The Curse of Conditions: Analyzing and Improving Optimal Transport for Conditional Flow-Based Generation

Supplementary Material

Table of Contents

A. Additional Experiments

A.1. Path Straightness

We measure path straightness using Eq. (18) from [13], with 10K samples (conditions sampled from the validation sets) and 25 steps in Tab. A1. Specifically, we measure

$$\mathbb{E}_{t,q(x_0)}\left[\|v_{\theta,c}(t,\psi_t(x_0))\|^2 - \|\psi_1(x_0) - x_0\|^2\right], \tag{A1}$$

where $\psi_t(x_0)$ denotes the numerical integration result when integrating x_0 from 0 to t following the flow field represented by $v_{\theta,c}$. Our method consistently achieves straighter paths than FM. Although OT has even straighter paths, the end points of these paths do not accurately model the target distribution, indicated by the higher FID.

	CIFAR		ImageNet32		ImageNet256	
	Straightness ↓	FID ↓	Straightness ↓	FID ↓	Straightness ↓	FID ↓
FM	110.83±0.20	**5.64**	133.95±0.34	7.17	4803.0±79.8	3.90
OT	**73.54**±0.24	6.82	**97.71**±0.23	7.85	**4088.6**±37.5	7.48
C^{2}OT (ours)	84.99±0.26	**5.64**	122.54±0.34	**7.07**	4625.2±54.5	**3.70**

Table A1. Path straightness and FID of FM, OT, and C^{2}OT.

A.2. Class-to-Image

We supplement our existing CIFAR-10 class-to-image experiments (**??**) with additional class-to-image results on ImageNet32. Results in Tab. A2 are consistent with existing findings.

ImageNet 32×32 Class-Conditioned Generation							
Method	Euler-2	Euler-5	Euler-10	Euler-25	Euler-50	Euler-100	Adaptive
FM	116.296	22.224	9.530	5.892	5.334	5.116	**4.993**
OT	**71.700**	20.703	11.385	8.065	7.492	7.303	7.316
C^{2}OT (ours)	81.480	**18.285**	**8.661**	**5.607**	**5.201**	**5.055**	5.035

Table A2. Class-to-image performance comparisons on ImageNet-32.

A.3. Predicting Conditions from Coupled Prior

We note that OT degrades less significantly in high-dim as it skews the prior less (but still does), since mini-batch OT becomes noisy in high-dim. To quantify, we train a condition-classifier with the coupled prior (x_0) as input (*e.g.*, in Fig. 2 of the main paper, a classifier can perfectly predict the condition based on x_0) on CIFAR with varying resolutions and number of dims. We collect 100K couplings for training, and divide them into an 80:20 train/test split. Tab. A3 shows the results: as the number of dimensions increases, the classifier becomes less accurate (less prior skew) for OT. Both FM and C^{2}OT lead to unbiased prior, so the classifier performs at random guess accuracy (10%).

Method	2 × 2	4 × 4	8 × 8	16 × 16	32 × 32
FM	10.0%±0.3	9.9%±0.1	10.0%±0.3	9.9%±0.2	9.8%±0.1
OT	29.7%±0.4	27.3%±0.3	24.4%±0.4	22.6%±0.3	20.1%±0.3
C^{2}OT (ours)	10.0%±0.3	10.0%±0.3	10.0%±0.3	9.9%±0.2	9.9%±0.2

Table A3. Test-set classification accuracy of predicting the conditioning class when given the coupled input noise in the CIFAR-10 dataset.

A.4. Extended Results with Different OT Batch Sizes

We compare OT batch size scaling of OT and C^{2}OT in Table A4. The results are consistent with our findings in **??** – OT has better FID with few steps but does not align well with the input condition (worse CE), and increasing OT batch size improves few-step performance and slightly harms many-step performance.

	Method (OT batch size)	Euler-2	Euler-5	Euler-10	Euler-25	Euler-50	Euler-100	Adaptive	NFE ↓
FID ↓	OT (128)	64.305±0.593	18.033±0.566	10.659±0.354	6.753±0.26	5.326±0.157	4.639±0.084	4.142±0.015	125.6±0.5
	OT (640)	59.506±0.976	18.710±0.673	11.588±0.475	7.555±0.301	5.997±0.222	5.232±0.175	4.624±0.122	126.1±1.1
	OT (1280)	57.656±1.075	18.206±2.063	11.284±1.707	7.452±1.116	6.036±0.696	5.374±0.408	4.895±0.044	125.1±6.5
	OT (2560)	56.186±0.876	17.976±1.752	11.084±1.682	7.381±1.078	6.068±0.633	5.473±0.325	5.071±0.074	125.8±7.1
	OT (5120)	54.966±0.212	19.399±0.194	12.563±0.112	8.498±0.056	6.869±0.038	6.042±0.042	5.352±0.074	128.0±3.0
	C^2OT (128)	74.517±0.075	18.281±0.443	9.704±0.326	5.536±0.193	4.077±0.135	3.391±0.101	2.880±0.059	124.0±2.5
	C^2OT (640)	64.849±0.474	17.572±0.263	9.770±0.195	5.634±0.112	4.150±0.083	3.446±0.068	2.905±0.047	127.2±0.3
	C^2OT (1280)	61.850±0.373	17.325±0.407	9.706±0.321	5.668±0.206	4.200±0.147	3.495±0.102	2.951±0.048	124.1±0.9
	C^2OT (2560)	59.361±0.324	17.213±0.297	9.795±0.200	5.712±0.149	4.213±0.102	3.496±0.068	2.942±0.030	125.5±0.7
	C^2OT (5120)	56.011±0.556	17.046±0.185	9.904±0.205	5.830±0.126	4.291±0.079	3.536±0.054	2.945±0.022	128.7±0.7
CE ↓	OT (128)	2.525±0.015	0.619±0.016	0.425±0.006	0.368±0.004	0.361±0.004	0.363±0.006	0.371±0.006	125.6±0.5
	OT (640)	2.386±0.048	0.672±0.018	0.470±0.011	0.405±0.008	0.393±0.006	0.392±0.007	0.398±0.008	126.1±1.1
	OT (1280)	2.348±0.057	0.690±0.030	0.487±0.015	0.427±0.009	0.422±0.005	0.425±0.003	0.435±0.007	125.1±6.5
	OT (2560)	2.298±0.031	0.702±0.020	0.505±0.010	0.443±0.006	0.436±0.008	0.438±0.011	0.447±0.014	125.8±7.1
	OT (5120)	2.277±0.029	0.742±0.006	0.537±0.012	0.464±0.009	0.453±0.008	0.453±0.007	0.461±0.006	128.0±3.0
	C^2OT (128)	2.636±0.033	0.485±0.007	0.317±0.004	0.270±0.004	0.265±0.001	0.266±0.002	0.271±0.004	124.0±2.5
	C^2OT (640)	2.270±0.021	0.477±0.012	0.321±0.005	0.276±0.008	0.272±0.004	0.273±0.005	0.280±0.005	127.2±0.3
	C^2OT (1280)	2.166±0.032	0.475±0.010	0.324±0.008	0.279±0.003	0.273±0.004	0.274±0.003	0.281±0.003	124.1±0.9
	C^2OT (2560)	2.049±0.005	0.468±0.005	0.321±0.003	0.278±0.006	0.272±0.007	0.272±0.007	0.277±0.008	125.5±0.7
	C^2OT (5120)	1.910±0.037	0.463±0.006	0.327±0.008	0.282±0.005	0.278±0.005	0.279±0.004	0.284±0.004	128.7±0.7

Table A4. Performance of OT and C^2OT in CIFAR-10 when trained with different OT batch sizes.

A.5. Extended Results with Different Target Ratios

We list additional results on ImageNet-32 and ImageNet-256 in Table A5: $r_{\text{tar}} = 0.01$ generally strikes a good balance, but we note that it might not be optimal for all datasets.

ImageNet 32×32 Caption-Conditioned Generation							
r_{tar}	Euler-2	Euler-5	Euler-10	Euler-25	Euler-50	Euler-100	Adaptive
0.005	104.386	22.254	10.939	**7.015**	**6.022**	**5.576**	**5.284**
0.01	102.380	21.965	**10.897**	7.069	6.084	5.638	5.350
0.1	**82.456**	**20.820**	11.035	7.438	6.501	6.080	5.843
ImageNet 256×256 Caption-Conditioned Generation							
0.005	201.906	**30.159**	**9.852**	5.114	3.795	3.373	3.377
0.01	201.010	30.578	10.032	**5.075**	**3.702**	**3.335**	**3.290**
0.1	**199.009**	37.014	13.293	6.277	4.608	4.268	4.570

Table A5. Results with different r_{tar}.

A.6. Reference Condition Adherence Metrics

We measure condition adherence in Section 4 via two condition adherence metrics. On CIFAR-10, we compute the average cross-entropy of the logits predicted by a pretrained classifier[1] on the generated images against the ground-truth conditioning labels. On ImageNet, we compute the average cosine distance between the CLIP embeddings extracted from the generated images versus the conditioning captions, using SigLip-2 [15][2]. For reference, we compute these metrics on the validation set using ground-truth images and present the results in Table A6.

B. Extended Plots

Figures A1 and A2 extend Figures 5 and 6 of the main paper to include all data points.

[1] `https://github.com/chenyaofo/pytorch-cifar-models`, commit d1c8e99b911da7d412979600c84d2a4fe3728473, ResNet56

[2] ViT-SO400M-16-SigLIP2-256

Dataset	CE (↓)
CIFAR-10	0.0005
Dataset	**CLIP (↑)**
ImageNet 32×32	0.1119
ImageNet 256×256	0.1363

Table A6. Reference condition adherence metrics on ground-truth images.

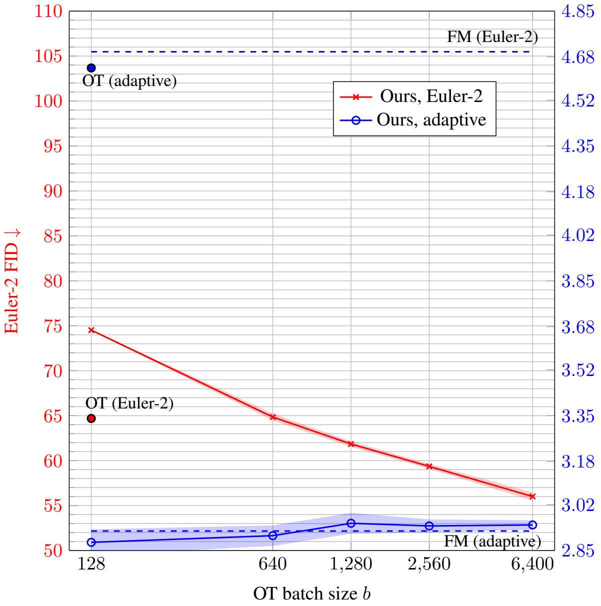

Figure A1. Changes in FID with respect to varying OT batch sizes b. We plot mean±std over three runs and represent std with a shaded region.

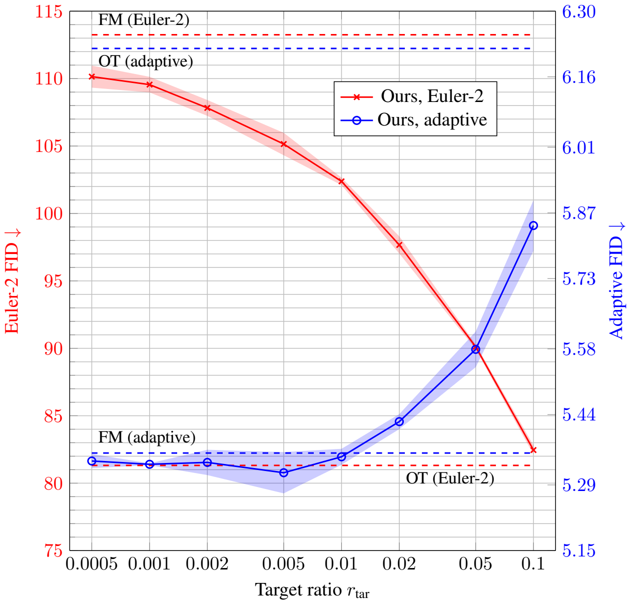

Figure A2. Changes in FID with respect to varying target ratio r_{tar}. We plot mean±std over three runs and represent std with a shaded region.

C. Data Coupling in 8 Gaussians→moons

Figure A3 extends Figure 1 with an additional row that shows coupling during training. Clearly, OT samples form a biased distribution at training time in conditional generation as discussed. Since we cannot sample from this biased distribution at test-time, we obtain a gap between training and testing. This gap degrades the performance of OT.

D. Implementation Details

D.1. Two-Dimensional Data

Data. Following the implementation of Tong et al. [14], we generate the "moons" data using the `torchdyn` library [12], and the "8 Gaussians" using `torchcfm` [14].

Network. We employ a simple multi-layer perceptron (MLP) network for this dataset. Initially, the two-dimensional input (x and y coordinates) and the flow timestep (a scalar uniformly sampled from $[0, 1]$) are projected into the hidden dimension using individual linear layers. When an input condition is provided, it is similarly projected into the hidden dimension. Discrete conditions are encoded as -1 or +1, while continuous conditions are represented by the x-coordinate of the target data point. After projection, all input features are summed and processed through a network comprising three MLP modules. Each MLP module consists of two linear layers, where the first layer uses an expansion ratio of 4, and is followed by a GELU activation function [5]. We use a residual connection to incorporate the output of each MLP block. Finally, another linear layer projects the features to two dimensions to produce the output velocity. The hidden dimension is set to 128.

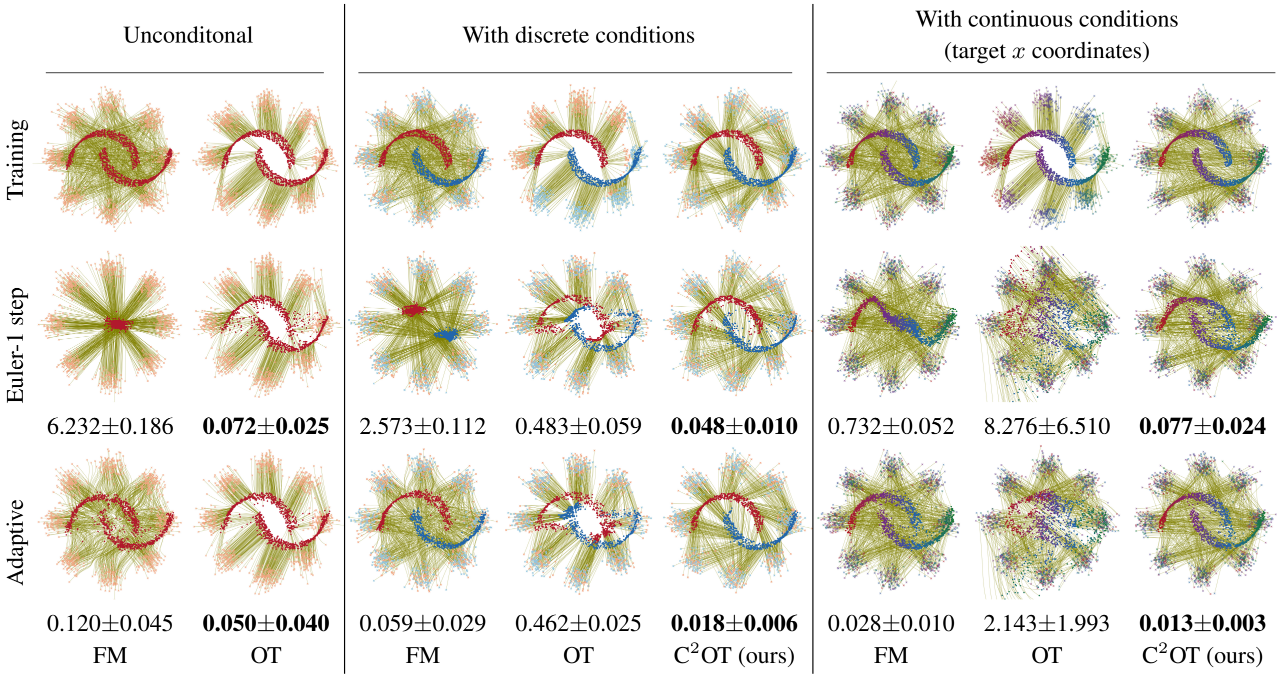

Figure A3. We visualize the flows learned by different algorithms using the `8gaussians`→`moons` dataset. Below each plot, we show the 2-Wasserstein distance (lower is better; mean±std over 10 runs). Compared to Figure 1, we have added a first row illustrating the prior-data coupling during training. Note that the OT coupled paths during training (first row) do cross. This is expected – the commonly referred to "no-crossing" property of OT coupling refers to the uniqueness of the pair (x_0, x_1) given x and t – at the same timestep t, no two paths may cross at x (see Proposition 3.4 in Tong et al. [14] and Theorem D.2 in Pooladian et al. [13]). Since we plot all timesteps simultaneously in this figure, there are apparent crossings. However, the intersecting paths do not share the same timestep t at the point of intersection.

	CIFAR-10	ImageNet-32
Channels	128	256
Depth	2	3
Channels multiple	1, 2, 2, 2	1, 2, 2, 2
Heads	4	4
Heads channels	64	64
Attention resolution	16	4
Dropout	0.0	0.0
Use scale shift norm	True	True
Batch size / GPU	128	128
GPUs	2	4
Effective batch size	256	512
Iterations	100k	300k
Learning rate	2.0e−4	1.0e−4
Learning rate scheduler	Warmup then constant	Warmup then linear decay
Warmup steps	5k	20k
OT batch size (per GPU, for C^{2}OT)	640	6400

Table A7. Hyperparameter settings for training on CIFAR-10 and ImageNet-32.

Training. We train each network for 20,000 iterations with the Adam [7] optimizer, a learning rate of 3e-4 without weight decay, and a "deep net" batch size of 256 for computing forward/backward passes/gradient updates. We use an OT batch size b of 1024 and a target ratio r_{tar} of 0.01.

D.2. CIFAR-10

In CIFAR-10, we employ the UNet architecture used by Tong et al. [14]. We list our hyperparameters in Table A7, following the format in [9, 13, 14]. To accelerate training, we use `bf16`. We use the Adam [7] optimizer with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay=0.0, and $\epsilon = 1\text{e}{-8}$. For learning rate scheduling, we linearly increase the learning rate from 1.0e$-$8 to 2.0e$-$4 over 5,000 iterations and then keep the learning rate constant. The "use scale shift norm" denotes employing adaptive layer normalization to incorporate the input condition, as implemented in [3]. To stabilize training, we clip the gradient norm to 1.0. We report the results using an exponential moving average (EMA) model with a decay factor of 0.9999. For FID computation, we use the `clean_fid` library [11] in `legacy_tensorflow` mode following [14].

D.3. ImageNet-32

Data. We use face-blurred ImageNet-1K [2, 16] following [10], and apply the downsampling script from [1]. Images are downsampled to 32××32 using the 'box' algorithm, and reference FID statistics are computed with respect to the downsampled validation set images. For text input, we use the captions provided by [8].

Network and Training. We largely follow the training pipeline described in Appendix D.2. We use a larger network following [13] and list our hyperparameters in Table A7. To encode text input, we use the `openclip` library [6] and the text encoder of the "DFN5B-CLIP-ViT-H-14" checkpoint [4], a pretrained CLIP-like model. CLIP feature vectors are normalized to unit norm before being used as input conditions. For learning rate scheduling, after the initial warmup phase, we linearly decay the learning rate to 1.0e$-$8 over time.

Evaluation. As stated in the main paper, we use 49,997 images from the validation set to compute FID. This is because the fine-grained nature of image captions might lead to overfitting, *i.e.*, memorizing the training set. For CLIP score computation, we evaluate the cosine similarity between the input caption and the generated image using SigLIP-2 [15], with the `ViT-SO400M-16-SigLIP2-256` checkpoint via the `openclip` library [6].

D.4. ImageNet-256

For this dataset, we use the open-source implementation of LightningDiT [17] and train the models under the '64 epochs' setting with minimal modifications to change the network from class-conditioned to caption-conditioned. In addition to integrating the coupling algorithms (OT and C^{2}OT, while the original LightningDiT [17] already employs FM), our modifications include:

1. Changing the input conditional mapping layer from an embedding layer (that takes a class label as input) to a linear layer (that takes CLIP features as input).
2. Adjusting the classifier-free guidance (CFG) scale. We find that the model benefits from a higher CFG scale when using caption conditioning. Specifically, we increase the CFG scale from 10.0 to 17.0, and adjust the CFG interval start parameter from 0.11 to 0.10.

   For data and evaluation, we follow the same setup as described in Appendix D.3.

E. Additional Generated Images

We present additional image generation results in this section. All showcased images are uncurated, meaning they were sampled completely at random. For consistency and direct comparison, we used the same random seed for each generation across different methods.

E.1. CIFAR-10

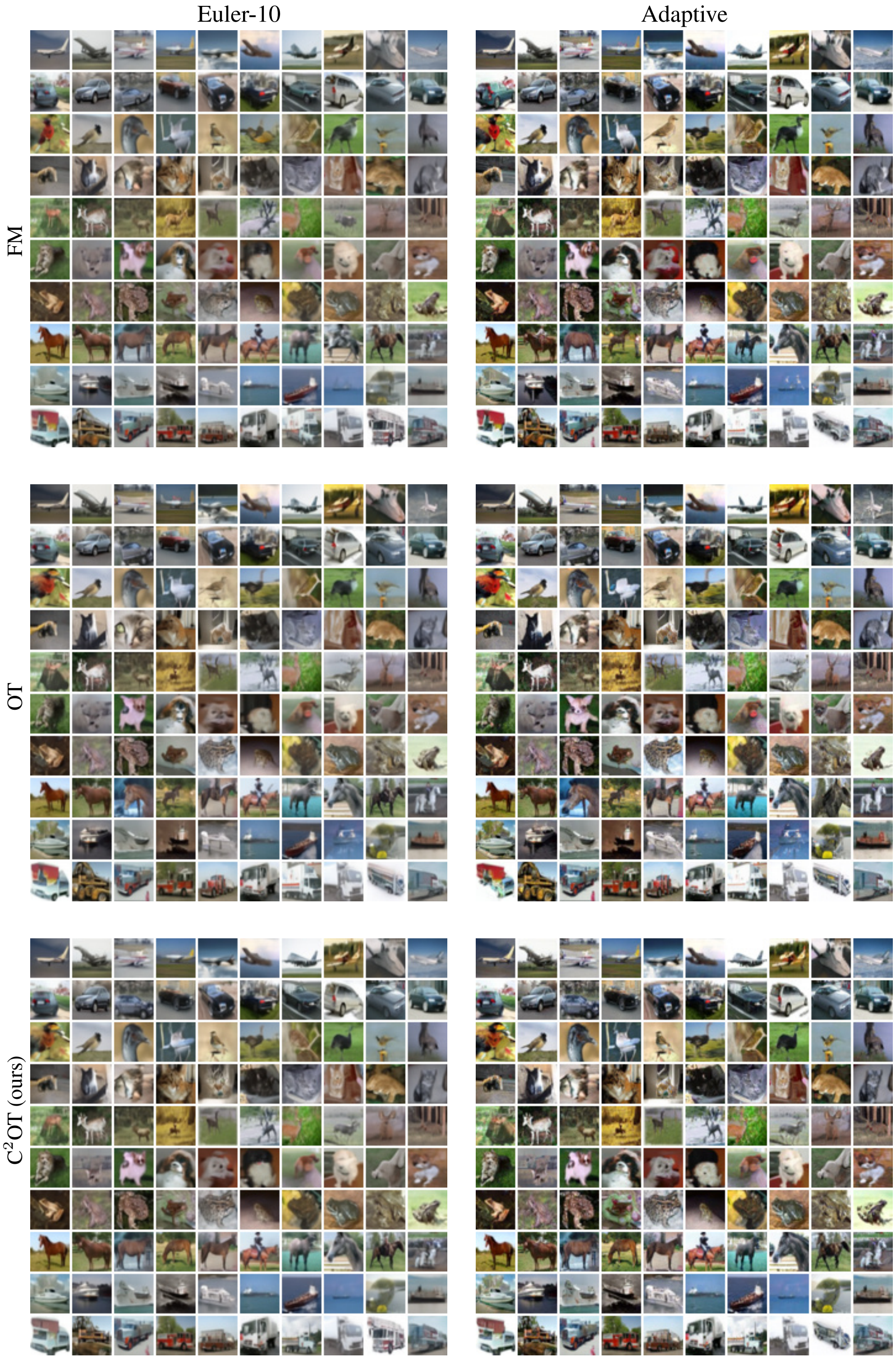

Figure A4. *Uncurated* generations in CIFAR-10, 10-per-class. We compare FM, OT, and C^{2}OT with both 10-step Euler's method and an adaptive solver for test-time numerical integration.

E.2. ImageNet-32

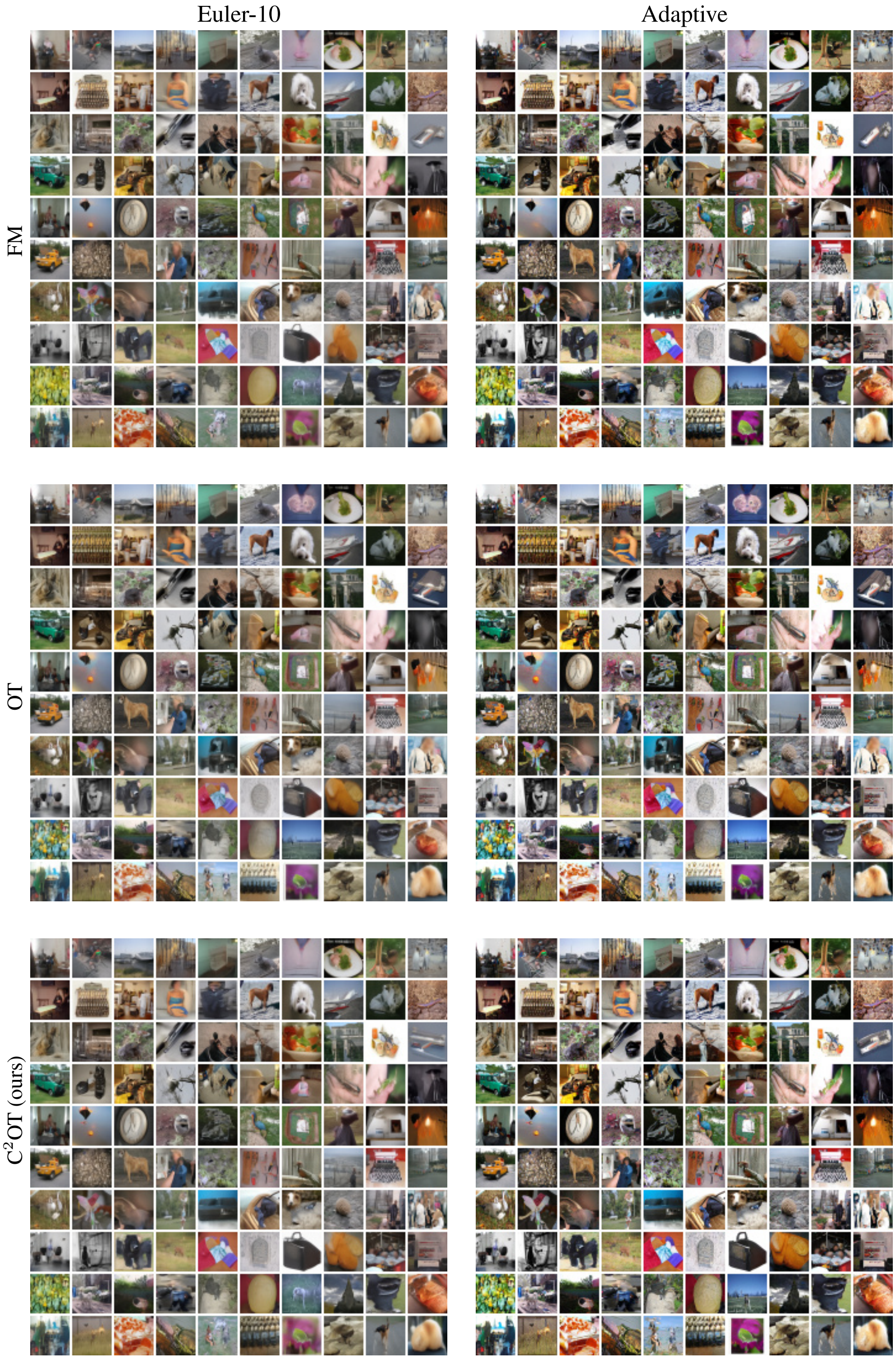

Figure A5. *Uncurated* generations in ImageNet-32. We compare FM, OT, and C^2OT with both 10-step Euler's method and an adaptive solver for test-time numerical integration.

E.3. ImageNet-256

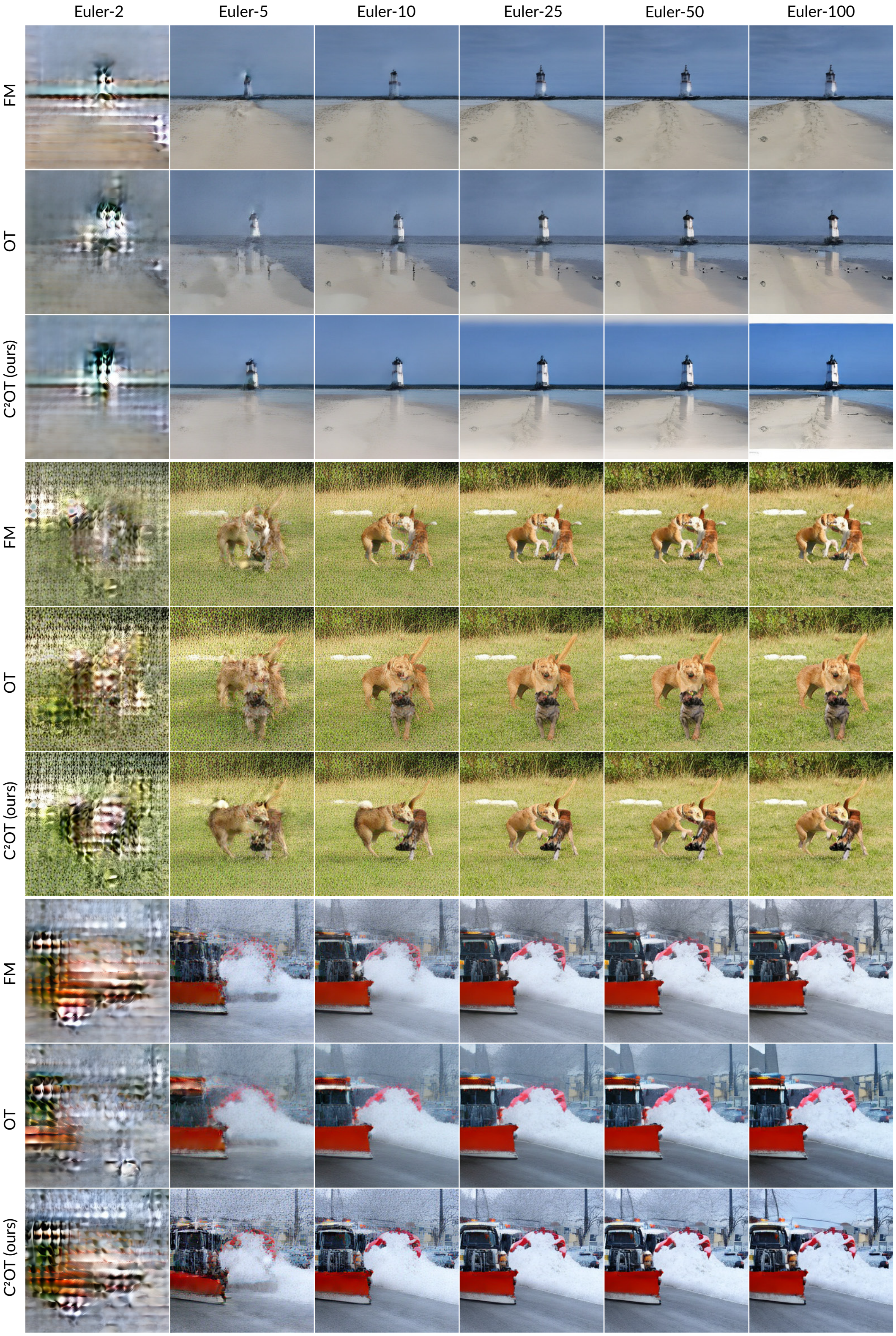

Figure A6. *Uncurated* generations in ImageNet-256. We compare FM, OT, and C^{2}OT with different amounts of sampling steps.

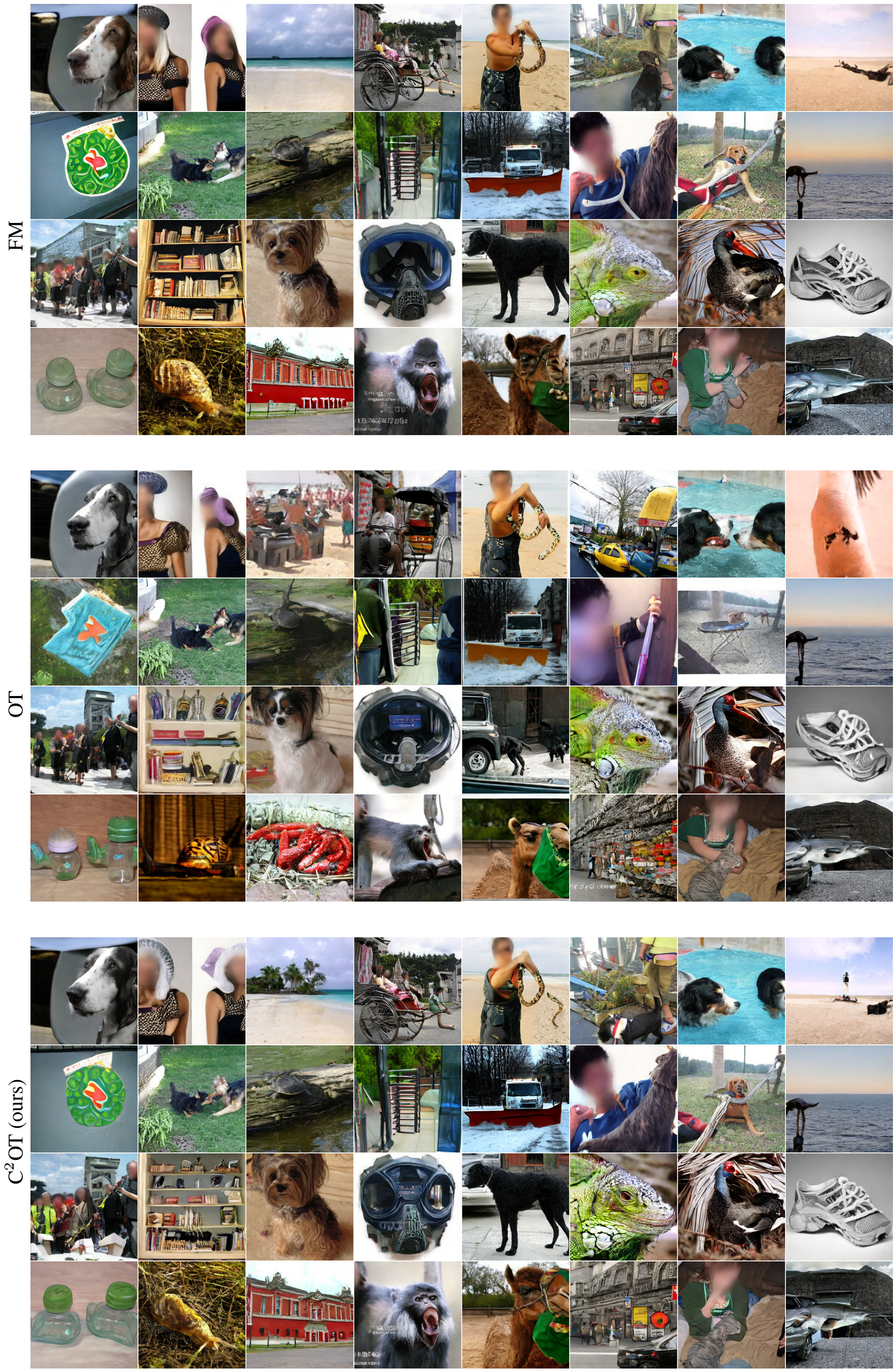

Figure A7. *Uncurated* generations in ImageNet-256. We compare FM, OT, and C^{2}OT with an adaptive solver for test-time numerical integration.

F. DeltaAI Acknowledgment

This work used the DeltaAI system at the National Center for Supercomputing Applications through allocation CIS250008 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants 2138259, 2138286, 2138307, 2137603, and 2138296.

References

[1] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv*, 2017. 6

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 6

[4] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *ICLR*, 2024. 6

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv*, 2016. 4

[6] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 6

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 5, 6

[8] Visual Layer. Imagenet-1k-vl-enriched. `https://huggingface.co/datasets/visual-layer/imagenet-1k-vl-enriched`, 2024. 6

[9] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2023. 6

[10] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv*, 2024. 6

[11] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022. 6

[12] Michael Poli, Stefano Massaroli, Atsushi Yamashita, Hajime Asama, Jinkyoo Park, and Stefano Ermon. Torchdyn: Implicit models and neural numerical methods in pytorch. *Physical Reasoning and Inductive Biases for the Real World at NeurIPS*, 2021. 4

[13] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *ICML*, 2023. 2, 5, 6

[14] 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. *TMLR*, 2024. 4, 5, 6

[15] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 3, 6

[16] Kaiyu Yang, Jacqueline H Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. A study of face obfuscation in imagenet. In *ICML*, 2022. 6

[17] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *CVPR*, 2025. 6