MITRAFLOW: LEARNING CORRECTIVE VELOCITY FIELDS ON MODEL-INDUCED TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion and flow-matching models generate images by progressively transforming noise into data samples. However, standard training supervises ideal noising or interpolation states, whereas sampling follows states induced by the model's own predictions. This mismatch creates exposure bias: prediction errors can shift subsequent states away from the training distribution, where recovery is not directly supervised. We introduce MITRAFLOW, a flow-matching objective that learns corrective velocity fields on model-induced trajectories. A controlled one-step rollout mixes the predicted and paired target velocities to construct a later state, at which the model learns the velocity needed to reach the paired data endpoint in the remaining time. Crucially, gradients propagate through both the rollout state and its corrective target, allowing the later loss to jointly optimize the correcting prediction and the earlier prediction that produced the state. We further align shallow features at the earlier state with detached deep features at the later state to regularize representations along the same transition. Our analysis shows that the corrective target compensates for the preceding velocity error, and ablations confirm that both controlled error exposure and cross-time gradients are essential. MITRAFLOW adds no inference cost and yields larger gains at fewer sampling steps. On ImageNet 256×256, it converges approximately $3\times$ faster than REPA and achieves an FID of 1.29 with guidance; the improvements also generalize to pixel and RAE spaces and to large-scale text-to-image generation.

1 INTRODUCTION

Diffusion and flow-matching models have become a central approach to high-quality conditional image generation (Ho et al., 2020; Lipman et al., 2023; Liu et al., 2023). Larger transformer backbones and representation-aligned training have further improved their quality and training efficiency (Peebles & Xie, 2023; Ma et al., 2024; Yu et al., 2025). Yet the standard training objective leaves an important gap between learning and generation. Flow matching supervises velocities at states drawn from prescribed interpolation paths, whereas a sampler advances using the model's own predictions. Errors in these predictions alter the states encountered at later times, where recovery from the preceding errors is not explicitly supervised. This training–sampling mismatch, often discussed as exposure bias (Ning et al., 2024), is illustrated in Figure 1 (upper left). Accurate regression on prescribed training states alone does not directly teach a model how to correct the deviations that arise during its own generation process.

Existing approaches mitigate this mismatch by training on perturbed inputs or model-induced states. Synthetic perturbations and slowed interpolations provide tractable approximations to sampling errors (Ning et al., 2023; Li et al., 2026), while model rollouts expose the learner to deviations produced by its own predictions (Qin et al., 2026; Huang et al., 2026). However, these deviations are themselves shaped by the evolving model: the prediction at one time determines the state on which a later prediction must operate. This temporal dependence motivates treating state construction and error recovery as a coupled learning problem. How can training expose the model to its own errors in a controlled manner while using later corrective feedback to improve the entire transition?

To couple state construction with error recovery, we introduce MITRAFLOW, which learns corrective velocity fields on controlled model-induced trajectories. As illustrated in Figure 1 (lower left), we let the model's predictions shift training states away from the ideal interpolation path while con-

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

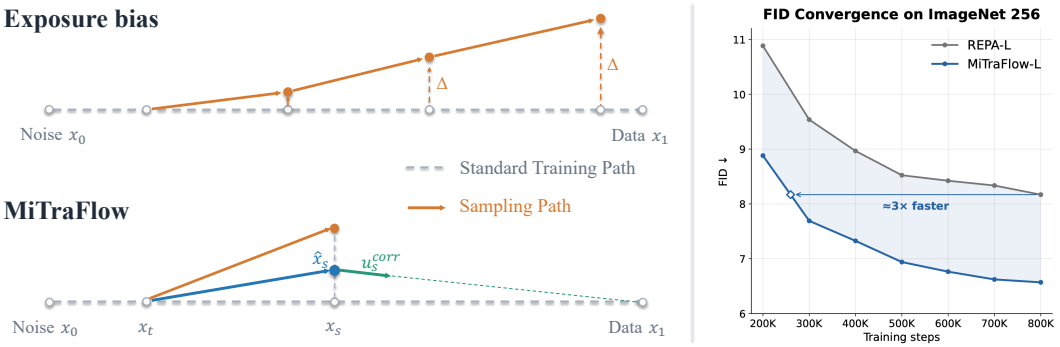


Figure 1: Motivation and convergence of MITRAFLOW. Upper left: standard training supervises ideal interpolation states, whereas sampling accumulates prediction error. Lower left: MITRAFLOW constructs a controlled model-induced state $\widehat{x}_s$ and learns a corrective velocity u_s^{corr} toward the paired data endpoint. Right: MITRAFLOW improves FID over the reproduced REPA baseline throughout training on ImageNet 256×256.

trolling the extent of deviation. The model then learns to steer these states toward the paired data endpoint. Corrective feedback passes through both the constructed state and its target, jointly improving the earlier prediction that produced the deviation and the later prediction that corrects it. The ideal and rollout states also form a natural pair for shallow–deep self-alignment (Jiang et al., 2026; Chefer et al., 2026), linking shallow features at the former to detached deep features at the latter to regularize representations along the same transition.

Our analysis and ablations suggest that model-induced errors can serve as useful training signals when their influence on training states is controlled and their correction is optimized across time. This perspective connects learning to recover from a deviation with learning the prediction that produces it, making the transition itself part of the learning process.

Experiments show that MITRAFLOW achieves approximately $3\times$ faster convergence than the REPA baseline and better generation quality under matched training compute. It achieves an FID of 1.29 with classifier-free guidance on ImageNet 256×256. Its benefits extend from class-conditioned to large-scale text-conditioned generation and across pixel, SD-VAE, and RAE spaces, supporting broad applicability. These gains require no changes to the inference architecture or sampling procedure, making MITRAFLOW a practical training improvement for existing generative models.

Our contributions are summarized as follows.

- **Corrective learning on model-induced states.** We pair ideal interpolation states with controlled model-induced rollout states and jointly optimize the predictions that produce and correct deviations through differentiable corrective velocity learning.
- **Shallow–deep alignment across the rollout transition.** We align shallow features at ideal states with detached deep features at rollout states, reusing the same controlled transition for representation learning.
- **Broad empirical validation.** We demonstrate approximately $3\times$ faster convergence relative to the REPA baseline and improved generation quality under matched training compute, with benefits across generation tasks, representation spaces, and sampling budgets.

2 RELATED WORK

Flow Matching. Flow Matching (Lipman et al., 2023) and Rectified Flow (Liu et al., 2023) provide simple objectives for learning velocity fields along conditional paths connecting noise and data. Stochastic interpolants provide a broader framework that connects deterministic flows and stochastic diffusion processes (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025), while conditional and multisample flow-matching variants use minibatch couplings to construct simpler transport paths (Tong et al., 2024; Pooladian et al., 2023). Rectified-flow transformers have further demonstrated



Figure 2: 512×512 text-to-image samples generated from prompts of different lengths. The corresponding prompts are provided in Appendix D.1.

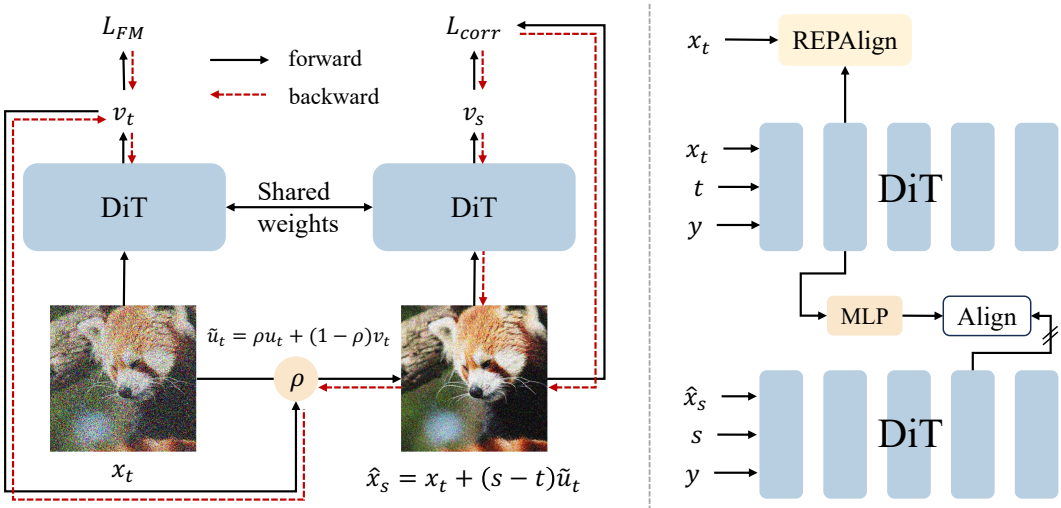


Figure 3: Selected class-to-image samples generated by MITRAFLOW on ImageNet at 256×256 resolution. Samples are produced from EMA weights using 250-step Euler–ODE sampling and classifier-free guidance with scale 1.8 over the interval $[0.3, 1.0]$.

the scalability of these objectives to high-resolution text-to-image generation (Esser et al., 2024). Related approaches explore alternative prediction targets for efficient generation: Consistency Models (Song et al., 2023) learn mappings from noisy states to data, while MeanFlow (Geng et al., 2025) models average velocities over time intervals. MITRAFLOW retains the instantaneous velocity parameterization and standard sampling procedure. Its focus is on improving the training states and supervision so that the model learns to recover from deviations caused by its own predictions.

Training on Model-Induced States. Prior work has identified the training–sampling mismatch and explored mitigations via input augmentation and trajectory-based training. Input Perturbation (Ning et al., 2023) perturbs training inputs to simulate prediction errors, and multi-step denoising scheduled sampling models errors accumulated over several steps (Ren et al., 2024). Sampling-side methods shift inference timesteps to match model-generated states (Li et al., 2024a). MixFlow (Li et al., 2026) uses slowed interpolation mixtures for post-training, while SOAR (Qin et al., 2026) learns correction from detached model rollouts followed by re-noising. MITRAFLOW instead differentiates through the constructed transition, letting later errors update the earlier predictions producing these states. Most closely related, DEFAR (Huang et al., 2026) combines directional rectification with frequency compensation. MITRAFLOW instead supervises the full corrective velocity with a simple objective, yielding substantial gains over representation-aligned baselines. We further evaluate its scalability on larger models and large-scale text-to-image generation.

Representation-Aligned Generative Training. Representation supervision offers another route to improving generative training. REPA (Yu et al., 2025) aligns denoiser features with pretrained visual encoder representations. SRA (Jiang et al., 2026) and Self-Flow (Chefer et al., 2026) further explore representation learning without external supervision. REG entangles image latents with pretrained semantic tokens, while Self-Transcendence constructs internal guidance features (Wu et al., 2025; Sun et al., 2026). REPA-E instead jointly adapts the latent autoencoder and generative model (Leng et al., 2025). Building on the self-alignment principle, MITRAFLOW integrates alignment into the dual-state controlled rollout, unifying representation learning and trajectory correction



(a) Differentiable Rollout and Velocity Correction

(b) Shallow-Deep Alignment

Figure 4: Method overview. (a) A differentiable controlled rollout mixes the predicted and paired velocities to construct a later state, where time-calibrated corrective supervision propagates gradients through the transition. (b) Shallow–deep feature alignment trains projected shallow features at the earlier state to match detached deep features at the rollout state.

within one training transition. This coupling reuses the rollout evaluations and remains compatible with external supervision such as REPA.

3 Methodology

In this section, we first formulate the training–sampling mismatch in flow matching, then introduce controlled rollout correction and shallow–deep feature alignment. Finally, we analyze the endpoint interpretation and residual-compensation mechanism of the corrective objective.

3.1 Preliminaries and Problem Formulation

Let $x_0 \sim p_0$ denote a noise sample, $x_1 \sim p_{\text{data}}$ a data sample, and c an optional condition. We consider the linear conditional probability path used in conditional flow matching (Lipman et al., 2023; Liu et al., 2023; Tong et al., 2024)

$$x_t = (1-t)x_0 + tx_1, \qquad u_t = \frac{\mathrm{d}x_t}{\mathrm{d}t} = x_1 - x_0, \tag{1}$$

whose paired target velocity is constant in time. Conditional flow matching trains $v_\theta(x_t, t, c)$ by minimizing the velocity regression objective

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\left[\|v_\theta(x_t, t, c) - u_t\|_2^2\right], \tag{2}$$

Equation 2 presents the model only with states sampled from the ideal conditional path. At generation time, however, a numerical solver uses the model's preceding predictions, so a velocity error changes the input states encountered at subsequent times. The model is consequently asked to predict on a state distribution on which it receives no direct flow-matching supervision, producing the training–sampling discrepancy studied as exposure bias in diffusion and flow models (Ning et al., 2023; 2024; Ren et al., 2024). We seek a training objective that (i) exposes the model to such model-induced states without allowing unstable early predictions to dominate training, and (ii) teaches it how to compensate for the error that produced each state. MiTraFlow realizes both goals with a single controlled transition between two sampled times.

3.2 Differentiable Controlled Rollout and Corrective Velocity Learning

As shown in Figure 4(a), for each training pair, we sample two independent times and order them so that $t < s$. We construct a controlled one-step rollout via an Euler step that mixes the predicted and paired velocities. After evaluating the model at the ideal state x_t, we form the mixed velocity

$$\widetilde{u}_t = \rho u_t + (1-\rho) v_\theta(x_t, t, c), \qquad \rho \in [0,1], \tag{3}$$

and take one Euler step to obtain

$$\widehat{x}_s = x_t + (s-t)\widetilde{u}_t. \tag{4}$$

The coefficient ρ controls the rollout mixture: $\rho = 1$ recovers the ideal paired state x_s, whereas $\rho = 0$ is a fully model-induced step. An intermediate value exposes realistic prediction errors while anchoring the resulting state to the paired path.

Merely applying the original target u_t at $\widehat{x}_s$ ignores the displacement accumulated before time s. Instead, we use the standard linear flow state-to-velocity transformation (derived from a general affine path in Appendix A.1) at $\widehat{x}_s$ and supervise the corrective velocity required to traverse the remaining distance to the paired endpoint,

$$u_s^{\mathrm{corr}} = \frac{x_1 - \widehat{x}_s}{\bar{\tau}_s}, \qquad \bar{\tau}_s = \max(1-s, \varepsilon), \tag{5}$$

and define

$$\mathcal{L}_{\mathrm{corr}} = \mathbb{E}\Big[\| v_\theta(\widehat{x}_s, s, c) - u_s^{\mathrm{corr}} \|_2^2 \Big]. \tag{6}$$

We set $\varepsilon = 0.05$ following JiT (Li & He, 2026). Crucially, $\widehat{x}_s$ and u_s^{corr} are not detached: the corrective loss therefore updates both the prediction at s and, through the constructed transition and moving target, the prediction at t. The resulting velocity objective is

$$\mathcal{L}_{\mathrm{vel}} = \lambda_t \mathcal{L}_{\mathrm{FM}} + \lambda_s \mathcal{L}_{\mathrm{corr}}. \tag{7}$$

It uses two denoiser evaluations during training and makes no change to the sampling algorithm or inference-time network, thus adding no inference cost.

3.3 Shallow–Deep Feature Alignment

As illustrated in Figure 4(b), the same transition also provides a correspondence between an earlier ideal state and a later state containing the model's own error. We use it to regularize how representations develop through both network depth and flow time, following the general shallow–deep alignment principle of SRA and Self-Flow (Jiang et al., 2026; Chefer et al., 2026). Let $h_\theta^\ell(x, t, c)$ denote the token features after denoiser block ℓ, and choose a shallow block ℓ_{sh} and a deeper block $\ell_{\mathrm{dp}} > \ell_{\mathrm{sh}}$. A learnable MLP P_ϕ maps the shallow representation to the denoiser feature dimension. With $\nu(z) = z/\|z\|_2$ denoting token-wise normalization (epsilon-stabilized in the implementation), we form the projected shallow and detached deep representations

$$q_t = \nu\Big(P_\phi\Big(h_\theta^{\ell_{\mathrm{sh}}}(x_t, t, c) \Big) \Big), \tag{8}$$

$$r_s = \mathrm{sg}\Big[\nu\Big(h_\theta^{\ell_{\mathrm{dp}}}(\widehat{x}_s, s, c) \Big) \Big], \tag{9}$$

where sg is stop-gradient; the shallow and deep features are computed by the same denoiser with shared parameters θ. The shallow–deep alignment loss is the mean token-wise cosine distance

$$\mathcal{L}_{\mathrm{align}} = \mathbb{E}\big[1 - q_t^\top r_s \big]. \tag{10}$$

Thus, shallow features at the earlier state are trained to anticipate the deep features that the denoiser forms at the later rollout state. The target is coupled to the controlled trajectory in the forward pass but, because it is detached, this loss updates only the shallow branch and projector. The alignment objective reuses the two denoiser evaluations already required by Equation 7.

Combining the output-space correction with the representation-space constraint gives

$$\mathcal{L}_{\textsc{MiTraFlow}} = \mathcal{L}_{\mathrm{vel}} + \lambda_{\mathrm{align}} \mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{ext}} \mathcal{L}_{\mathrm{ext}}, \tag{11}$$

where $\mathcal{L}_{\mathrm{ext}}$ denotes an optional external representation objective, such as the REPA loss retained in our experiments (Yu et al., 2025). Neither alignment branch is retained at inference.

Table 1: Component ablation on ImageNet 256×256. Controlled rollout correction and shallow–deep alignment are added cumulatively to REPA. Bold denotes the best result in each metric.

Method	Iters	FID ↓	sFID ↓	FDr^6 ↓	IS ↑	Prec. ↑	Rec. ↑
REPA (Yu et al., 2025)	400K	8.97	4.89	17.20	118.1	0.68	0.70
	700K	8.43	5.23	15.84	126.4	0.68	0.71
+ controlled rollout correction	400K	7.23	5.01	15.82	**133.7**	0.69	0.73
+ shallow–deep alignment	400K	**6.91**	**4.77**	**15.54**	133.0	**0.70**	**0.79**

Table 2: Ablation of controlled rollout and cross-time gradient paths. The two branch columns indicate whether gradients reach the earlier prediction through the target branch u_s^{corr} or the prediction branch $v_\theta(\widehat{x}_s, s, c)$ of $\mathcal{L}_{\text{corr}}$. Detaching their shared input $\widehat{x}_s$ blocks both branches; the EMA variant constructs rollout states using an EMA model.

Variant	ρ	Target branch u_s^{corr}	Prediction branch $v_\theta(\widehat{x}_s, s, c)$	Iters	FID ↓	sFID ↓	FDr^6 ↓	IS ↑	Prec. ↑	Rec. ↑
(a) Rollout mixture										
REPA (Yu et al., 2025)	–	–	–	700K	8.43	5.23	15.84	126.4	0.68	0.71
Ideal-path rollout	1.0	–	–	400K	9.18	5.64	17.66	124.2	0.67	0.64
Controlled rollout	0.9	✓	✓	400K	**7.23**	**5.01**	**15.82**	**133.7**	**0.69**	**0.73**
Controlled rollout	0.7	✓	✓	400K	8.02	5.27	17.21	125.6	0.69	0.65
Pure model rollout	0.0	✓	✓	400K	28.20	30.62	39.95	59.8	0.55	0.47
(b) Gradient variants at $\rho = 0.9$										
Fully differentiable	0.9	✓	✓	400K	**7.23**	**5.01**	**15.82**	**133.7**	**0.69**	0.73
Detach u_s^{corr}	0.9	–	✓	400K	8.63	6.00	16.78	125.9	0.68	**0.74**
Detach $\widehat{x}_s$	0.9	–	–	400K	8.24	5.49	16.35	125.9	0.68	0.65
EMA rollout teacher	0.9	–	–	400K	8.38	5.53	16.44	127.2	0.68	**0.74**

3.4 Analysis of Learning on Controlled Trajectories

Velocity-to-endpoint interpretation. Under the velocity-prediction parameterization, the linear path gives the standard change of variables $x_1 = x_s + (1-s)v(x_s, s, c)$. Accordingly, a velocity prediction at time s can be converted into the implied endpoint prediction $F_\theta(x_s, s, c) = x_s + (1-s)v_\theta(x_s, s, c)$. Without denominator clipping,

$$\|v_\theta(\widehat{x}_s, s, c) - u_s^{\text{corr}}\|_2^2 = \frac{\|F_\theta(\widehat{x}_s, s, c) - x_1\|_2^2}{(1-s)^2}. \tag{12}$$

Thus the corrective velocity loss equals a time-weighted endpoint-error objective.

Why train on a rollout state? For a fixed pair (x_0, x_1), let $e_t = v_\theta(x_t, t, c) - u_t$ be the early velocity residual. The controlled transition satisfies

$$\widehat{x}_s - x_s = (s-t)(1-\rho)e_t. \tag{13}$$

It therefore presents the model with a later rollout state displaced by its own prediction error, as occurs during inference, rather than only with an ideal interpolation state. Supervising the corrective velocity at $\widehat{x}_s$ directly trains recovery from this model-induced state perturbation. Moreover, because $\widehat{x}_s$ is not detached, the corrective loss updates both the velocity evaluated at s and the earlier prediction that created its input at t; Section 4.2 examines these gradient paths empirically.

4 Experiments

4.1 Experimental Setup

We evaluate MiTraFlow on C2I and T2I generation. Our primary C2I experiments use ImageNet (Deng et al., 2009) at 256×256 resolution, with DiT-L and DiT-XL (Peebles & Xie, 2023) operating in the SD-VAE latent space (Rombach et al., 2022). We additionally evaluate the method in pixel space with PixNerd (Wang et al., 2026a) and in RAE latent space with DiT^{DH}-XL (Zheng et al.,

Table 3: Representation-space generalization at 80 epochs without CFG. Our method generalizes well across different representation spaces.

Space	Method	FID ↓	sFID ↓	FDr^6 ↓	IS ↑	Prec. ↑	Rec. ↑
Pixel	PixNerd (Wang et al., 2026a)	29.07	**7.79**	33.9	54.0	0.51	0.66
	+ Ours	**23.56**	9.47	**33.9**	**60.8**	**0.54**	**0.67**
RAE	RAE (Zheng et al., 2026)	2.16	–	–	214.8	**0.82**	0.59
	+ Ours	**1.86**	**5.17**	**4.5**	**226.5**	**0.82**	**0.60**

Table 4: FID on 10K samples across sampling steps; parentheses show the reduction over REPA.

		Sampling steps				
Sampler	Method	16	32	64	128	250
SDE	REPA	27.5	16.4	13.2	11.9	11.3
	+ Ours	**21.2** (−22.9%)	**13.5** (−17.7%)	**11.5** (−12.9%)	**11.0** (−7.6%)	**10.6** (−6.2%)
ODE	REPA	17.5	13.5	12.1	11.6	11.4
	+ Ours	**14.1** (−19.4%)	**11.8** (−12.6%)	**11.0** (−9.1%)	**10.6** (−8.6%)	**10.5** (−7.9%)

2026) to examine its applicability across representations. For T2I generation, we use an approximately 846M-parameter denoiser conditioned on a frozen Qwen3-1.7B text encoder (Yang et al., 2025). We pretrain the denoiser on 42.7M image–text pairs assembled from BLIP3-o, DenseFusion-1M, and two FLUX-generated collections (Chen et al., 2025; Li et al., 2024b). Text-to-image fine-tuning additionally uses Echo-4o and JourneyDB data (Ye et al., 2025; Sun et al., 2023).

We use REPA (Yu et al., 2025) as the primary baseline for our SD-VAE experiments and retain its external representation-alignment objective when adding MITRAFLOW. For all C2I experiments, we initialize MITRAFLOW from the corresponding baseline checkpoint trained for 100K iterations.

For evaluation, we report FID (Heusel et al., 2017), sFID (Nash et al., 2021), Inception Score (Salimans et al., 2016), precision and recall (Kynkäänniemi et al., 2019), and FDr^6 (Yang et al., 2026) where available for C2I, and GenEval (Ghosh et al., 2023) and the DPG score on DPG-Bench (Hu et al., 2024) for T2I. Unless otherwise specified, evaluation uses EMA model weights. Ablation studies use generation without classifier-free guidance (Ho & Salimans, 2022), while the main C2I comparison reports unguided and guided results separately. Our guided C2I results use scale 1.8 over the interval $[0.3, 1.0]$ (Kynkäänniemi et al., 2024). For sampling, we use 250-step fixed-step Euler integration of the learned flow ODE by default (Lipman et al., 2023; Chen et al., 2018). The ODE and SDE sampling sweep follows the corresponding deterministic and stochastic generative formulations (Song et al., 2021; Albergo et al., 2025; Ma et al., 2024).

4.2 ABLATION STUDIES

For the ablation studies, we initialize MITRAFLOW from REPA checkpoint at 100K iterations and train for a further 300K iterations, yielding a cumulative training length of 400K iterations. Rollout requires two denoiser evaluations per iteration, doubling the per-iteration compute relative to the REPA baseline. Counting each rollout iteration as two baseline iterations, the total training budget is therefore $100\text{K} + 2 \times 300\text{K} = 700\text{K}$ baseline-equivalent iterations. We thus compare MITRAFLOW at 400K iterations with the baseline at 700K iterations under matched compute.

Contributions of individual components. Table 1 shows progressive gains as controlled rollout correction and shallow–deep alignment are added to REPA. Rollout correction establishes the main improvement in generation quality while also increasing distributional coverage. Under the compute accounting above, this suggests that learning to recover from model-induced errors makes better use of the additional evaluations than extending baseline training alone. Shallow–deep alignment builds on these gains, further improving FID, FDr^6, precision, and recall while maintaining a comparable IS. The consistent gains across these complementary metrics support a robust overall improvement in quality and coverage as the two components are combined. Together, they reduce FID by 18.0%

Table 5: Class-conditional generation on ImageNet 256×256. Gray rows denote models using alternative representation spaces.

			256×256, w/o CFG					256×256, w/ CFG				
Method	Params	Epochs	FID ↓	sFID ↓	IS ↑	Prec. ↑	Rec. ↑	FID ↓	sFID ↓	IS ↑	Prec. ↑	Rec. ↑
ADM (Dhariwal & Nichol, 2021)	554M	400	10.94	6.02	101.0	0.69	0.63	3.94	6.14	215.8	0.83	0.53
RIN (Jabri et al., 2023)	410M	480	3.42	–	182.0	–	–	–	–	–	–	–
REPA-L (Yu et al., 2025)	458M	140	8.43	5.23	126.4	0.68	0.71	–	–	–	–	–
+ MITRAFLOW	458M	80	6.91	4.77	133.0	0.70	0.79	1.64	4.24	267.7	0.79	0.62
+ MITRAFLOW	458M	200	6.09	4.76	148.6	0.70	0.67	1.53	4.18	288.9	0.79	0.62
VA-VAE (Yao et al., 2025)	675M	800	2.05	4.37	207.7	0.77	0.66	1.25	4.15	295.3	0.80	0.65
REPA-E (Leng et al., 2025)	675M	800	1.69	4.17	219.3	0.77	0.66	1.12	4.09	302.9	0.79	0.66
RAE (Zheng et al., 2026)	839M	800	1.51	5.31	242.9	0.79	0.63	1.13	4.74	262.6	0.78	0.67
REGLUE (Petsangourakis et al., 2026)	677M	160	–	–	–	–	–	1.53	4.26	320.4	0.78	0.65
Hi-DiT (Shen et al., 2026)	689M	/	1.66	–	213.6	–	–	1.06	–	296.6	–	–
DiT (Peebles & Xie, 2023)	675M	1400	9.62	6.85	121.5	0.67	0.67	2.27	4.60	278.2	0.83	0.57
MaskDiT (Zheng et al., 2024)	675M	1600	5.69	10.34	177.9	0.74	0.60	2.28	5.67	276.6	0.80	0.61
SiT (Ma et al., 2024)	675M	1400	8.61	6.32	131.7	0.68	0.67	2.06	4.50	270.3	0.82	0.59
MDTv2 (Gao et al., 2024)	675M	1080	–	–	–	–	–	1.58	4.52	314.7	0.79	0.65
REG (Wu et al., 2025)	677M	800	1.80	4.59	230.8	0.77	0.66	1.36	4.25	299.4	0.77	0.66
MixFlow (Li et al., 2026)	675M	200	5.00	4.87	171.4	0.72	0.67	1.22	4.49	313.4	0.80	0.64
Self-Transcendence (Sun et al., 2026)	675M	400	–	–	–	–	–	1.44	4.85	311.3	0.79	0.66
REPA-XL (Yu et al., 2025)	675M	800	5.84	5.79	158.7	0.70	0.68	1.28	4.68	305.7	0.79	0.64
+ MITRAFLOW	675M	80	6.24	4.78	142.1	0.71	0.63	1.51	4.16	278.3	0.79	0.63
+ MITRAFLOW	675M	180	5.48	4.84	157.1	0.70	0.67	1.45	4.24	296.8	0.79	0.63
DDT-XL (Wang et al., 2026b)	675M	400	6.27	–	154.7	0.68	0.69	1.26	–	310.6	0.79	0.65
+ MITRAFLOW	675M	250	4.84	4.93	172.5	0.71	0.69	1.29	4.19	304.4	0.79	0.65

relative to REPA-700K, with alignment reusing the same two denoiser evaluations. These results support the complementarity of corrective velocity learning and representation alignment.

Rollout mixture and gradient paths. Table 2(a) isolates the value of exposing the model to its own errors. The ideal-path variant ($\rho = 1$) adds supervision at a second state but does not improve on REPA-700K, indicating that a second supervised evaluation alone is insufficient. A small model contribution ($\rho = 0.9$) performs best among the tested mixtures, whereas removing the paired-path anchor ($\rho = 0$) severely degrades generation. Together, these results support controlled error exposure: the rollout should introduce deviations that the model can learn to correct while remaining anchored to the paired trajectory.

In $\mathcal{L}_{\text{corr}}$, both the prediction $v_\theta(\widehat{x}_s, s, c)$ and the target u_s^{corr} depend on $\widehat{x}_s$, allowing gradients to propagate back to the earlier prediction through two cross-time branches. The gradient ablations in Table 2(b) suggest that these branches are most useful when optimized together. Detaching u_s^{corr} retains only the prediction branch, yet performs worse than detaching $\widehat{x}_s$ and blocking both branches. Thus, retaining a single branch is not necessarily beneficial; the strongest result comes from coupling the corrective prediction with its state-dependent target. This is consistent with the objective's joint optimization of the earlier prediction that constructs the state and the later prediction that corrects it.

An EMA rollout teacher, similar to the detached construction in SOAR (Qin et al., 2026), also falls short of the fully differentiable variant. In this comparison, using an averaged teacher does not recover the benefit of joint optimization. These findings motivate the default configuration: $\rho = 0.9$ with both cross-time gradient branches enabled.

4.3 GENERALIZATION AND SAMPLING EFFICIENCY

Generalization across representation spaces. The improvements in Table 3 extend beyond the SD-VAE setting to both pixel-space PixNerd and the RAE latent baseline. Lower FID together with higher IS and recall in both settings suggests that corrective supervision can benefit different representations, rather than depending on the particular geometry of SD-VAE latents. The higher pixel-space sFID may reflect local texture perturbations introduced by rollout that endpoint-based correction does not fully resolve.

Performance across sampling steps. Table 4 shows that MITRAFLOW improves FID across all evaluated step counts under both SDE and ODE sampling, with proportionally larger gains when fewer steps are used. At 16 steps, the relative FID reductions are 22.9% and 19.4% for SDE and

ODE, respectively, compared with 6.2% and 7.9% at 250 steps. This pattern suggests that corrective training is particularly beneficial when the sampling budget is limited.

These gains translate into improved sampling efficiency: at 64 steps, MITRAFLOW surpasses REPA at 128 steps with SDE sampling and at 250 steps with ODE sampling. By reaching comparable or better quality with fewer model evaluations, MITRAFLOW improves the quality–compute trade-off using the same sampling procedures. The stronger relative gains at low step counts further suggest that learning to correct model-induced deviations could be a useful foundation for few-step generation, motivating targeted exploration of this regime.

4.4 MAIN RESULTS

Qualitative results. Figures 2 and 3 illustrate the visual quality of MITRAFLOW across text- and class-conditioned generation. The samples exhibit clear object structure and fine texture detail across diverse subjects, while the text-conditioned examples also capture varied materials, lighting, and imaginative scene compositions. These examples highlight the method's ability to generate visually coherent images across both tasks.

Class-to-image generation. Table 5 shows that MITRAFLOW improves unguided FID over the corresponding baselines across model sizes, with further gains from continued training. With classifier-free guidance, it achieves comparable generation quality to established methods in the SD-VAE latent space. The compute-matched comparison is provided in Section 4.2.

Text-to-image generation. Table 6 provides a broader view of MITRAFLOW in large-scale text-to-image generation while disentangling the effects of its two components.

Specifically, the full MITRAFLOW (last row) achieves stronger overall results than REPA: it improves DPG while matching GenEval during pretraining, and improves both benchmarks after finetuning. Rollout correction alone yields more modest and stage-dependent gains. One possible reason is that correcting model-induced trajectory errors does not directly address all aspects of text–image alignment and compositional reasoning assessed by these benchmarks. Shallow–deep alignment may complement this correction by strengthening the learned representations, allowing the combined method to deliver more consistent improvements over the baseline.

Table 6: Text-to-image generation before and after finetuning. Rows cumulatively add controlled rollout correction (Rollout) and shallow–deep alignment (Alignment).

	Pretraining		Finetuning	
Method	GenEval ↑	DPG ↑	GenEval ↑	DPG ↑
REPA (Yu et al., 2025)	67.1	79.8	77.2	80.7
+ Rollout	63.7	79.9	**79.1**	80.7
+ Alignment	**67.1**	**81.1**	77.8	**81.9**

5 CONCLUSION

We presented MITRAFLOW, which couples ideal interpolation states with controlled model-induced rollout states to learn corrective velocity fields. Differentiable corrective feedback jointly optimizes the predictions that produce and correct deviations, while shallow–deep self-alignment supports representation learning across the same transition. Our analysis and experiments show that appropriately controlled prediction errors can provide useful learning signals, improving training efficiency and generation quality across tasks and representation spaces without additional inference cost. These findings highlight the value of jointly learning how training states are constructed and how the model recovers from them.

Future work could adapt the degree of model-induced deviation to training progress and extend corrective learning to multiple rollout steps, capturing longer-term error propagation while managing training cost. Another direction is to better understand how self-alignment affects the balance between sample quality and distributional coverage. More broadly, MITRAFLOW offers a practical foundation for training generative models to learn from the consequences of their own predictions.

AI USE STATEMENT

AI-based tools were used during the preparation of this manuscript for limited purposes, including language editing, polishing, and proofreading; retrieval and discovery of related literature; and assisting with research ideation and code implementation. In addition, AI-based vision–language models were used to generate captions for images used as training data in our text-to-image experiments; these captions serve only as training inputs and are not presented as results or findings. All retrieved references were read and verified by the authors, and all AI-suggested ideas were critically evaluated and refined by the authors. All mathematical derivations, theoretical results, algorithmic designs, and scientific conclusions were checked for correctness by the authors. All AI-assisted code was manually reviewed, verified, and validated by the authors, and all experimental implementations and empirical analyses were independently checked by the authors.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. In *International Conference on Machine Learning*, 2026.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, Le Xue, Caiming Xiong, and Ran Xu. BLIP3-o: A family of fully open unified multimodal models—architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, volume 235, pp. 12606–12633, 2024.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. MDTv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2024.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 75460–75482, 2025.

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*, volume 36, pp. 52132–52152, 2023.

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.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Guanbo Huang, Jingjia Mao, Fanding Huang, Fengkai Liu, Xiangyang Luo, Yaoyuan Liang, Jiasheng Lu, Xiaoe Wang, Pei Liu, Ruiliu Fu, Ruqi Huang, and Shao-Lun Huang. Exposure bias can alleviate itself via directional and frequency rectification in flow matching. In *European Conference on Computer Vision*, 2026.

Allan Jabri, David J. Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *International Conference on Machine Learning*, volume 202, pp. 14569–14589, 2023.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. Representation alignment for diffusion transformers without external components. In *International Conference on Learning Representations*, 2026.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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.

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

Hui Li, Jiayue Lyu, Fu-Yun Wang, Kaihui Cheng, Siyu Zhu, and Jingdong Wang. MixFlow training: Alleviating exposure bias with slowed interpolation mixture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9095–9105, 2026.

Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. In *International Conference on Learning Representations*, 2024a.

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 (CVPR)*, pp. 36115–36125, June 2026.

Xiaotong Li, Fan Zhang, Haiwen Diao, Yueze Wang, Xinlong Wang, and Ling-Yu Duan. DenseFusion-1M: Merging vision experts for comprehensive multimodal perception. In *Advances in Neural Information Processing Systems*, volume 37, pp. 18535–18556, 2024b.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

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

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations. In *International Conference on Machine Learning*, volume 139, pp. 7958–7968, 2021.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *International Conference on Machine Learning*, volume 202, pp. 26245–26265, 2023.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *International Conference on Learning Representations*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Giorgos Petsangourakis, Christos Sgouropoulos, Bill Psomas, Theodoros Giannakopoulos, Giorgos Sfikas, and Ioannis Kakogeorgiou. REGLUE your latents with global and local semantics for entangled diffusion. In *European Conference on Computer Vision*, 2026.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, volume 202, pp. 28100–28127, 2023.

You Qin, Linqing Wang, Hao Fei, Roger Zimmermann, Liefeng Bo, Qinglin Lu, and Chunyu Wang. SOAR: Self-correction for optimal alignment and refinement in diffusion models. *arXiv preprint arXiv:2604.12617*, 2026.

Zhiyao Ren, Yibing Zhan, Liang Ding, Gaoang Wang, Chaoyue Wang, Zhongyi Fan, and Dacheng Tao. Multi-step denoising scheduled sampling: Towards alleviating exposure bias for diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4667–4675, 2024.

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

Tim Salimans, 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.

Yedong Shen, Yehao Li, Yingwei Pan, Yanyong Zhang, and Ting Yao. Hi-DiT: Hybrid latent-pixel diffusion transformer for image generation. In *European Conference on Computer Vision*, pp. 194–212, 2026.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, volume 202, pp. 32211–32252, 2023.

Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, Jifeng Dai, Yu Qiao, Limin Wang, and Hongsheng Li. JourneyDB: A benchmark for generative image understanding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49659–49678, 2023.

Lingchen Sun, Rongyuan Wu, Zhengqiang Zhang, Ruibin Li, Yujing Sun, Shuaizheng Liu, and Lei Zhang. Self-transcendence: Is external feature guidance indispensable for accelerating diffusion transformer training? In *European Conference on Computer Vision*, pp. 417–434, 2026.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion. In *International Conference on Learning Representations*, 2026a.

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

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*, volume 38, pp. 7714–7743, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

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

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

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.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026.

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

A Additional Derivations

A.1 State-to-Velocity Transformation for a General Affine Path

Consider the general affine conditional probability path

$$x_r = \alpha_r x_1 + \sigma_r x_0, \qquad u_r = \frac{\mathrm{d}x_r}{\mathrm{d}r} = \dot{\alpha}_r x_1 + \dot{\sigma}_r x_0, \tag{14}$$

where α_r and σ_r are differentiable schedules. Given a state x at time r and its paired endpoint x_1, we eliminate the unknown source state using $x_0 = (x - \alpha_r x_1)/\sigma_r$ (for $\sigma_r \neq 0$). Substitution into Equation 14 yields a deterministic state-to-velocity transformation. We denote this map by $\mathcal{T}_r(x, x_1)$: it maps a state x and its paired endpoint x_1 to the instantaneous velocity consistent with the affine path at time r. It depends only on the path schedules and has no learnable parameters:

$$\mathcal{T}_r(x, x_1) = \frac{\dot{\sigma}_r}{\sigma_r} x + \left(\dot{\alpha}_r - \frac{\dot{\sigma}_r}{\sigma_r}\alpha_r\right) x_1 \tag{15}$$

$$= \frac{\dot{\sigma}_r x + (\sigma_r \dot{\alpha}_r - \dot{\sigma}_r \alpha_r) x_1}{\sigma_r}. \tag{16}$$

For a state on the affine path, $\mathcal{T}_r(x_r, x_1) = u_r$.

For the linear path used in this paper, $\alpha_r = r$ and $\sigma_r = 1 - r$, so $\dot{\alpha}_r = 1$ and $\dot{\sigma}_r = -1$. Equation 16 therefore simplifies to

$$\mathcal{T}_r(x, x_1) = -\frac{x}{1-r} + \left(1 + \frac{r}{1-r}\right) x_1 = \frac{x_1 - x}{1 - r}. \tag{17}$$

Applying this transformation at $r = s$ to the controlled rollout state $x = \widehat{x}_s$ gives exactly the corrective target in the main text,

$$u_s^{\mathrm{corr}} = \mathcal{T}_s(\widehat{x}_s, x_1) = \frac{x_1 - \widehat{x}_s}{1 - s}. \tag{18}$$

In practice, replacing $1 - s$ by $\bar{\tau}_s = \max(1 - s, \varepsilon)$ gives Equation 5 and avoids numerical amplification near the terminal time.

A.2 Velocity–Endpoint Identity and Clipped Target

For completeness, let $\Delta = s - t$, $\beta = 1 - \rho$, and $\tau = 1 - s$. Substituting $v_\theta(x_t, t, c) = u_t + e_t$ into the controlled rollout gives

$$\widehat{x}_s = x_t + \Delta\left[\rho u_t + \beta(u_t + e_t)\right] \tag{19}$$

$$= x_t + \Delta u_t + \Delta\beta e_t \tag{20}$$

$$= x_s + \Delta\beta e_t. \tag{21}$$

This proves Equation 13. Since $x_1 - x_s = \tau u_t$, the unclipped corrective target is

$$u_s^{\mathrm{corr}} = \frac{x_1 - \widehat{x}_s}{\tau} \tag{22}$$

$$= \frac{x_1 - x_s - \Delta\beta e_t}{\tau} \tag{23}$$

$$= u_t - \frac{\Delta\beta}{\tau} e_t, \tag{24}$$

and Equation 12 follows directly from

$$v_\theta(\widehat{x}_s, s, c) - u_s^{\mathrm{corr}} = \frac{F_\theta(\widehat{x}_s, s, c) - x_1}{\tau}. \tag{25}$$

When the implementation uses $\bar{\tau} = \max(\tau, \varepsilon)$, the target instead becomes

$$u_s^{\mathrm{corr}} = \frac{\tau}{\bar{\tau}} u_t - \frac{\Delta\beta}{\bar{\tau}} e_t, \tag{26}$$

so the velocity target remains well-defined near $s = 1$, but the exact velocity–endpoint equivalence uses $\bar{\tau}$ rather than the physical remaining time τ.

B Experimental Protocols

B.1 Class-to-Image Experimental Details

We summarize the common C2I training settings and the hyperparameters specific to MiTraFlow in Table 7. Settings changed only for a dedicated ablation are marked explicitly.

Table 7: Hyperparameters for the class-to-image experiments.

Parameter	Setting
Optimization	
Optimizer	AdamW
Base learning rate	1×10^{-4}
AdamW betas	$(0.9, 0.95)$
Weight decay	0
Global batch size	256
Learning-rate schedule	Constant
Flow matching	
Probability path	$x_t = (1-t)x_0 + tx_1$
Time sampler	$\mathrm{LogitNormal}(0,1)$, shift $= 1$
Flow-matching loss	$\lambda_t = 1.0$
REPA supervision	
External feature encoder	DINOv2-B/14
REPA source layer	$\ell_{\mathrm{ext}} = 8$
REPA loss	$\lambda^t_{\mathrm{ext}} = 0.5,\ \lambda^s_{\mathrm{ext}} = 0$
MiTraFlow	
Time-pair construction	$t = \min(t_a, t_b),\ s = \max(t_a, t_b)$
Rollout integrator	$\widehat{x}_s = x_t + (s-t)\widetilde{u}_t$
Rollout mixture	$\rho = 0.9$
Corrective-target clamp	$\varepsilon = 0.05$
Corrective loss	$\lambda_s = 1.0$
Shallow–deep layers	$\ell_{\mathrm{sh}} = 8,\ \ell_{\mathrm{dp}} = 20$
Shallow–deep projector	$\mathrm{MLP}_3 + \mathrm{SiLU}$
Shallow–deep loss	$\lambda_{\mathrm{align}} = 0.8$
Sampling	
Diffusion sampler	Euler–ODE
Diffusion steps	250
Evaluation suite	ADM

B.2 Text-to-Image Data

T2I pretraining uses 42.7M image–text pairs from BLIP3o (Chen et al., 2025), DenseFusion (Li et al., 2024b), and two FLUX-generated collections, sampled in proportion to their corpus sizes (68.76%, 2.48%, 14.11%, and 14.64%, respectively). During finetuning, we sample Echo-4o (Ye et al., 2025) with probability 0.9 and use the remaining 0.1 for replay from the pretraining sources and JourneyDB (Sun et al., 2023). Images are converted to RGB, resized and center-cropped to 256×256, and normalized to $[-1, 1]$; captions are truncated or padded to 128 tokens. All quantitative T2I results in Table 6 are obtained at 256×256. For the 512×512 visualizations in Figures 2 and 5, the finetuned model is further finetuned at 512×512 for 10K iterations.

C Additional Quantitative Results

C.1 Homotopy Between Single-State and Dual-State Training

The dual-state objective of MiTraFlow differs from single-state flow matching in two coupled ways: it draws an ordered time pair instead of a single time, and it adds the corrective loss on

the rollout state $\widehat{x}_s$. Both differences can be removed continuously, which lets us view single-state training as a degenerate case of the dual-state construction rather than a separate objective. We make this explicit with a homotopy that deforms one objective into the other, and use it to probe whether the corrective objective can be reached without warm-starting from a single-state checkpoint.

Construction. Let k denote the number of iterations elapsed since training begins, let K be the homotopy length, and define the progress variable

$$\gamma(k) = \tfrac{1}{2}[1 - \cos(\pi \min(k/K,\, 1))] \in [0, 1]. \tag{27}$$

For each training pair we sample two times t_a, t_b independently from the time sampler, and instead of reordering them directly we interpolate from the shared time t_a towards the ordered pair,

$$t(\gamma) = t_a + \gamma\,(\min(t_a, t_b) - t_a)\,, \qquad s(\gamma) = t_a + \gamma\,(\max(t_a, t_b) - t_a)\,. \tag{28}$$

The same progress variable scales the corrective term, so the training objective becomes

$$\mathcal{L}(\gamma) = \lambda_t \mathcal{L}_{\mathrm{FM}} + \gamma\, \lambda_s \mathcal{L}_{\mathrm{corr}} + \lambda_{\mathrm{ext}}^{t} \mathcal{L}_{\mathrm{ext}}, \tag{29}$$

with all other hyperparameters, including the rollout mixture $\rho = 0.9$, fixed at their defaults.

The two endpoints are exact. At $\gamma = 0$ we have $t = s = t_a$, so the rollout displacement $(s - t)\widetilde{u}_t$ vanishes and $\widehat{x}_s = x_t$ for any ρ; the corrective term simultaneously receives zero weight. What remains is flow matching plus representation alignment evaluated at a single state, which is precisely the REPA objective that serves as our single-state baseline. At $\gamma = 1$ the time pair is fully ordered and the corrective term carries its full weight, recovering MITRAFLOW. Every intermediate value of γ defines a valid objective in which the two states separate gradually and the rollout distance grows in step with the corrective supervision. Scaling only the loss weight would lack this property, since the marginal distribution of t would jump discontinuously once the ordered pair is introduced.

Results. Table 8 reports this study at 300K cumulative iterations. The three REPA-initialized rows start from the REPA checkpoint at 100K iterations and train the corrective objective for a further 200K iterations, while the randomly initialized row trains it for all 300K; all rows use the rollout objective without shallow–deep alignment, matching the setting of Table 2.

Table 8: Homotopy between single-state and dual-state training on ImageNet 256×256 without classifier-free guidance. All rows are evaluated at 300K cumulative iterations and train the corrective objective without shallow–deep alignment. *Schedule* gives the homotopy length K of Equation 27, where $K = 0$ means the corrective objective is active from the first iteration. *Corr. iters* counts the iterations trained with the corrective objective, which is smaller than the cumulative count for the REPA-initialized rows. The shaded row is the default configuration used elsewhere in the paper. Bold denotes the best result per column.

Initialization	Schedule	Corr. iters	FID ↓	sFID ↓	FDr^6 ↓	IS ↑	Prec. ↑	Rec. ↑
REPA 100K	Direct switch ($K = 0$)	200K	7.68	**4.79**	16.66	126.8	0.69	0.75
Random	Direct switch ($K = 0$)	300K	8.98	4.82	18.73	116.2	0.68	0.75
REPA 100K	Homotopy ($K = 25$K)	200K	7.69	**4.79**	**16.63**	**127.3**	0.69	0.75
REPA 100K	Homotopy ($K = 50$K)	200K	**7.62**	4.82	16.64	126.9	**0.70**	**0.79**

Training from random initialization is clearly worse, reaching FID 8.98 against 7.68 for the warm-started default, with IS falling from 126.8 to 116.2 and FDr^6 rising from 16.66 to 18.73. The gap is concentrated in fidelity rather than coverage: sFID and recall are almost unchanged. As the *Corr. iters* column shows, the randomly initialized run reaches this deficit after 50% more corrective updates than the warm-started ones, so the comparison understates it. Learning to correct model-induced error therefore appears to require a model whose predictions are already informative enough for the rollout state to be meaningful.

Annealing the objective recovers part of this behavior but yields only a small benefit over direct switching. With $K = 50$K the homotopy reaches FID 7.62 against 7.68, with precision improving from 0.69 to 0.70 and recall from 0.75 to 0.79, the largest single difference in the table. With $K = 25$K every metric agrees with the direct-switch default to within 0.01, indicating that a homotopy

short relative to the total training length reaches essentially the same model. Because the longer schedule's advantage is modest and was measured late in the project, we do not use the homotopy in our main results and report it as a property of the objective rather than a recommended recipe.

All four rows can in fact be placed on a single schedule. Training the REPA objective is exactly training at $\gamma = 0$, so initializing from the REPA checkpoint amounts to holding $\gamma = 0$ for 100K iterations; the direct-switch rows then jump to $\gamma = 1$, whereas the homotopy rows ramp over K iterations. Read this way, the randomly initialized row is the only configuration in Table 8 that never trains at $\gamma = 0$, and it is also the only one that degrades substantially.

Taken together, these observations support reading single-state training as a collapsed case of dual-state training, obtained by driving the time gap and the corrective weight to zero. Under this reading the warm-start phase is not a separate pretraining stage but the $\gamma = 0$ segment of one schedule, and the homotopy replaces the discontinuous end of that segment with a continuous path. A run could therefore begin at $\gamma = 0$ as an ordinary single-state model and arrive at the full corrective objective without any discontinuous change of target. Establishing whether such a schedule matches warm-starting when run to convergence from random initialization remains open.

D Qualitative Results

D.1 Text-to-Image Visualization

Figure 5 shows additional unselected text-to-image samples at 512×512 resolution.

Prompts for the main-text visualization. The prompts for Figure 2, in row-major order, are:

1. A lone motorcycle courier crossing a rain-soaked neon plaza at midnight, crisp cyan reflections and a clear cinematic silhouette.
2. A compact lunar rover parked beneath an enormous ringed planet, silver dust and deep indigo shadows.
3. A terraced solarpunk library wrapped in flowering vines above a turquoise canal, bright morning light.
4. Macro photograph of an iridescent jewel beetle on wet emerald moss, razor-sharp shell texture.
5. A living fox formed from smoky quartz and clear crystal running across fresh snow, coherent anatomy, sparkling fragments trailing behind, low winter sunlight.
6. A modern concrete house cantilevered over a tropical waterfall, warm interior light, mist rising through dense green foliage, photographed from across the valley.
7. A weathered temple bell hangs beneath the skeleton of an ancient whale on a windswept coastal cliff; a single monk approaches along stone steps, sea birds cross the bright horizon, and soft dawn light unifies bone, bronze, grass, and ocean textures.
8. A circular astronomical observatory built into rust-colored desert cliffs, open silver dome, astronomers under a dense star field, moonlit architectural panorama.
9. A high-fashion sneaker sculpted from white porcelain with cobalt botanical inlays, suspended above pale stone, soft directional light revealing every surface detail.
10. A colossal biomechanical whale gliding above a flooded futuristic city, copper ribs and blue bioluminescence, tiny boats below for scale, clear sunrise atmosphere.

D.2 Class-to-Image Visualization

Figures 6 and 7 show ImageNet class-conditional samples at 256×256 resolution using EMA weights, 250-step Euler–ODE sampling, CFG 1.8, and guidance interval $(0.3, 1.0)$. The displayed samples are drawn randomly and shown without further manual filtering.

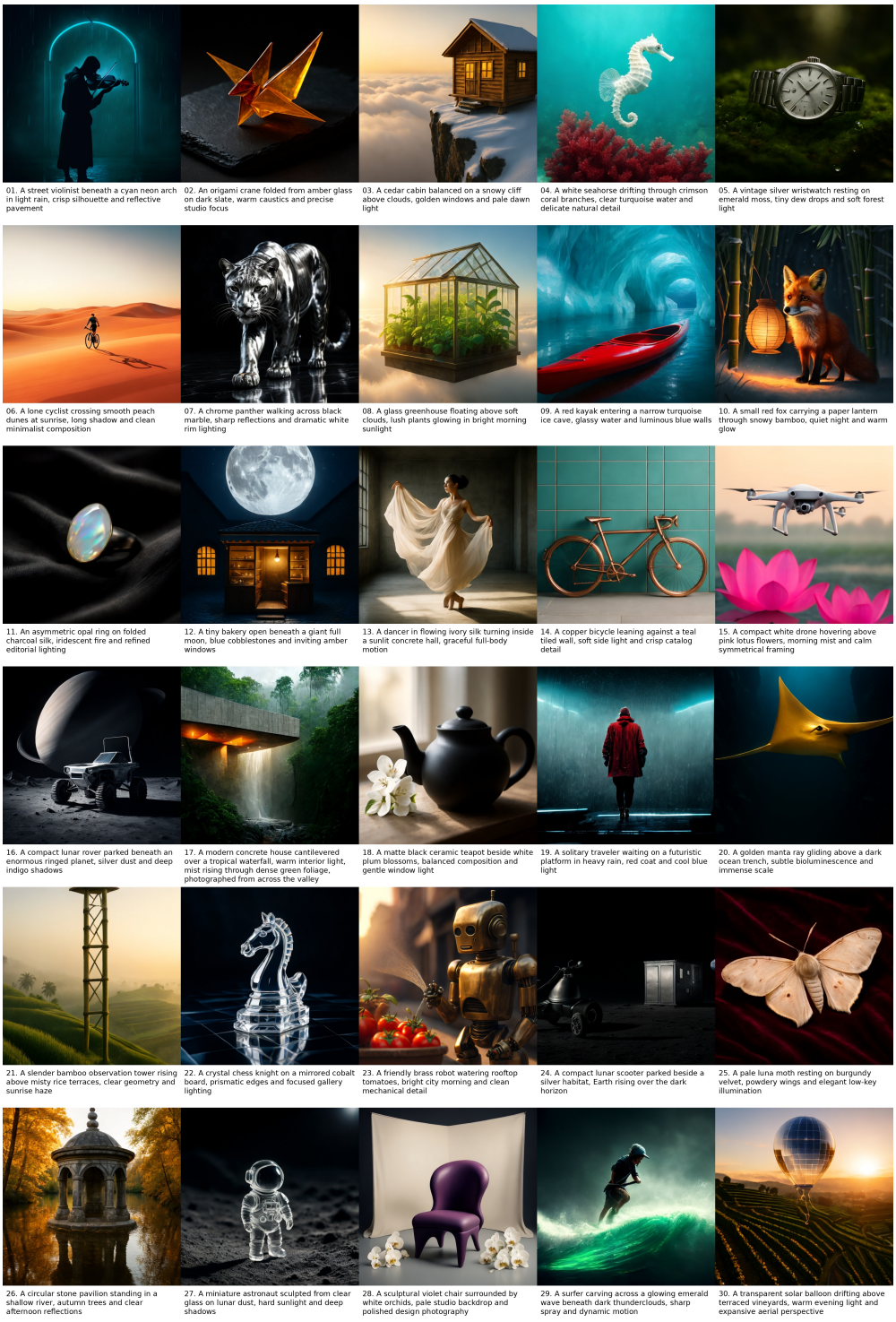


Figure 5: Unselected text-to-image 512×512 visualization.

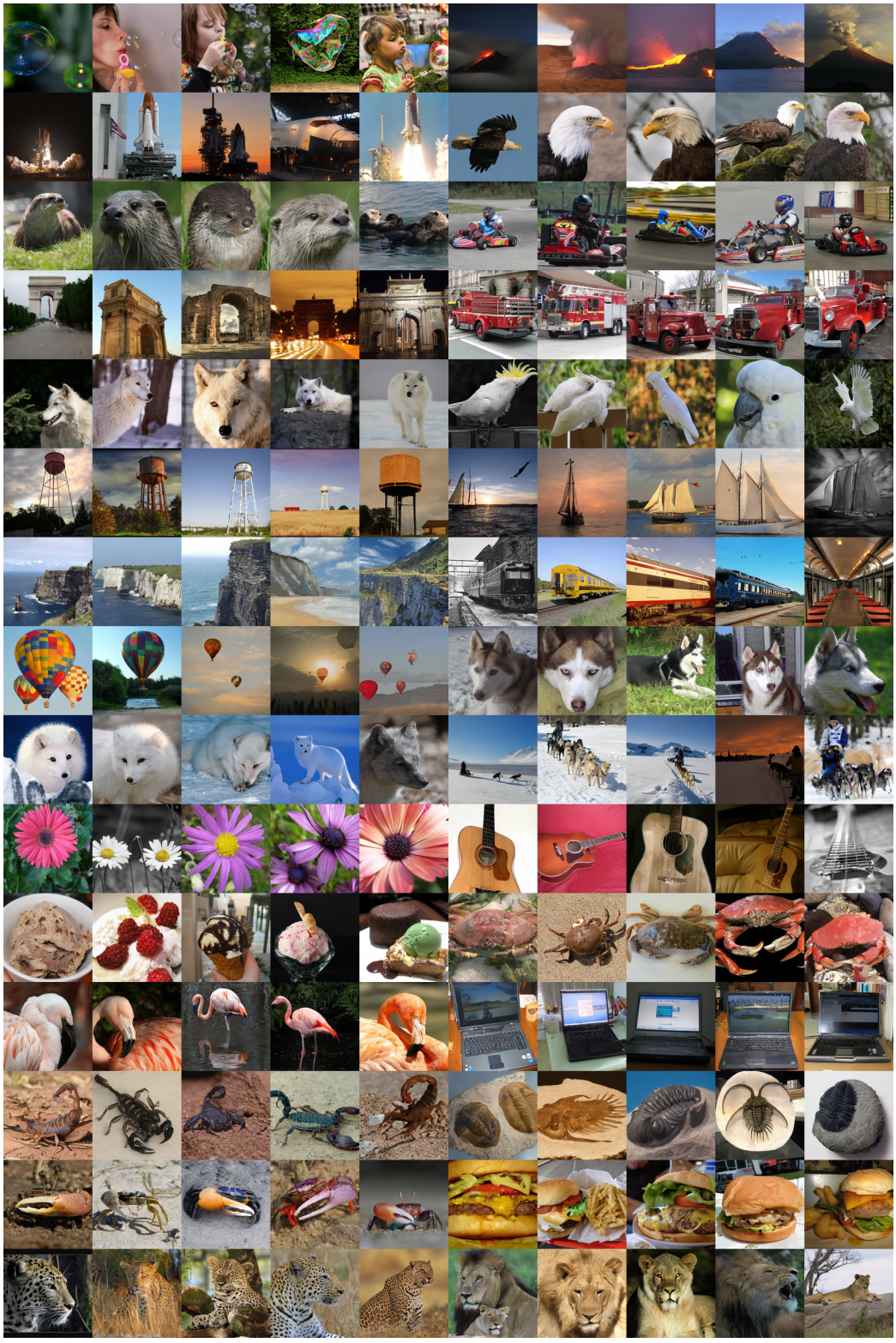


Figure 6: Randomly drawn class-to-image samples generated by MITRAFLOW on ImageNet 256 × 256 (part 1 of 2).

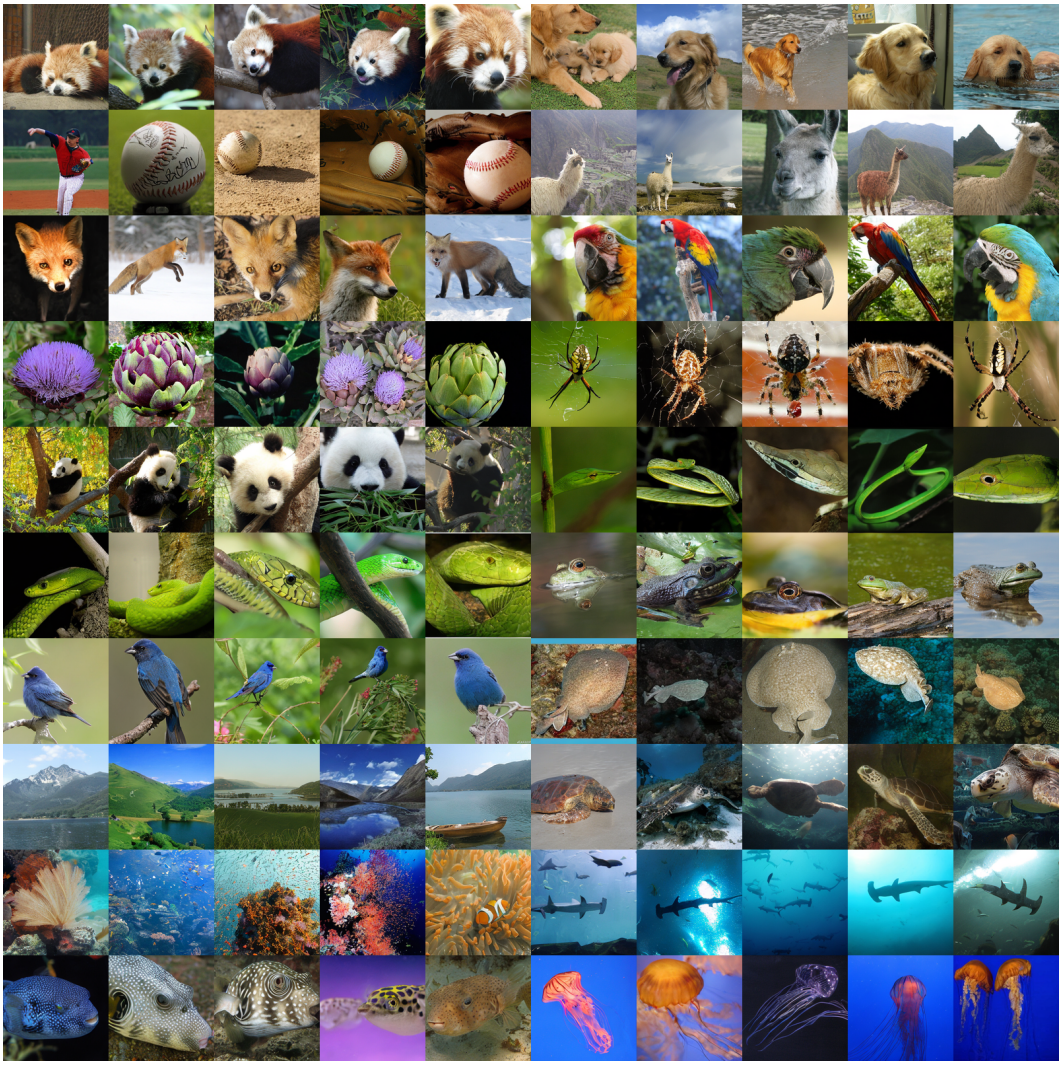


Figure 7: Randomly drawn class-to-image samples generated by MITRAFLOW on ImageNet 256 × 256 (part 2 of 2).