity in Eq. (8), this update uses R_{nk}^* for both scores rather than separate mixture posteriors.

B.5 Energy descent and detached-target regression

Continuous distributional descent. Let $\phi_t = \delta\mathcal{F}/\delta Q|_{Q=Q_t}$, where t denotes continuous flow time. For a smooth density satisfying $\partial_t Q_t + \nabla \cdot (Q_t v_t) = 0$, assume sufficient decay or no-flux boundary conditions so that the boundary term below vanishes. The chain rule and integration by parts give (Jordan et al., 1998; Peyré & Cuturi, 2019)

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}(Q_t) &= \int \phi_t(z)\, \partial_t Q_t(z) \,\mathrm{d}z \\
&= -\int \phi_t(z) \nabla \cdot (Q_t(z) v_t(z)) \,\mathrm{d}z \\
&= \int Q_t(z) \nabla \phi_t(z)^{\mathsf{T}} v_t(z) \,\mathrm{d}z.
\end{aligned} \tag{44}$$

Substituting $v_t = -\nabla\phi_t$ proves Eq. (3):

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}(Q_t) = -\mathbb{E}_{z\sim Q_t}\|v_t(z)\|_2^2 \leq 0. \tag{45}$$

Applying a field to generator training. For $z_n = E(G_\theta(\xi_n, c_n), c_n)$, define $\bar{z}_n = \mathrm{sg}[z_n + \eta v_n]$ and $\mathcal{L}_v = (2B)^{-1} \sum_n \|z_n - \bar{z}_n\|_2^2$. The entire target is constant during differentiation, so

$$\nabla_{z_n} \mathcal{L}_v = \frac{z_n - \bar{z}_n}{B} = -\frac{\eta}{B} v_n, \qquad \nabla_\theta \mathcal{L}_v = -\frac{\eta}{B} \sum_n J_n^\mathsf{T} v_n, \quad J_n = \frac{\partial z_n}{\partial \theta}. \tag{46}$$

No derivative of the estimated field enters this gradient. The regression transfers the field through the encoder and generator Jacobians; the continuous-flow energy identity is not a finite-step guarantee for the parameter optimizer.

B.6 Conditional image scores in a joint Gaussian

Let $p(z, u)$ be a joint Gaussian over image features z and text features u, with

$$\mu = \begin{bmatrix} \mu_z \\ \mu_u \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} \Sigma_{zz} & \Sigma_{zu} \\ \Sigma_{uz} & \Sigma_{uu} \end{bmatrix} \succ 0. \tag{47}$$

Write $a = z - \mu_z$, $b = u - \mu_u$, and $[\xi_z; \xi_u] = \Sigma^{-1}[a; b]$. The Gaussian image score is $-\xi_z$. Instead of forming the full inverse, solve the block equations:

$$\begin{aligned} \Sigma_{zz}\xi_z + \Sigma_{zu}\xi_u &= a, \\ \Sigma_{uz}\xi_z + \Sigma_{uu}\xi_u &= b. \end{aligned} \tag{48}$$

The second equation gives $\xi_u = \Sigma_{uu}^{-1}(b - \Sigma_{uz}\xi_z)$. Substituting into the first yields

$$\underbrace{(\Sigma_{zz} - \Sigma_{zu}\Sigma_{uu}^{-1}\Sigma_{uz})}_{\Sigma_{z|u}} \xi_z = a - \Sigma_{zu}\Sigma_{uu}^{-1} b. \tag{49}$$

Define $\mu_{z|u} = \mu_z + \Sigma_{zu}\Sigma_{uu}^{-1}(u - \mu_u)$. We obtain

$$\nabla_z \log p(z, u) = -\xi_z = \Sigma_{z|u}^{-1}(\mu_{z|u} - z) = \nabla_z \log p(z \mid u). \tag{50}$$

The last equality also follows from $\log p(z, u) = \log p(z \mid u) + \log p(u)$, since u is fixed. For paired GM matching, we apply this calculation to each joint Gaussian component and combine the image-score differences with R^*_{nk}. Image–text cross-covariance therefore affects the image update even though text features receive no gradient.

C IMPLEMENTATION

C.1 ALGORITHMS

We fit the real-data distributions once, initialize generated statistics from the pretrained generator, and then update the generator and its statistics jointly. Each encoder and component count has a separate reference and moment state. A $K = 1 + 4$ configuration therefore contains one Gaussian branch and one four-component branch. Algorithms 1 and 2 are run once for each encoder–branch pair; Algorithm 3 is repeated at each training step. We write $E(x, c)$ for the features being matched: image features for image-only training, or concatenated image and text features for joint matching.

Reference fitting. Algorithm 1 gives the full-data EM update (Dempster et al., 1977). Features are cached and processed in blocks, but parameters are updated only after a complete pass. The saved covariance is the raw centered second moment.

Algorithm 1 Full-covariance reference GM fitting

Require: Cached real features $Y = \{y_n\}_{n=1}^{N}$, component count K, initial parameters (π, μ, Σ), maximum EM iterations $T_{\max}$
Ensure: Frozen reference $P = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$
if $K = 1$ **then**
  $\mu_1 \leftarrow N^{-1} \sum_n y_n$; $\Sigma_1 \leftarrow N^{-1} \sum_n y_n y_n^\top - \mu_1 \mu_1^\top$
  return $P = \mathcal{N}(\mu_1, \Sigma_1)$, with $\pi_1 = 1$
end if
for $t = 1, \ldots, T_{\max}$ **do**
  $(\pi^{\text{old}}, \mu^{\text{old}}, \Sigma^{\text{old}}) \leftarrow (\pi, \mu, \Sigma)$
  Set $N_k = 0$, $b_k = 0$, $S_k = 0$ for all k ▷ Reset full-data accumulators
  for each feature block with indices $\mathcal{I}$ **do**
    For all $n \in \mathcal{I}$ and k, compute the E-step responsibilities:

$$r_{nk} \leftarrow \frac{\pi_k^{\text{old}} \mathcal{N}(y_n; \mu_k^{\text{old}}, \Sigma_k^{\text{old}})}{\sum_j \pi_j^{\text{old}} \mathcal{N}(y_n; \mu_j^{\text{old}}, \Sigma_j^{\text{old}})}$$

    For all k, accumulate $N_k \leftarrow N_k + \sum_{n \in \mathcal{I}} r_{nk}$
    $b_k \leftarrow b_k + \sum_{n \in \mathcal{I}} r_{nk} y_n$; $S_k \leftarrow S_k + \sum_{n \in \mathcal{I}} r_{nk} y_n y_n^\top$ for all k
  end for
  For all k, perform the M-step after processing all N features:

$$\pi_k \leftarrow N_k / N;\ \mu_k \leftarrow b_k / N_k;\ \Sigma_k \leftarrow S_k / N_k - \mu_k \mu_k^\top$$

  Compute parameter changes relative to $(\pi^{\text{old}}, \mu^{\text{old}}, \Sigma^{\text{old}})$
  if the convergence test below passes **then**
    break
  end if
end for
return frozen reference $P = \sum_k \pi_k \mathcal{N}(\mu_k, \Sigma_k)$

The final full-data fits start from a hard-assignment initialization. We check maximum absolute weight change and relative RMS changes in means and covariances, using thresholds 0.002, 0.006, and 0.020, respectively. The test must pass on two consecutive updates without an increase in these changes. These checks concern the fitted reference; the training-time LP does not refit or alter P.

Generated statistics. For assignments R_{nk}, define the batch statistics $\widehat{a}_k = B^{-1}\sum_n R_{nk}$, $\widehat{b}_k = B^{-1}\sum_n R_{nk} z_n$, and $\widehat{S}_k = B^{-1}\sum_n R_{nk} z_n z_n^\mathsf{T}$. The stored state is $H = \{a_k, b_k, S_k\}_k$, from which $\mu_{q,k} = b_k/a_k$ and $\Sigma_{q,k} = S_k/a_k - \mu_{q,k}\mu_{q,k}^\mathsf{T}$. Algorithm 2 initializes this state without a generator update. Subsequent updates use an EMA, following FD-Loss (Yang et al., 2026a).

For the LP assignments R^* in Eq. (10), the capacity constraint fixes $a_k = \widehat{a}_k = \pi_k$. The conditional batch mean and raw second moment are therefore

$$\widehat{\mu}_k = \frac{1}{B\pi_k}\sum_n R^*_{nk} z_n, \qquad \widehat{M}_k = \frac{1}{B\pi_k}\sum_n R^*_{nk} z_n z_n^\mathsf{T}. \tag{51}$$

Writing $M_{q,k} = S_k/a_k$, the EMA with decay β gives

$$\mu^+_{q,k} = \beta\mu_{q,k} + (1-\beta)\widehat{\mu}_k, \qquad M^+_{q,k} = \beta M_{q,k} + (1-\beta)\widehat{M}_k, \qquad \Sigma^+_{q,k} = M^+_{q,k} - \mu^+_{q,k}(\mu^+_{q,k})^\mathsf{T}. \tag{52}$$

The reference remains the EM-fitted P: applying the LP to real features need not recover the same moments, since LP assignments differ from EM responsibilities.

Algorithm 2 Warm-starting generated component statistics

Require: Pretrained generator G_{θ_0}, frozen encoder E, reference $P = \sum_k \pi_k p_k$, sample budget N_0, assignment batch size B_0
Ensure: Initial moment state $H = \{a_k, b_k, S_k\}_{k=1}^K$; no generator update

Disable gradient recording; set processed count $m = 0$
Set totals $A_k = 0$, $U_k = 0$, $V_k = 0$ for all k
while $m < N_0$ **do**
  $B \leftarrow \min(B_0, N_0 - m)$ ▷ Include the final partial batch
  Sample $\{(\xi_n, c_n)\}_{n=1}^B$ using the training sampling scheme
  Compute $z_n = E(G_{\theta_0}(\xi_n, c_n), c_n)$ for all n
  if $K = 1$ **then**
    Set $R_{n1} = 1$ for every n
  else
    $C_{nk} \leftarrow -\log(\pi_k p_k(z_n))$ for all n, k
    Solve $R \in \arg\min_{R\geq 0} \sum_{n,k} R_{nk} C_{nk}$
      subject to $R\mathbf{1}_K = \mathbf{1}_B$ and $R^\mathsf{T}\mathbf{1}_B = B\pi$
  end if
  For all k, accumulate $A_k \leftarrow A_k + \sum_n R_{nk}$; $U_k \leftarrow U_k + \sum_n R_{nk} z_n$
  $V_k \leftarrow V_k + \sum_n R_{nk} z_n z_n^\mathsf{T}$ for all k; $m \leftarrow m + B$
end while
For all k, set $(a_k, b_k, S_k) \leftarrow (A_k, U_k, V_k)/N_0$
return $H = \{a_k, b_k, S_k\}_k$ ▷ $a_k = \pi_k$ by the LP constraints

We use $B_0 = 1{,}024$ and $N_0 = 50{,}000$. Encoder microbatches are collected into these assignment batches before solving the LP. Initialization averages the accumulated moments; it does not apply EMA.

LP-paired training. Algorithm 3 distinguishes the two gradient paths. OT differentiates through the current batch in the candidate EMA state. KL evaluates the field using the stored state before its update and detaches the regression target. In both cases the assignment is fixed during back-propagation, and the EMA is committed once per global batch. The Gaussian OT cost uses raw covariances, clipping numerical negative eigenvalues to zero when evaluating matrix square roots. Here sg stops gradients without changing values. Encoder parameters remain fixed, but gradients pass through their image inputs to the generator. Within each encoder–branch iteration below, we omit the indices (e, K) on component parameters and moment entries.

Algorithm 3 One generator update with LP-paired OT or KL

Require: Generator G_θ and its optimizer; frozen encoders E_e and references $P_{e,K}$
Require: Stored states $H_{e,K}$, branch counts $K \in \mathcal{K}$, weights $w_{e,K}$, global batch size B, EMA decay β_t
Require: Matching objective (OT or KL); for KL, score ridges $\lambda_{e,K}$ and field scale η
Ensure: Updated generator parameters and one EMA update per encoder–branch pair

Clear generator gradients; set $\mathcal{L} \leftarrow 0$
Sample $\{(\xi_n, c_n)\}_{n=1}^{B}$; generate $x_n = G_\theta(\xi_n, c_n)$ once
For each encoder e, compute $z_{e,n} = E_e(x_n, c_n)$ for all n
for each encoder e and branch K **do**
  Set $z_n = z_{e,n}$ and read the fixed reference $P_{e,K} = \sum_k \pi_k p_k$
  if $K = 1$ **then**
    Set $R_{n1} = 1$ for every n
  else
    Compute $C_{nk} = -\log(\pi_k p_k(\mathrm{sg}[z_n]))$ and solve Eq. (10)
  end if
  $R \leftarrow \mathrm{sg}[R]$ ▷ Reuse this assignment for moments and paired fields
  $\widehat{a}_k \leftarrow B^{-1} \sum_n R_{nk}$; $\widehat{b}_k \leftarrow B^{-1} \sum_n R_{nk} z_n$ for all k
  $\widehat{S}_k \leftarrow B^{-1} \sum_n R_{nk} z_n z_n^\mathsf{T}$ for all k
  $H^+_{e,K} \leftarrow \beta_t \,\mathrm{sg}[H_{e,K}] + (1 - \beta_t)(\widehat{a}, \widehat{b}, \widehat{S})$
    Keep $H_{e,K}$ unchanged until after the optimizer step.
  if OT matching **then**
    $\mu^+_{q,k} \leftarrow b^+_k / a^+_k$; $\Sigma^+_{q,k} \leftarrow S^+_k / a^+_k - \mu^+_{q,k} (\mu^+_{q,k})^\mathsf{T}$ for all k
    $\ell_{e,K} \leftarrow \sum_k \pi_k W_2^2(\mathcal{N}(\mu^+_{q,k}, \Sigma^+_{q,k}), p_k)$
  else ▷ KL matching
    From stored $H_{e,K}$, set $\mu_{q,k} = b_k / a_k$ and $\Sigma_{q,k} = S_k / a_k - \mu_{q,k} \mu_{q,k}^\mathsf{T}$
    Without gradients, solve $(\Sigma_{r,k} + \lambda_{e,K} I) s_{r,k,n} = \mu_{r,k} - z_n$
      for $r \in \{p, q\}$, all k, n, using Cholesky factors shared across samples
    $v_n \leftarrow \sum_k R_{nk}(s_{p,k,n} - s_{q,k,n})$; $\bar{z}_n \leftarrow \mathrm{sg}[z_n + \eta v_n]$
    $\ell_{e,K} \leftarrow (2B)^{-1} \sum_n \|z_n - \bar{z}_n\|_2^2$
  end if
  $\mathcal{L} \leftarrow \mathcal{L} + w_{e,K} \ell_{e,K}$
end for
Backpropagate $\mathcal{L}$ to θ and take one optimizer step
Commit $H_{e,K} \leftarrow \mathrm{sg}[H^+_{e,K}]$ for every encoder and branch

Global-batch gradients with limited memory. When a global batch spans several devices or accumulation passes, we first collect detached features and solve the LP on the complete batch. We compute the loss gradient with respect to these features, then replay each generator microbatch with the same noise, conditions, and random state. Sequential encoder vector–Jacobian products propagate its slice of the feature gradient to the generator (Gao et al., 2021). The feature loss is already normalized by the global batch size; parameter gradients are summed without a second batch-size normalization. The assignment and candidate EMA state remain fixed across replay passes.

C.2 WEIGHTING MULTIPLE REPRESENTATION LOSSES

The discrepancies have different scales across encoders. FD-Loss (Yang et al., 2026a) divides each encoder's FD by its detached current value plus a constant, giving a weight $[\mathrm{sg}\{\mathrm{FD}(R_e, G_e)\}+c]^{-1}$. Here R_e and G_e denote real and generated feature statistics. This weight changes with the generator. We instead use the discrepancy between real training features R_e and real validation features V_e to set a fixed scale for each encoder.

For OT, we use $w_e = 1/\mathrm{FD}(R_e, V_e)$, with the same encoder weight for every component-count branch. In SigLIP/Inception/MAE order, the denominators are $(0.62469, 1.67956, 0.04212)$. Table 6 compares the published FD-Loss results with our aligned single-Gaussian FD runs. The aligned recipe uses this fixed normalization and the statistics schedule in Appendix C.4, without a GM branch. It reduces FDr^6 on JiT-B.

Table 6: **Single-Gaussian FD with fixed reference normalization.** $\mathrm{FDr}^6 \downarrow$ after 100 epochs with SIM encoders. FD-Loss results are from Yang et al. (2026a); the aligned runs use our statistics-update recipe.

Training recipe	Encoder normalization	pMF-B
FD-Loss	$\mathrm{sg}\{\mathrm{FD}(R, G)\} + c$	3.50
Aligned Gaussian FD	$\mathrm{FD}(R, V)$	**3.17**

For KL, we similarly use $w_e = 1/D_{\mathrm{KL}}(R_e \| V_e)$, where R_e and V_e are single-Gaussian fits. These calibration distributions are distinct from the training objective $D_{\mathrm{KL}}(Q\|P)$. The resulting SigLIP/Inception/MAE weights are $(0.327, 0.110, 0.276)$ for the ImageNet reference. Each weight is reused across the encoder's Gaussian and mixture branches; we do not estimate a separate marginal GM KL to normalize each branch.

For multiple branches, the final loss weight is $w_{e,K} = w_e \alpha_K$. All multi-branch configurations use $\alpha_K = 1$ for every included branch. Text-to-image weights are recalculated for the corresponding image-only or joint reference.

C.3 TRAINING CONFIGURATIONS

C.3.1 IMAGENET

Table 7 summarizes the common ImageNet settings for the six JiT/pMF backbones. We use frozen SigLIP2 (Tschannen et al., 2025), Inception-v3 (Szegedy et al., 2016), and MAE (He et al., 2022) encoders, abbreviated as SIM. All six models use AdamW (Loshchilov & Hutter, 2019), global batch size 1,024, and 100 epochs. One epoch corresponds to 1,250 optimizer updates. Learning rates are 10^{-5} for JiT and 10^{-6} for pMF, with five warmup epochs followed by cosine decay. We use no gradient clipping or dropout.

Table 7: **ImageNet post-training configurations.** Slash-separated pMF sampling values follow B/L/H order.

Configuration	JiT (Li & He, 2026)	pMF (Lu et al., 2026)
Model and representations		
Model sizes	B, L, H	B, L, H
Image resolution	256×256	
Initialization	Official pretrained weights	
Frozen encoders	SigLIP2, Inception-v3, MAE	
Training		
Training epochs	100	
Global batch size	1,024	
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.95)$	
Learning rate	10^{-5}	10^{-6}
Learning-rate schedule	Cosine decay	
Warmup epochs	5	
Weight decay	0	
Precision	BF16 mixed precision	
Statistics estimation		
Estimator	EMA	
Warm-start samples	50,000 from the pretrained generator	
EMA decay	$0.995 \rightarrow 0.999$, linear over epochs 8–32	
Sampling and evaluation		
NFE	1	
CFG	1.0	8.5 / 7.0 / 7.0
Noise scale	1.0	1.0 / 1.0 / 2.0
Evaluation samples	50,000	

The guidance intervals (Ho & Salimans, 2022; Kynkäänniemi et al., 2024) for pMF-B/L/H are $[0.1, 0.7]$, $[0.2, 0.7]$, and $[0.2, 0.6]$; JiT uses $[0.1, 1.0]$. Evaluation uses 50,000 one-step samples. Each component count in a configuration has its own fitted reference and generated statistics. Encoder and branch weights are specified in Appendix C.2.

C.3.2 TEXT-TO-IMAGE GENERATION

Table 5 uses two checkpoints initialized from the official distilled FLUX.2 [klein] 4B model (Black Forest Labs, 2026). All use MGFlow-KL with equally weighted $K = 1$ and $K = 4$ branches and frozen SigLIP, MAE, and Inception image encoders. The component-count choice is discussed in Appendix I.2. Following iRDM (Feng et al., 2026), the joint variant concatenates image features with scaled, normalized SigLIP2 (Tschannen et al., 2025) text features and retains the full image–text cross-covariance. For each caption c, we normalize its text embedding as $\widehat{t}(c) = t(c)/\|t(c)\|_2$ and form the joint feature $[f_e(x); \beta_e \widehat{t}(c)]$, without additional normalization of the image feature $f_e(x)$. We set $\beta_e = 0.25\, m_e/m_t$, where m_e and m_t are the square roots of the median pairwise squared distances of image and normalized text features on the same 2,000 reference samples (seed 3407), excluding self-pairs. In SigLIP/Inception/MAE order, $\beta_e \approx (3.76114, 4.11487, 0.51504)$. These scales remain fixed during reference fitting and training and are shared by the $K = 1$ and $K = 4$ branches. The image-only variant omits text from the matching objective, not from the generator conditioning.

Reference data. We follow the reference-data construction protocol of iRDM (Feng et al., 2026) and AMFD (Liu et al., 2026). For each of 82,783 COCO (Lin et al., 2014) train2014 captions, we retain the three highest-PickScore images from 24 teacher-generated candidates, yielding 248,349 reconstructed images. We add 53,357 accepted teacher images from 553 GenEval prompts, with at most 100 accepted images per prompt. Both variants use this shared reference set of 301,706 images, sampled uniformly by image.

Training and evaluation. Table 8 lists the optimizer, statistics, and sampling settings. Inverse-KL encoder weights are computed for each variant. Evaluation uses one Euler step. GenEval (Ghosh et al., 2023) averages its six category scores equally, using the same prompt set as the GenEval reference subset. PickScore (Kirstain et al., 2023) is evaluated on 499 Pick-a-Pic prompts. All evaluation protocols are aligned with AMFD (Liu et al., 2026).

Table 8: **Text-to-image post-training configurations.** Both variants use reconstructed COCO and GenEval reference images.

Configuration	Joint	Image-only
Model and reference		
Initialization	Distilled FLUX.2 [klein] 4B	
Image resolution	512×512	
Image encoders	SigLIP2, MAE, Inception-v3	
Text features	Frozen SigLIP2	–
Reference images	301,706, including 53,357 GenEval images	
Distribution branches	$K = 1 + 4$	
Branch weights	$1 : 1$	
Optimization and statistics		
Training steps	1,000	
Global batch size	1,024	
Optimizer	AdamW (Loshchilov & Hutter, 2019)	
Learning rate	5×10^{-6}	
Learning-rate schedule	150-step warmup, then constant	
Statistics EMA decay	0.99	
Sampling and evaluation		
NFE / CFG	1 / 1	
Text context length	512 tokens	
GenEval samples	2,212 (four per prompt)	
PickScore prompts	499 Pick-a-Pic prompts	

C.4 STATISTICS EMA SCHEDULE

An EMA averages out batch noise but also delays the response to a changing generator. We use decay 0.995 for the first eight ImageNet epochs, increase it linearly to 0.999 over epochs 8–32, and keep it fixed thereafter. The smaller initial decay lets the component statistics follow early distribution changes; the larger final decay smooths the estimates later in training. This EMA updates feature statistics, not generator parameters.

Table 9 compares the schedule with a constant 0.999 decay for JiT-B in the aligned single-Gaussian FD implementation. After 50 epochs, the schedule reduces FDr^6 from 4.99 to 4.87.

Table 9: **Statistics EMA decay.** JiT-B trained for 50 epochs with aligned Gaussian-FD, fixed $\mathrm{FD}(R, V)$ normalization, and SIM encoders. We report FDr^6 ↓.

Statistics decay	FDr^6 ↓
Constant 0.999	4.99
$0.995 \rightarrow 0.999$	**4.87**

D ADDITIONAL IMAGENET RESULTS

Table 10 supplements the JiT/pMF comparison in Table 4 with other discrete-, latent-, and pixel-space generators.

Table 10: **Additional ImageNet** 256×256 **baselines.** Published results from FD-Loss (Yang et al., 2026a), AdvFD (Gao et al., 2026), and AMFD (Liu et al., 2026). FDr^6 averages normalized Fréchet distances over six encoders; FDr^3 excludes SigLIP, Inception, and MAE. † denotes the full-CFG NFE upper bound for interval CFG; dashes denote unavailable results. JiT/pMF results are in Table 4.

Method (K)	NFE	Space	#Params	FDr^6 ↓	FDr^3 ↓	FID ↓	IS ↑
Reference (real images)							
50k validation images	N/A	N/A	N/A	1.00	1.00	1.68	232.2
Discrete-space models							
VAR-d30 (Tian et al., 2024)	10×2	discrete	2B	6.70	6.77	1.97	304.6
BAR-L (Yu et al., 2026)	$256\times2\times4$	discrete	1.1B	3.57	3.35	1.01	281.9
Latent-space models, multi-step							
without semantic distillation							
SiT-XL/2 (Ma et al., 2024)	250×2	latent	675M	8.44	9.20	2.12	256.7
MAR-L (Li et al., 2024)	$256\times2\times100$	latent	478M	6.68	7.32	1.80	293.4
FlowAR-H (Ren et al., 2025)	$50\times2^\dagger$	latent	1.9B	6.13	6.16	1.68	274.1
MAR-H (Li et al., 2024)	$256\times2\times100$	latent	942M	5.61	6.31	1.56	299.5
MAR-L, DeTok (Yang et al., 2026b)	$256\times2\times100$	latent	478M	5.49	6.05	1.39	306.2
with semantic distillation							
REG (Wu et al., 2025)	$250\times2^\dagger$	latent	685M	4.64	5.15	1.54	302.9
SiT-XL/2-REPA (Yu et al., 2025)	$250\times2^\dagger$	latent	675M	5.45	6.05	1.42	306.1
LightningDiT (Yao et al., 2025)	250×2	latent	675M	4.57	5.02	1.42	294.3
DDT-XL (Wang et al., 2026b)	250×2	latent	675M	5.70	6.38	1.26	309.3
REPA-E (Leng et al., 2025)	$250\times2^\dagger$	latent	676M	3.04	3.33	1.17	298.3
RAE-XL (Zheng et al., 2026)	$50\times2^\dagger$	latent	839M	3.26	3.92	1.16	261.0
Latent-space models, one-step							
Drift-L (latent) (Deng et al., 2026)	1	latent	463M	10.92	11.32	1.53	257.2
iMF-XL (Geng et al., 2026)	1	latent	610M	8.39	8.72	1.82	278.9
iMF-XL (Geng et al., 2026)	2	latent	610M	7.48	7.79	1.61	289.1
Pixel-space models							
PixNerd-XL (Wang et al., 2026a)	100×2	pixel	1.0B	5.01	–	2.10	318.8
Drift-L (pixel) (Deng et al., 2026)	1	pixel	465M	10.51	11.18	1.43	305.8

E Evaluation and Comparison Protocols

E.1 The 10-epoch distribution-model comparison

Shared setup. All six methods in Table 1 start from the same pretrained pMF-B and use frozen Inception features on ImageNet 256×256. We train for 10 epochs with global batch size 1,024 and learning rate 10^{-6}. Sampling uses one step, CFG 8.5, guidance interval $[0.1, 0.7]$, and noise scale 1. Class labels are drawn uniformly, and each feature objective matches the global class-marginal distribution. We evaluate 50,000 generated images using FDr^6 under the protocol in Appendix E.2.

The Gaussian and GM methods fit full-covariance reference distributions to real features and keep them fixed during training. Generated-distribution statistics are initialized from 50,000 samples of the pretrained generator and updated by EMA of first and second moments. GM methods additionally use the capacity-constrained LP in Eq. (10) to assign each batch to components with the reference weights.

FD-Loss. The aligned FD-Loss (Yang et al., 2026a) baseline uses a single Gaussian and minimizes its squared Wasserstein distance to the reference. It retains the Gaussian reference, statistics estimator, and fixed real-training/validation normalization of our OT implementation, with no mixture term. Gradients pass through the current batch's contribution to the EMA statistics, while historical statistics are detached.

Gaussian KL. This baseline uses $K = 1$ and the Gaussian score difference $v_{\mathrm{KL}}(z) = s_p(z) - s_q(z)$. We train with detached-target regression in Eq. (14), using $\eta = 1$, and update the generated statistics after the generator step.

MGFlow-W_2. We use a single $K = 4$ branch with LP assignments and minimize the weighted sum of paired Gaussian costs in Eq. (11). The loss uses the same fixed normalization as the aligned FD-Loss baseline. We differentiate through the current batch statistics, keeping the LP assignments and historical statistics detached.

MGFlow-KL. We use a single $K = 16$ branch. The LP assignments weight the paired component score differences in Eq. (13). Training uses the same detached-target regression and statistics-update order as Gaussian KL. Neither GM method includes an additional Gaussian branch or another component count.

W-Flow. W-Flow (Han et al., 2026) uses the released quadratic-cost Sinkhorn objective with $\epsilon = 0.05$, ten iterations, and its feature and force normalization. Each update uses 1,024 gradient-carrying queries, an independent batch of 1,024 generated support samples, and 1,024 real feature-bank samples.

Gaussian-kernel Drifting. Gaussian-kernel Drifting (Deng et al., 2026; Turan et al., 2026) uses the current generated batch as detached negative support and 1,024 real feature-bank samples. Its fixed bandwidths satisfy $2\sigma^2 \in \{0.02, 0.05, 0.2\}$ in normalized feature coordinates, with the self-interaction diagonal masked and per-scale force normalization. Both sample-based methods gather the complete global batch before computing their fields. Their sample interactions replace the moment estimator, so they require no generated statistics initialization or EMA.

E.2 Evaluation in training and held-out representations

Representation encoders. Table 11 lists the six frozen encoders used for ImageNet evaluation. SIM uses SigLIP2, Inception-v3, and MAE for training; ConvNeXt-v2, DINOv2, and CLIP are held out. We use the pooled or CLS features without the classification or contrastive projection head. The five `timm` encoders use bicubic resizing and their pretrained input normalization; Inception-v3 uses TensorFlow-compatible FID preprocessing.

Metric aggregation. We evaluate 50,000 generated images using FDr^6 (Yang et al., 2026a). For an evaluation encoder e, let F_e be the Fréchet distance between generated and real features, and let

Table 11: **ImageNet representation encoders.** Input denotes the resized image side length before patch padding. The first three encoders form SIM; the last three define FDr^3.

Encoder	Variant	Input	Dimension	Feature
SigLIP2 (Tschannen et al., 2025)	ViT-SO400M/16	224	1,152	Attention pool
Inception-v3 (Szegedy et al., 2016)	FID weights	299	2,048	Global average pool
MAE (He et al., 2022)	ViT-L/16	224	1,024	CLS token
ConvNeXt-v2 (Woo et al., 2023)	Base, IN-22K → IN-1K	224	1,024	Global average pool
DINOv2 (Oquab et al., 2023)	ViT-L/14	256	1,024	CLS token
CLIP (Radford et al., 2021)	ViT-L/14, OpenAI	256	1,024	CLS token

b_e be the corresponding real-validation reference value. For Inception features, the unnormalized distance F_e is FID (FD-Inception) (Heusel et al., 2017). The normalized score is $\mathrm{FDr}_e = F_e/b_e$. We use the arithmetic mean across the six evaluation spaces,

$$\mathrm{FDr}^6 = \frac{1}{6}\sum_{e\in\mathcal{H}_6}\mathrm{FDr}_e, \qquad \mathcal{H}_6 = \{\mathrm{Inception}, \mathrm{ConvNeXt}, \mathrm{DINOv2}, \mathrm{MAE}, \mathrm{SigLIP}, \mathrm{CLIP}\}. \tag{53}$$

For a single training encoder e_{train}, we additionally report

$$\mathrm{FDr}^5 = \frac{1}{5}\sum_{e\in\mathcal{H}_6\setminus\{e_{\mathrm{train}}\}}\mathrm{FDr}_e. \tag{54}$$

For SIM training, the corresponding held-out score is

$$\mathrm{FDr}^3 = \frac{1}{3}\left(\mathrm{FDr}_{\mathrm{ConvNeXt}} + \mathrm{FDr}_{\mathrm{DINOv2}} + \mathrm{FDr}_{\mathrm{CLIP}}\right). \tag{55}$$

F Reference GMs and Component Structure

Reference fitting is described in Algorithm 1. This section examines the structure of the fitted components.

We measure class–component association on 200 randomly sampled training images per ImageNet class, for 200,000 images in total. For class c, let

$$r_{ck} = \frac{1}{N_c}\sum_{n:y_n\in c}\gamma_k(y_n), \qquad q_c = \max_k r_{ck}. \tag{56}$$

Here q_c is the average mass assigned to the dominant component of class c.

Table 12: **Class concentration versus image-level assignment.** We use 200 images per ImageNet class. Class concentration q_c is the mean responsibility of class c for its dominant component; "Image peak" averages the maximum responsibility of each image.

Encoder	K	Mean q_c	Classes with $q_c \geq 0.9$	Image peak
MAE	4	81.24%	398	99.88%
SigLIP	4	91.12%	744	99.96%
Inception	4	82.23%	461	99.97%
MAE	16	64.79%	113	99.79%
SigLIP	16	71.88%	229	99.94%
Inception	16	77.21%	305	99.96%

In Table 12, the mean image-level peak remains above 99%, while class concentration decreases at $K = 16$. Individual images have sharp assignments, but images within the same class are distributed across several components.

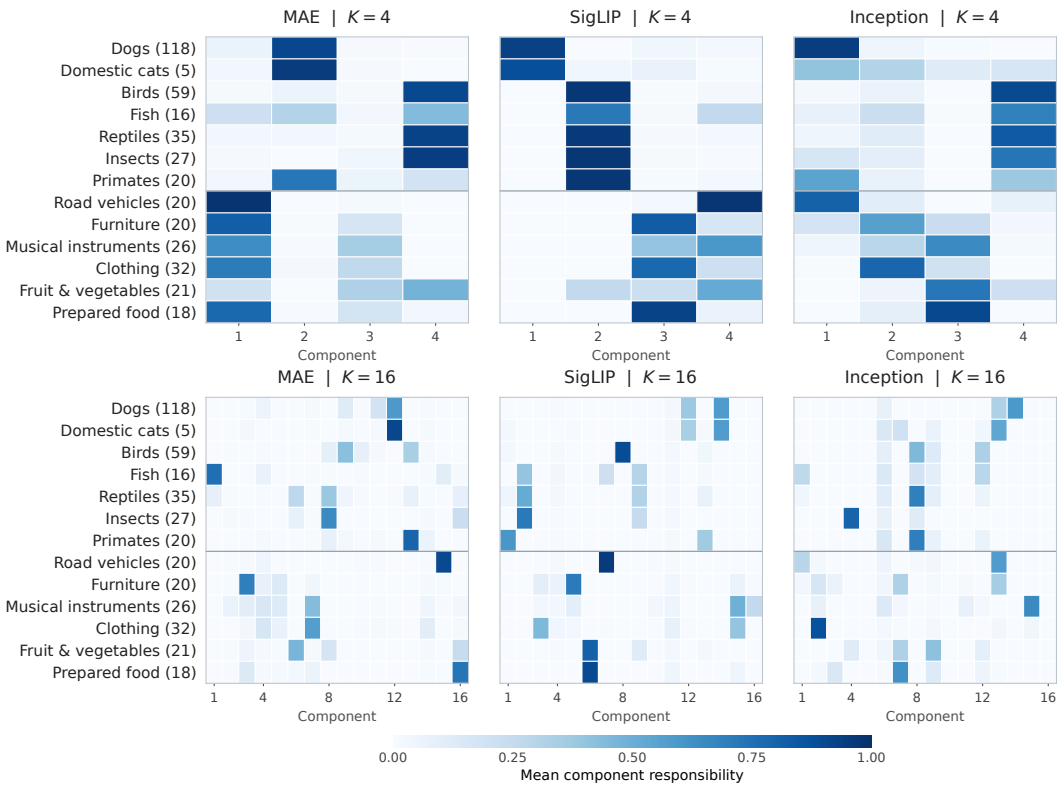


Figure 4: **Semantic groups across independently fitted mixtures.** Rows show mean posterior responsibilities for 13 label-defined groups covering 417 ImageNet classes, using 200 images per class. We average r_{ck} equally over the classes in each group; parentheses give the number of classes. Component indices are local to each GM, not aligned across panels.

Figure 4 shows associations beyond animals and vehicles. For SigLIP, furniture and prepared food share a dominant component at $K = 4$ but peak in distinct components at $K = 16$. The grouping also depends on the encoder: at $K = 4$, Inception places dogs and road vehicles in the same dominant component, whereas MAE and SigLIP separate them.

G SCORE RIDGE SETTINGS

We vary the score ridge λ with $K = 1$, training pMF-B for 10 epochs with MAE, SigLIP, or Inception alone. This preliminary sweep initializes the generated statistics with 5,000 samples, retains a prior of the same size, and uses initial gradient-norm calibration. The aligned comparison in Table 1 instead uses 50,000 initialization samples, no prior, and fixed loss scaling. Table 13 reports the sweep results; their absolute scores are not directly comparable to the aligned runs.

The lowest FDr^6 is obtained at $\lambda = 0.001$ for MAE and $\lambda = 0.03$ for both SigLIP and Inception, reaching 5.769, 5.742, and 10.720, respectively. Larger ridge does not consistently improve the score. The setting with the lowest discrepancy in the training representation can also differ from the one with the lowest FDr^6: for Inception, FID is lowest at $\lambda = 0.01$, whereas FDr^6 is lowest at 0.03.

Table 13: **Single-encoder ridge selection.** pMF-B trained with MGFlow-KL, $K = 1$, for 10 epochs. All metrics are lower-is-better. Bold marks the lowest FDr^6 for each training encoder.

Encoder	λ	FDr^6	FID	$\mathrm{FD}_{\mathrm{MAE}}$	$\mathrm{FD}_{\mathrm{SigLIP}}$
MAE	10^{-5}	5.867	2.559	0.091	8.470
	10^{-4}	5.821	2.749	0.089	8.326
	3×10^{-4}	5.800	2.867	0.086	8.260
	5×10^{-4}	5.837	2.966	0.077	8.353
	10^{-3}	**5.769**	3.207	0.076	8.107
	2×10^{-3}	5.881	3.504	0.072	8.274
	3×10^{-3}	5.902	3.876	0.067	8.380
	10^{-2}	6.312	5.288	0.064	8.780
SigLIP	10^{-5}	6.497	6.116	0.525	3.918
	10^{-4}	6.110	6.535	0.414	3.834
	10^{-3}	6.010	6.487	0.410	3.730
	3×10^{-3}	6.103	7.311	0.416	3.707
	5×10^{-3}	5.975	7.215	0.400	3.588
	10^{-2}	5.970	7.000	0.408	3.576
	3×10^{-2}	**5.742**	7.338	0.366	3.380
	5×10^{-2}	5.831	7.605	0.368	3.406
Inception	10^{-5}	11.109	1.153	0.353	16.464
	10^{-4}	10.966	1.072	0.366	16.070
	3×10^{-4}	10.957	1.087	0.355	16.118
	10^{-3}	11.064	1.139	0.351	16.496
	2×10^{-3}	10.874	1.087	0.351	16.047
	3×10^{-3}	10.828	1.096	0.352	15.793
	5×10^{-3}	10.995	1.105	0.360	16.211
	10^{-2}	10.898	1.057	0.359	15.727
	3×10^{-2}	**10.720**	1.104	0.367	15.528
	5×10^{-2}	10.836	1.105	0.354	15.731

Ridge configuration. For ImageNet, the $K = 1$ branches use $\lambda_1 = (0.03, 0.03, 0.001)$ in SigLIP/Inception/MAE order. These values lie at the $(56.34\%, 62.26\%, 73.24\%)$ percentiles of the corresponding ImageNet $K = 1$ reference covariance spectra. For text-to-image generation, we select the eigenvalues at these same percentiles in the T2I $K = 1$ reference spectra and use nearby rounded values for λ_1 (Table 14). Joint matching uses the full image–text covariance; image-only matching uses the image covariance. We use $\lambda_4 = 3\lambda_1$ in both tasks and $\lambda_{16} = 9\lambda_1$ for ImageNet, with the same ridge added to the reference and generated covariances.

Table 14: **T2I covariance eigenvalue quantiles and score ridges.** The $K = 1$ references use 301,706 COCO and GenEval images. Matched denotes the eigenvalue at the encoder-specific ImageNet percentile; λ_1 is the value used in training.

Encoder	25th	50th	75th	Matched	λ_1
Joint image–text					
SigLIP	1.10×10^{-4}	1.90×10^{-3}	1.71×10^{-2}	3.22×10^{-3}	0.003
Inception	2.16×10^{-3}	5.60×10^{-3}	2.10×10^{-2}	9.44×10^{-3}	0.01
MAE	1.06×10^{-5}	5.36×10^{-5}	2.14×10^{-4}	1.91×10^{-4}	0.0002
Image-only					
SigLIP	6.90×10^{-4}	1.33×10^{-2}	6.88×10^{-2}	1.97×10^{-2}	0.02
Inception	4.44×10^{-3}	1.00×10^{-2}	4.30×10^{-2}	1.87×10^{-2}	0.02
MAE	5.35×10^{-5}	1.51×10^{-4}	5.42×10^{-4}	4.83×10^{-4}	0.0005

H Component Matching

Assignment-ablation settings. All configurations in Table 2 use $K = 4$, without a $K = 1$ term. We train for 10 epochs and evaluate with 50k images. Training time is measured on eight H200 GPUs using the protocol in Appendix I.1.

We inspected a pMF-H run using SIM encoders, $K = 1 + 4$, and a global batch of 1,024. For each encoder, the training code forms all 16 Gaussian costs and solves the component transport LP every 100 updates. The logs cover 34.8 epochs and contain 435 distinct plan refreshes per encoder. Every recorded plan has diagonal mass 1 and exactly four nonzero entries, giving $\Gamma = \mathrm{diag}(\pi)$ in all three feature spaces. This is an observed property of this run, not a consequence of matched weights alone.

I Computation and Training Efficiency

I.1 Training-time comparison and protocol

Table 15 compares training times for the six JiT/pMF backbones. MGFlow-KL runs faster per training step than the official FD-Loss implementation on all L/H backbones, but slightly slower on the two B backbones under the benchmark settings below.

Table 15: **Training time per step.** Seconds per optimizer update on eight H200 GPUs with a global batch of 1,024. MGFlow-W_2 uses $K = 1 + 4$; MGFlow-KL uses $K = 1 + 4 + 16$. Training states and precision settings are specified below. Lower is better.

Backbone	FD-Loss (Yang et al., 2026a)	MGFlow-W_2	MGFlow-KL
JiT-B	**0.799**	0.969	0.846
JiT-L	1.131	1.137	**1.009**
JiT-H	2.192	1.368	**1.247**
pMF-B	**0.829**	0.874	0.882
pMF-L	1.660	**1.103**	1.112
pMF-H	2.572	**1.703**	1.882

The training times in Tables 2 and 15 are measured on the same node with eight H200 GPUs and a global batch of 1,024. We use 128 samples per GPU unless memory requires gradient accumulation, as listed in Table 16. All assignment/objective ablations use $a = 1$.

We synchronize CUDA and record the slowest rank for each complete optimizer step, including generation, all training encoders, the objective, backward passes, communication, the optimizer, and native metric reduction. Compilation, statistics initialization, checkpointing, evaluation, and benchmark-log I/O are excluded. After at least ten warmup steps, we report the first twenty-step

mean with both coefficient of variation and relative drift between its two halves at most 5%. GPUs are not shared with other jobs.

Training state and precision. The assignment/objective ablations, MGFlow-W_2 ($K = 1+4$), and FD-Loss are timed from their pretrained-base initial training states. The MGFlow-KL ($K = 1+4+16$) column instead reports plateau measurements from 60-epoch checkpoints, restoring the model, optimizer, and all nine encoder–branch statistics queues. MGFlow uses BF16 neural forwards with its original covariance precision. FD-Loss uses the official implementation[1], retaining FP32 training, enabled TF32, and the original FD-statistics precision, without added autocast. These are end-to-end implementation timings, with the training states and precision settings specified above.

Table 16: **Gradient accumulation in the timing benchmark.** Entries are a; the per-GPU microbatch is $128/a$ and the global batch remains 1,024. Values above one follow a recorded CUDA OOM at the preceding larger microbatch.

Backbone	MGFlow-W_2	MGFlow-KL	Official FD-Loss
JiT-B	1	1	1
JiT-L	1	1	1
JiT-H	1	1	2
pMF-B	1	1	1
pMF-L	1	1	2
pMF-H	2	2	4

Full-batch FD-Loss with accumulation. At $a = 1$, we use the upstream training step. For $a > 1$, we evaluate one FD objective over the full global batch, then replay microbatches to backpropagate its feature gradients, as in Appendix C.1. Each global batch performs one optimizer update and one statistics update; replay overhead is included in the measured time.

Why use plateau KL measurements? LP solve time can change during training. On JiT-L, step time falls from 1.805 seconds at initialization to 1.009 seconds after 60 epochs, with CPU LP time falling from 1.031 to 0.241 seconds. The corresponding pMF-B/L times change little: $0.890/1.115$ seconds initially and $0.882/1.112$ seconds after 60 epochs.

I.2 Scaling the number of components

Computational cost. Consider one encoder with feature dimension d, batch size B, and K full-covariance components. Both variants form the Gaussian assignment costs and update component moments in $O(BKd^2)$ time. The component moment state requires $O(Kd^2)$ storage per encoder. The sample LP has BK variables and $B + K - 1$ independent equality constraints; denote its solve time by $T_{\mathrm{LP}}(B, K)$.

MGFlow-W_2 evaluates K paired Gaussian costs. Each requires a dense spectral decomposition for the matrix-square-root trace, giving $O(Kd^3)$ work. Evaluating all component pairs instead would require $O(K^2d^3)$ work before solving a $K \times K$ component transport LP (Delon & Desolneux, 2020). Fixed pairing removes this component-level LP, but retains the sample-assignment LP. MGFlow-KL uses Cholesky factorizations in $O(Kd^3)$ time and evaluates the Gaussian scores in $O(BKd^2)$ time. Global and paired KL scores have the same leading dense cost: pairing reuses R^* instead of computing two mixture posteriors, but still evaluates K component scores per feature. Excluding the shared neural-network computation, both paired variants therefore have cost

$$O(BKd^2 + Kd^3) + T_{\mathrm{LP}}(B, K). \tag{57}$$

For joint resolutions, these costs sum over $K \in \mathcal{K}$, while generated images and encoder features are shared.

[1] `https://github.com/Jiawei-Yang/FD-Loss`, commit `5c03b8112fec`.

Why KL is cheaper in practice. The cubic terms involve different matrix operations. For KL, we factor the covariance used in score evaluation as $C_{q,k} = L_{q,k}L_{q,k}^\top$ and compute each score by two triangular solves:

$$L_{q,k}u = \mu_{q,k} - z, \qquad L_{q,k}^\top s_{q,k}(z) = u. \tag{58}$$

Each generated-component factor is reused across the batch, and reference factors are cached. Cholesky factorization has a smaller computational constant than dense spectral decomposition; the subsequent solves need neither eigenvectors nor an explicit covariance inverse.

For W_2, even with the reference square root cached, each update forms $C_k = \Sigma_{p,k}^{1/2}\Sigma_{q,k}\Sigma_{p,k}^{1/2}$ through two dense matrix multiplications and computes $\mathrm{tr}(C_k^{1/2})$ from its eigenvalues. Gradients then pass through this spectral term, the matrix products, and the current batch's contribution to the covariance. KL instead computes the entire field with gradients stopped. Its backward pass differentiates only the squared regression loss with respect to the features, followed by the shared encoder and generator backward passes. KL therefore saves the spectral computation, its backward pass, and the memory needed to differentiate the current-batch covariance estimate. The practical advantage lies in these cheaper matrix operations and the shorter backward path, rather than a lower asymptotic order.

Larger Wasserstein objectives. We extend the timing protocol in Appendix I.1 to MGFlow-W_2 with $K = 16$ and $K = 1 + 4 + 16$ on another eight-H200 node, using SIM features and global batch size 1,024. Runs start from pretrained JiT-H and pMF-H weights. JiT-H uses a per-GPU batch of 128 without accumulation; pMF-H uses 64 with two accumulation steps after an OOM at 128, matching the accepted geometry in Table 16. Table 17 compares these measurements with Table 15. Relative to $K = 1 + 4$, the two larger Wasserstein objectives take $1.68\times$ / $1.95\times$ as long per step on JiT-H and $1.26\times$ / $1.40\times$ on pMF-H. This cost motivates our default $K = 1 + 4$ for MGFlow-W_2. MGFlow-KL uses $K = 1 + 4 + 16$ at 1.247 and 1.882 seconds per step on the two backbones in the plateau benchmark above.

Table 17: **Training cost of larger mixtures for MGFlow-W_2.** Seconds per optimizer update on eight H200 GPUs with SIM features and global batch size 1,024. All runs start from pretrained weights. The $K = 1 + 4$ result is from Table 15.

Backbone	$K = 1 + 4$	$K = 16$	$K = 1 + 4 + 16$
JiT-H	1.368	2.294	2.661
pMF-H	1.703	2.146	2.384

Sample support for component statistics. Increasing K also divides the statistics budget among more components. Under the LP constraints, component k receives total assignment mass $B\pi_k$ per batch and $N_0\pi_k$ over the N_0 initialization samples. We fit $K = 64$ full-covariance GMs to $N = 1{,}281{,}167$ real features for each encoder with seeds 3407–3410. All twelve EM runs reach the parameter-change stopping criterion. No Inception or MAE fit satisfies $\pi_{\max}/\pi_{\min} \le 10$, whereas all four SigLIP fits do (Table 18).

Inception is the most restrictive: its smallest component has only 978–1,358 units of reference responsibility mass in $d = 2{,}048$ dimensions. With $N_0 = 50{,}000$ and $B = 1{,}024$, the corresponding LP budgets are only 38–53 at initialization and 0.78–1.09 per batch. These small components provide weak support for estimating a full covariance. EMA accumulates information over time, but extending the averaging window also slows adaptation to the changing generator. This sample-support limitation motivates retaining $K = 16$ as the largest branch in MGFlow-KL rather than using $K = 64$ across the three encoders.

Table 18: **Component balance and sample support at $K = 64$.** Ranges across seeds 3407–3410. $N\pi_{\min}$ is the smallest component's reference responsibility mass; $N_0\pi_{\min}$ and $B\pi_{\min}$ are its LP allocation budgets for 50,000-sample initialization and a batch of 1,024. Passed denotes the number of fits with $\pi_{\max}/\pi_{\min} \leq 10$.

Encoder	$\pi_{\max}/\pi_{\min}$	$N\pi_{\min}$	$N_0\pi_{\min}$	$B\pi_{\min}$	Passed
Inception	50.58–75.54	978–1,358	38.2–53.0	0.78–1.09	0/4
MAE	15.44–16.66	2,766–3,112	107.9–121.5	2.21–2.49	0/4
SigLIP	3.02–5.64	7,767–10,146	303.1–396.0	6.21–8.11	4/4

Component count for text-to-image generation. Our text-to-image reference set contains only 301,706 images. At $K = 16$, the Inception fits have weight ratios $\pi_{\max}/\pi_{\min} = 79.61$ for joint matching and 72.31 for image-only matching. Their smallest components have reference responsibility masses $N\pi_{\min}$ of only 699 and 683 in 3,200- and 2,048-dimensional feature spaces, respectively. These small components provide insufficient sample support for stable full-covariance estimation. At $K = 4$, both weight ratios are below 6.2, and the smallest components each have over 25,000 units of reference responsibility mass. We therefore use $K = 1 + 4$ for both text-to-image variants.

J Results in Individual Evaluation Representations

Table 19 separates the training and held-out terms of FDr^6 for the configurations in Table 4. Each value is the encoder's Fréchet distance divided by its real-validation baseline.

Table 19: **Evaluation in training and held-out representations.** Normalized Fréchet distances after 100 epochs with SIM encoders; lower is better. MGFlow-W_2 uses $K = 1 + 4$, and MGFlow-KL uses $K = 1+4+16$. FD-Loss results (Yang et al., 2026a) are supplemented with 50,000-sample evaluations of the released SIM checkpoints for JiT-B/L and pMF-B.

		Training encoders			Held-out encoders		
Backbone	Method	Inception	MAE	SigLIP	ConvNeXt	DINOv2	CLIP
JiT-B	FD-Loss	0.60	4.22	3.34	1.33	5.13	18.78
	MGFlow-W_2	0.87	1.63	2.05	1.44	4.97	15.65
	MGFlow-KL	0.76	1.75	2.60	1.22	4.62	11.26
JiT-L	FD-Loss	0.46	0.66	1.98	0.87	2.63	12.91
	MGFlow-W_2	0.63	0.58	1.24	1.00	2.63	10.00
	MGFlow-KL	0.59	0.53	1.50	0.97	2.35	5.55
JiT-H	FD-Loss	0.45	0.43	1.68	0.86	2.10	10.37
	MGFlow-W_2	0.59	0.40	1.08	0.80	2.05	10.39
	MGFlow-KL	0.56	0.33	1.25	0.77	1.85	5.09
pMF-B	FD-Loss	0.51	1.86	5.37	0.77	4.10	8.51
	MGFlow-W_2	0.98	1.78	3.36	1.19	3.64	7.24
	MGFlow-KL	0.93	1.99	4.35	1.09	3.59	6.58
pMF-L	FD-Loss	0.47	0.56	3.03	0.57	2.21	5.68
	MGFlow-W_2	0.71	0.67	2.00	0.81	2.03	4.58
	MGFlow-KL	0.69	0.64	2.40	0.65	1.99	4.08
pMF-H	FD-Loss	0.46	0.35	2.46	0.57	1.74	5.77
	MGFlow-W_2	0.65	0.45	1.72	0.78	1.68	3.70
	MGFlow-KL	0.64	0.40	2.03	0.68	1.61	3.35

For pMF-L/H, MGFlow-KL improves DINOv2 and CLIP over FD-Loss, while FD-Loss retains lower ConvNeXt distance.

K QUALITATIVE IMAGENET RESULTS

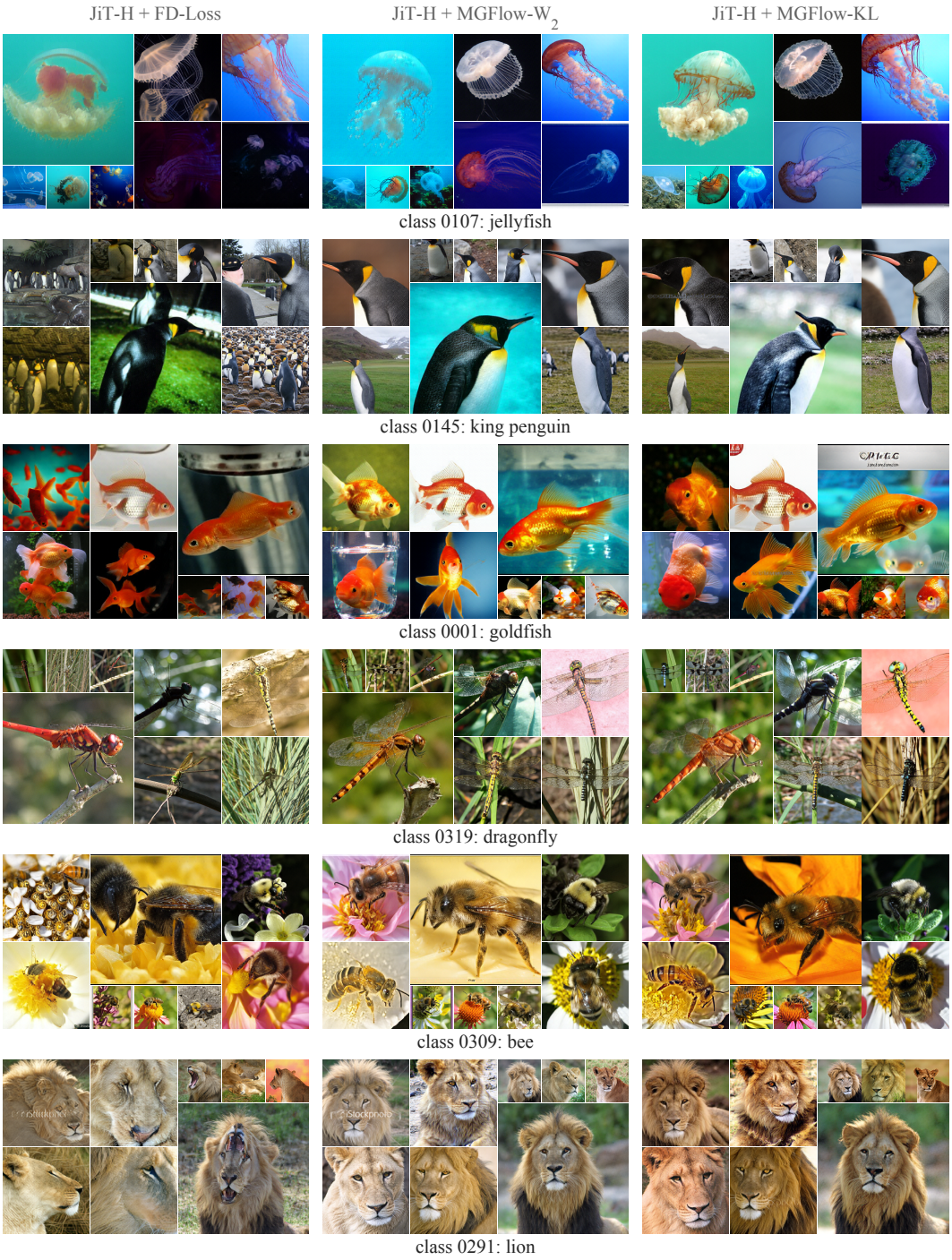


Figure 5: **Uncurated ImageNet 256 × 256 samples from JiT-H.** FD-Loss (Yang et al., 2026a) (left), MGFlow-W_2 (middle), and MGFlow-KL (right), all with one NFE and identical input noise at corresponding positions.

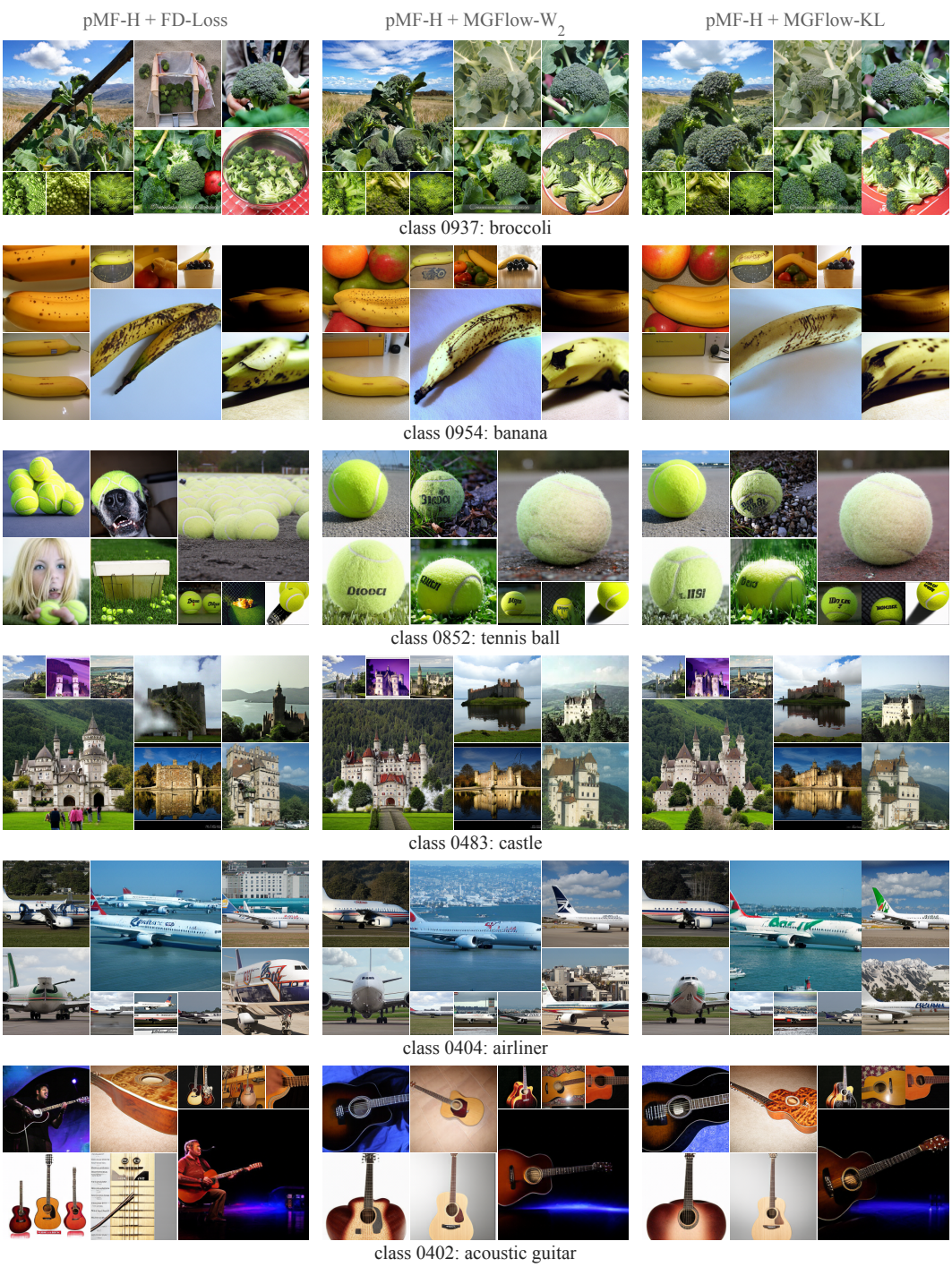


Figure 6: **Uncurated ImageNet 256 × 256 samples from pMF-H.** FD-Loss (Yang et al., 2026a) (left), MGFlow-W_2 (middle), and MGFlow-KL (right), all with one NFE and identical input noise at corresponding positions.

L Additional Text-to-Image Examples



Prompt. Giant glowing letters spelling "Nina Lu" rising above a neon-lit futuristic city skyline at night, low-angle wide shot, rain-slicked streets reflecting pink and blue light, cinematic sci-fi digital art.



Prompt. A beautiful young woman with long dark hair stands in a sunlit garden, soft golden light on her face, delicate flowers around her, shallow depth of field, elegant portrait photography.



Prompt. A fluffy cream-colored alpaca standing in a sunny green meadow, soft morning light, gentle expression, shallow depth of field, pastel color palette, whimsical children's book illustration style.



Prompt. A red sports car drifts through a rain-soaked stadium beneath a blazing and neat MGFlow neon sign; flying spray, wet reflections, low-angle sports photograph



Prompt. Claymation fashion portrait, woman in purple beret and orange ruffled collar posing beside a solid-colored awning, terracotta storefront, morning light casting soft shadows, medium close-up, stop-motion clay texture.



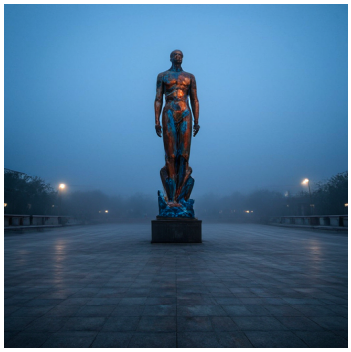
Prompt. A cute plush puppy with two large floppy ears, sitting on a clean table, seen from a side angle, soft daylight, shallow depth of field, cozy still-life photograph.



Prompt. A hidden valley revealed through a rocky archway, lush green terraces, mist drifting between cliffs, a still turquoise pool reflecting soft dawn light, wide landscape view, painterly digital art.



Prompt. Fashion portrait in watercolor: a Somali model in an indigo headwrap and ochre linen jacket poses against a sunlit mud-brick wall, three-quarter view, soft dry-brush washes, warm dusty palette, loose paper texture.



Prompt. Wide cinematic still of a towering sculpture of weathered copper and blue resin standing in a misty deserted plaza at soft twilight, camera pulled far back to show the full statue small within the empty square, calm deep blue tones.

Figure 7: **One-step text-to-image samples.** Selected 512×512 images from FLUX.2 [klein] 4B post-trained with MGFlow for 1,000 steps.