TILED RESIDUAL FLOW: MEMORY-BOUNDED EXACT-TRANSFORM GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

High-resolution image-space generative models are limited by activation memory that grows with image area, while latent-space models reduce memory only by relying on a learned decoder. We propose Tiled Residual Flow, a VAE-free architecture that decomposes an image into a deterministic low-resolution base and a high-resolution residual with exact additive reconstruction. A global flow models the base; a shared tile-local flow models residual tiles conditioned on crops of the upsampled base. Peak residual-stage memory is set by tile geometry rather than image size: the measured per-step training floor is $10.7\times$ below a full-frame counterpart at 512×512, growing to $167\times$ at 2048×2048, where full-frame training exceeds an 80 GB accelerator at batch size 2. Ablations at base resolution 128 show that removing local base conditioning nearly doubles FID, while the stitching rule, halo width, tile size and model width have no detectable effect on FID; halving the base resolution to 64 roughly halves FID at the same parameter count. On FFHQ at 512×512, with 7.4 M- and 7.6 M-parameter models trained on the same data, this memory bound costs no FID we can detect: the tiled model at base resolution 64 matches the best checkpoints of a full-frame flow-matching baseline whose receptive field is wider than the frame, after the same 350k full-resolution training steps (FID 56.8 against 57.8) and remains level at equal training compute, under a shared evaluation protocol and with no state-of-the-art claim.

1 INTRODUCTION

The activation memory of full-frame image-space generators grows with image area, batch size and model width, so at high resolution they run out of memory headroom. Latent generative models (Rombach et al., 2022; Peebles & Xie, 2023) reduce this memory by working in a compressed latent, but their sample quality is then bounded by the autoencoder that encodes the training images and decodes the samples.

Exact transforms offer a different trade-off. A fixed operator $T(x) = (B, A)$ with inverse $T^{-1}(B, A) = x$ removes the learned decoder: the generator models the joint distribution of (B, A), and reconstruction is deterministic and parameter-free. This paper uses a Laplacian-style decomposition, $B_M = \mathrm{downsample}(x_N, M\times M)$, which lowpass filters before each factor-of-two subsampling, and $A_N = x_N - \mathrm{upsample}(B_M, N\times N)$, where $M \ll N$ and the inverse is $x_N = \mathrm{upsample}(B_M, N\times N) + A_N$. The base B_M has a small spatial size; the residual A_N has the target resolution and holds whatever the downsample-upsample round trip does not reproduce, including part of the frequency band the base could represent.

Our central design choice is that the residual model is shared and local (Figure 1). Rather than a single full-frame residual U-Net at $N\times N$, we train one residual flow over fixed-size tile crops of A_N conditioned on local crops of the upsampled base plus normalized position channels. At sample time, we evaluate the residual model on a fixed, non-overlapping tile grid with halo context and stitch with a Hann overlap-add. The residual model's activation memory is set by the tile geometry, not by N. An exact-transform formulation with a full-frame residual model gains nothing in memory; the contribution is the tile-locality of the residual, which changes what scales with N.

Empirical scope. At 64×64 full-frame RGB flow matching beats our two-stage recipe on edge quality, a combined quality score, FID and KID, both at a similar parameter count and at half

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.

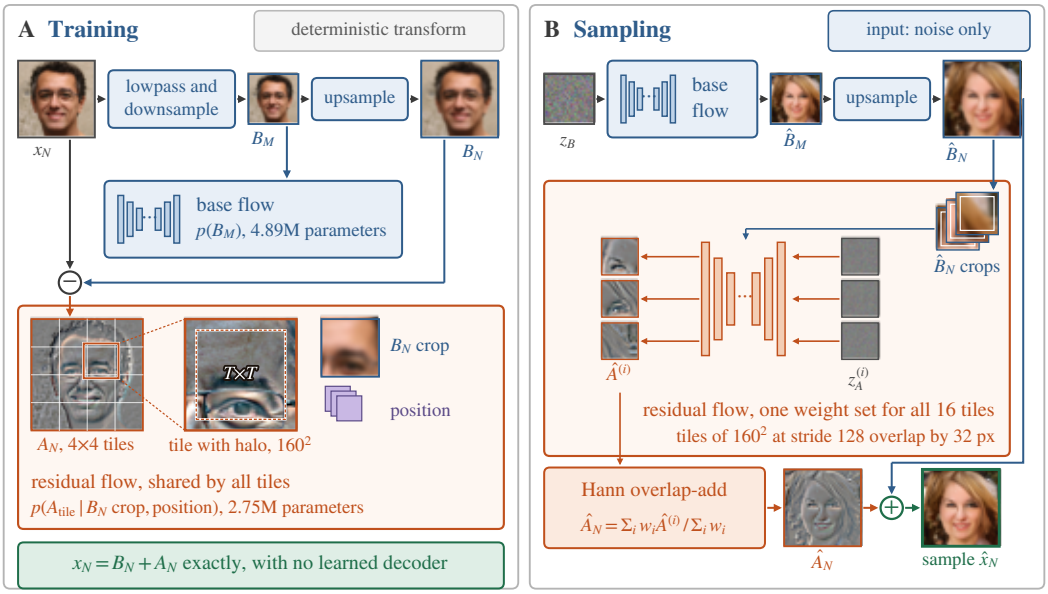


Figure 1: Tiled Residual Flow. Training uses a deterministic base/residual decomposition $x_N = B_N + A_N$. Sampling uses noise only: a low-resolution base flow generates $\hat{B}_M$, a residual flow shared by all tiles generates residual tiles conditioned on crops of $\hat{B}_N$ and per-tile position channels, and Hann overlap-add plus addition gives $\hat{x}_N = \hat{B}_N + \hat{A}_N$, clipped to $[0, 1]$. There is no learned decoder between the representation and the image. Panels are data and model outputs at $M = 64$: panel A decomposes a held-out image chosen for visible fine detail, and the sample shown is the most face-like of eight drawn at a fixed seed, not a random or median one. Residual panels use a fixed $2.3\times$ display gain.

of it (Section 5.1), so we make no low-resolution claim. At 512×512 we compare against full-frame RGB flow matching with matched data and parameters (Section 5.2), using $n = 3500$ FID/KID against a held-out FFHQ validation split, native-512 and downsampled quality metrics, and diversity. Two-level full-frame baselines, whose receptive field spans 17% of the frame width, fail on global structure, so our main comparison is against a five-level baseline whose receptive field is wider than the frame. At base resolution 64 the tiled model matches its best-checkpoint FID and beats all fourteen of its checkpoints on diversity and on low-resolution, downsampled-edge and color statistics, while four of them beat it on the native 512 radial spectrum and thirteen on native 512 edge statistics.

Contributions.

1. A VAE-free, image-space exact-transform formulation: a base flow over a lowpass-filtered, downsampled image plus a shared tile-local flow over the residual, with halo conditioning and overlap-add stitching.
2. A measured memory floor: the minimum footprint of one training step is set by tile geometry rather than image size, and the ratio to a full-frame counterpart grows with resolution.
3. An ablation result. Of the design choices the method combines, two change FID at $M = 128$, local base conditioning and base resolution; the stitching rule, halo width, tile size, position channels and model width show no FID effect we can detect, and stitching and the halo improve seam continuity.
4. A comparison at 512 against full-frame flow matching with matched data and parameters. At base resolution 64 the tiled model matches the best-checkpoint FID of a five-level baseline whose receptive field is wider than the frame, also at equal training compute, and leads it between 7 and 15 GPU-hours.

2 RELATED WORK

Flow matching. Flow matching trains a velocity field whose ODE transports a noise distribution to the data distribution (Lipman et al., 2023; Liu et al., 2023; Tong et al., 2024). We train every model on the straight-line path between independently drawn noise and data samples: the conditional optimal-transport path of Lipman et al. (2023) with $\sigma_{\min}{=}0$, which is also the linear-interpolation objective of Liu et al. (2023) without reflow. We do not use the minibatch optimal-transport coupling of Tong et al. (2024). The same objective trains the base flow, the residual flow and every full-frame baseline, and we integrate the ODE with Euler steps.

Latent generative models and learned decoders. Latent diffusion (Rombach et al., 2022) and diffusion transformers (Peebles & Xie, 2023) train a generative model on the latent grid of a pretrained autoencoder, KL- or VQ-regularized in latent diffusion and the Stable Diffusion VAE in DiT. This sidesteps full-resolution memory, but the final image is then what the decoder makes of the generator's latent. We model a fixed, invertible transform of the image instead, as wavelet diffusion does with Haar subbands (Phung et al., 2023), so the map from generator output to image is closed-form.

Laplacian multi-scale generation. Laplacian decomposition has a long history in image processing (Burt & Adelson, 1983) and in generation, from LapGAN (Denton et al., 2015) to LapFlow (Zhao et al., 2026), so the framing is not novel on its own. LapFlow builds a Laplacian pyramid on the VAE latent of an image and processes the scales in parallel through a mixture-of-transformers backbone with causal attention across scales. It reports better sample quality with fewer GFLOPs than single-scale and multi-scale flow matching baselines and scales to $1024{\times}1024$; an image-space variant of LapFlow trails its latent one. We differ from its latent model on five axes: (i) representation: an exact transform of the image, not of a VAE latent; (ii) backbone: separate base and residual U-Nets rather than a shared mixture-of-transformers, so each scale is sized independently; (iii) scale handling: one low-resolution global base plus a shared local residual evaluated tile by tile, rather than a multi-scale token sequence with cross-scale attention; (iv) decoder: the image is $x_N = B_N + A_N$ in pixel space, not pyramid addition in latent space followed by a learned VAE decoder; (v) efficiency: a memory floor set by tile geometry in residual training and sampling, not fewer sampling GFLOPs. We run no side-by-side comparison against it.

Cascades and generation by parts. Generating a low-resolution image and then a high-resolution image conditioned on its upsampling is the cascade of SR3 (Saharia et al., 2023) and cascaded diffusion (Ho et al., 2022). Our two stages follow that split, but the second stage generates only the exact additive residual, tile by tile, so no network runs on the full frame at full resolution. Generating by parts with coordinate conditioning, which lowers memory, goes back to COCO-GAN (Lin et al., 2019), and InfinityGAN (Lin et al., 2022) synthesizes images of any size patch by patch at constant memory; both are adversarial, and our parts are flow samples of an exact residual conditioned on crops of a generated base. Inf-DiT (Yang et al., 2024) upsamples a given image with block attention whose memory grows with the image side, and Matryoshka Diffusion (Gu et al., 2024) denoises several resolutions jointly in pixel space; neither generates an exact residual with a shared local model. Patch Diffusion (Wang et al., 2023) trains on random crops with per-pixel coordinate channels but samples the full frame. Patch-DM (Ding et al., 2024) and Arakawa et al. (2023) train and sample diffusion models patch by patch to cut memory, the first with a feature collage across neighboring patches and a learned global embedding, the second with a one-hot position and a downsampled copy of the image; their patches carry the full image at full resolution, whereas ours carry only the exact residual above a generated base. MultiDiffusion (Bar-Tal et al., 2023) runs a pretrained model on overlapping crops and, at every denoising step, sets each pixel to a weighted average of the crop predictions. With Hann weights this average is our stitch. Crossing per-step or final fusion with per-tile or shared noise at $M{=}128$, the four FIDs span 111.8 to 113.5 (Appendix O), so per-step fusion gives no gain we can detect.

Pixel-space, decomposed and decoder-free models. We place the closest recent work on three axes. On where generation happens, PixelDiT (Yu et al., 2026), JiT (Li & He, 2026), simple diffusion (Hoogeboom et al., 2023) and HDiT (Crowson et al., 2024) are pixel-space and decoder-free, as we are, but every network pass takes the full frame. In PixelDiT, simple diffusion and HDiT acti-

vation memory grows with image area; JiT instead keeps 256 tokens at every resolution by growing the patch size. PixelDiT also separates global semantics and layout, modeled at the patch level, from texture, refined at the pixel level. On how the image is decomposed, Fractal Generative Models (Li et al., 2025) recursively compose generative modules for pixel-by-pixel generation, whereas ours is a fixed two-level decomposition whose purpose is a memory bound. On whether the inverse is learned, RAE (Zheng et al., 2026) replaces the VAE with a frozen pretrained representation encoder and a trained decoder. This is the sharpest contrast: RAE keeps a learned decoder and improves the latent it decodes, while we remove the learned decoder.

3 METHOD

3.1 EXACT-TRANSFORM BASE/RESIDUAL REPRESENTATION

Let $x_N \in [0,1]^{3\times N\times N}$ be a target image at resolution N. Choose a base resolution M with $N/M = 2^k$. Define

$$B_M = \text{downsample}(x_N, M\times M) = (\text{ds}_2)^k(x_N), \tag{1}$$

$$B_N = \text{upsample}(B_M, N\times N) = (\text{us}_2)^k(B_M), \tag{2}$$

$$A_N = x_N - B_N, \tag{3}$$

where ds_2 is anti-aliased stride-2 downsampling, comprising a 5-tap depthwise-separable Gaussian convolution with σ=1 (reflection padding) followed by stride-2 subsampling, and us_2 is bilinear $2\times$ upsampling with half-pixel alignment. Stacking $k = \log_2(N/M)$ identical factor-of-two stages keeps the construction parameter-free and exposes a single hyperparameter M.

The decomposition is exact in arithmetic:

$$x_N = B_N + A_N \quad \text{by construction.} \tag{4}$$

The base B_M has size $M\times M$; the residual A_N has size $N\times N$ and holds whatever the downsample-upsample round trip does not reproduce, including part of the frequency band the base could represent. Because the inverse is this fixed sum, no learned decoder stands between the representation and the image: there is no decoder to train and no decoder reconstruction error.

3.2 BASE FLOW $p(B_M)$

The base flow models B_M with flow matching. Architecturally it is the two-level member of the RGB U-Net family used for the full-frame baselines (Section 5.2), instantiated at the smaller spatial size $M\times M$. We denote the U-Net base-channel width by C; C=64 uses 64 channels at the input block and channel multipliers $(1, 2, 4)$ across its three resolution stages. Because M is fixed and small, the base flow's compute and memory are roughly $(M/N)^2$ times those of a full-frame model, so it is cheap regardless of how big N becomes.

3.3 TILED RESIDUAL FLOW $p(A_{\text{tile}} \mid B_{\text{ctx}}, \text{pos})$

The residual flow operates on fixed-size tile crops, not on the full $N\times N$ residual. Choose a central tile size T such that $T \mid N$, and a halo h. The image is partitioned into $(N/T)^2$ non-overlapping *central* tiles of size $T\times T$. Each central tile is paired with an *effective tile* of size $(T+2h)\times(T+2h)$ obtained by extending the central tile by h pixels on every side, with reflection padding at the image boundary.

The residual model takes three inputs: the residual flow state $A_t \in \mathbb{R}^{3\times(T+2h)\times(T+2h)}$, a base-context crop $B_{\text{ctx}} \in \mathbb{R}^{3\times(T+2h)\times(T+2h)}$ obtained by extracting the same effective region from B_N, and three position channels $\text{pos} \in \mathbb{R}^{3\times(T+2h)\times(T+2h)}$ that broadcast the normalized tile center and size $(c_y/N, c_x/N, T/N)$. These six conditioning channels are concatenated with A_t at the U-Net's input projection.

The residual flow is a single model shared by all tiles: position information enters only through pos and through the local base context, not through tile-specific weights. This lets all tiles of an image be sampled in parallel and keeps the parameter count independent of N.

3.4 HALO CONTEXT AND OVERLAP-ADD RECONSTRUCTION

Two stitching modes are implemented. The simple mode (*central stitching*) takes only the central $T\times T$ region of each predicted effective tile and copies it into the canvas with no overlap. The default mode (*windowed stitching*) keeps the full $(T+2h)\times(T+2h)$ prediction and combines tiles with a Hann window:

$$w(u,v) = \frac{1}{2}\left(1-\cos\frac{2\pi u}{S-1}\right)\cdot\frac{1}{2}\left(1-\cos\frac{2\pi v}{S-1}\right), \quad S = T+2h, \tag{5}$$

with each factor floored at $\epsilon{=}10^{-6}$ so the normalization never divides by zero. The canvas accumulates $\hat{A}_N(y,x) = \sum_i w_i(y,x)\cdot\hat{A}^{(i)}(y,x)/\sum_i w_i(y,x)$, where the sum runs over tiles whose effective region contains (y,x). Adjacent tiles' effective regions overlap by $2h$ along each shared boundary, with up to four tiles' regions meeting at each $h\times h$ corner of a central region; the central $T\times T$ regions themselves partition the image without overlap. Averaging independently sampled tiles would shrink the standard deviation near seams to 0.88 of its value elsewhere; at $M{=}128$ we measure 1.011, partly explained by overlapping tiles seeing nearly the same crop of the shared base, although the measurement pools samples with different bases (Appendix N).

3.5 SAMPLING

Sampling runs the base flow, upsamples its output, and evaluates the residual flow on batches of effective tiles, invoking it on a $(K, 3, T{+}2h, T{+}2h)$ tensor for K tiles per call (Algorithm 1, Appendix K). At $N{=}512$ with $T{=}128$ and $h{=}16$ all 16 tiles fit in one call; at larger N the tiles are processed in batches of capped size, which bounds the residual model's activation memory (Section 4.2).

3.6 COMPLEXITY AND MEMORY SCALING

Memory. Activations of a full-frame U-Net at $N\times N$, residual or RGB, scale with N^2; those of the tiled residual U-Net scale with $(T{+}2h)^2$ and the number of tiles per call, independently of N. What grows with N in our implementation holds no activations: at sampling, the upsampled base $\hat{B}_N$, the stitching buffers and the per-tile state, about 130 bytes per pixel; in training, the residual and upsampled base of each sampled image, which the trainer copies whole before cutting its tile. Section 4 measures the model step on one tile, the minimum footprint of a training step.

Compute. Per-image sample cost is $C_{\text{base}}(M) + (N/T)^2\cdot C_{\text{tile}}(T+2h)$. Since $(N/T)^2(T+2h)^2 = N^2(1+2h/T)^2$, the residual pass processes $1.56\times$ the pixels of a full frame at $T{=}128$, $h{=}16$, so the method saves memory, not compute, at the same model width.

4 SYSTEMS AND SCALING

Training-step memory is measured with the fp32 microbenchmark of Appendix H on a single NVIDIA H200 (141 GB HBM3e; 139.8 GiB usable).

4.1 THE MEMORY FLOOR OF A TRAINING STEP

Peak memory at a fixed batch says little about resolution, since a batch of tiles is a fraction of an image; what scales is the *floor*, the minimum footprint of one optimizer step, measured in Table 1 for one image against one tile.

The full-frame floor scales as N^2, measured at $3.93\times$ then $3.98\times$ per doubling. The five-level baseline of Section 5.2 needs 3.63, 14.21 and 56.56 GiB, ratios of $11.0\times$, $43.1\times$ and $171.6\times$, so depth does not change this scaling (Appendix H). The tiled model step is fixed by $(T{+}2h)^2$: we measure it at $N{=}512$, and it is independent of N by construction rather than by measurement at 1024 and 2048; the trainer's whole-image copies (Section 3.6) add 24 bytes per pixel per sampled image, 0.09 GiB at 2048. The practical consequence appears at 2048, where a full-frame step needs 55.02 GiB for a single image, beyond a 40 GB accelerator, and a measured 109.96 GiB at batch 2, beyond an 80 GB one, while the tiled model step stays at 0.33 GiB.

Table 1: Minimum footprint of one fp32 training step on an H200, with no out-of-memory events. Full-frame is the two-level RGB $C{=}64$ generator (4.89 M parameters) on one image; tiled is the $C{=}48$ residual flow (2.75 M) on one $(T{+}2h)^2$ tile at $T{=}128$, $h{=}16$, measured at $N{=}512$ and independent of N by construction. The ratio is not width-matched; against the five-level baseline it is $11.0\times$ to $171.6\times$. Optimizer state is excluded; with Adam state the ratios are $10.2\times$ to $157.3\times$ (Appendix H).

N	full-frame, one image	tiled, one tile	ratio
512	3.52 GiB	0.33 GiB	$10.7\times$
1024	13.82 GiB	0.33 GiB	$41.9\times$
2048	55.02 GiB	0.33 GiB	$166.9\times$

The systems claim is the floor, not speed. At matched peak memory the two-level $C{=}64$ full-frame model trains faster, at 125 to 131 images per second against at most 114 image-equivalents (16 tiles each) for the tiled model (Appendix P). Tiling gives a step that fits once N outgrows the accelerator, not throughput at 512.

4.2 Sampling-time memory at fixed tile geometry

Holding $T{=}128$ and $h{=}16$ fixed and varying only N, the residual stage's peak sampling memory per image grows $1.82\times$ over a $16\times$ increase in pixels, against $14.37\times$ for the two-level $C{=}64$ full-frame model; the base flow, whose memory depends only on M, is not included.

Two qualifications apply. For a single image at $N{=}512$ tiling needs more memory (0.581 against 0.465 GiB, random weights, 10 solver steps), with the crossover between 512 and 1024; and the bound requires capping how many tiles are processed per call, since all 256 tiles of a 2048 image at once peak at 8.61 GiB, above the 6.69 GiB full-frame peak.

5 Experiments

Quality metrics. *edge*, *lowres*, *color*, *radial* are edge magnitude histogram L_1, RBF MMD^2 on 16×16 downsamples of the first 512 samples and validation images, 3-channel color histogram L_1, and radial Fourier power L_1 against the validation split; *balanced* is a weighted combination (Appendix C). Subscripts $_{512}$ and $_{64}$ mark native-512 and 64×64-downsampled values. *div* is the mean pairwise L_2 distance between 16×16 downsamples (higher means more variation). FID (Heusel et al., 2017) and KID (Bińkowski et al., 2018) use Inception-v3 (Szegedy et al., 2016) features from torchvision's IMAGENET1K_V1 weights, not the 2015-12-05 Inception-v3 model that Heusel et al. (2017) used, so our values are not comparable to published FID.

5.1 Baseline study at 64×64

On FFHQ-64 (Karras et al., 2019), full-frame RGB flow matching beats our two-stage recipe on edge quality, balanced score, FID and KID, both at a similar parameter count (FID 31.28 against 38.40 for $C{=}96$) and with half our parameters (31.03 for $C{=}64$), so we make no 64×64 claim (Appendix C).

5.2 Comparison at 512×512

Every model trains on the same 30,000 images, the first 30,000 of our 63,000-image FFHQ training split, with the straight-line flow-matching objective of Section 2 and AdamW at a constant learning rate of 2×10^{-4}; full-frame models and residual flows train for 350k steps and base flows for 100k. The tiled model pairs a base flow with the $C{=}48$ residual flow ($T{=}128$, $h{=}16$) and is reported at base resolution $M{=}64$ unless stated; that setting comes from the base-resolution row of Table 3, scored on the same validation split. Batch sizes, and so samples seen, differ between the paths (Appendix D).

Table 2: Comparison at 512, n=3500 samples against the 3,500 held-out validation images, mean and standard deviation over training seeds (three for the tiled and two-level C=64 rows, two for the five-level rows, one for C=80). Tiled rows pair a 100k-step base with a 350k-step residual flow. Rows are at 350k steps except the best checkpoints, chosen by FID on this split: 300k and 350k for the two five-level seeds, 300k for C=80. Lower is better except for div, on which real validation images score 8.81. Per-seed values of every row and all nine metrics are in Appendices E and L.

configuration	FID	KID	div	edge$_{64}$	edge$_{512}$	lowres$_{512}$	color$_{512}$
two-level C=64	$298.9_{\pm 25.7}$	$0.381_{\pm 0.039}$	$4.79_{\pm 0.33}$	$0.282_{\pm 0.022}$	$0.330_{\pm 0.147}$	$0.105_{\pm 0.074}$	$1.52_{\pm 0.17}$
two-level C=80, best	261.9	0.327	5.58	0.250	0.188	0.078	1.23
five-level	$66.6_{\pm 2.5}$	$0.054_{\pm 0.001}$	$5.04_{\pm 0.33}$	$0.171_{\pm 0.001}$	$0.369_{\pm 0.065}$	$0.071_{\pm 0.052}$	$1.27_{\pm 0.10}$
five-level, best	$57.8_{\pm 9.9}$	$0.046_{\pm 0.013}$	$6.18_{\pm 1.29}$	$0.132_{\pm 0.053}$	$0.344_{\pm 0.030}$	$0.031_{\pm 0.004}$	$0.89_{\pm 0.45}$
tiled M=128	$104.1_{\pm 7.3}$	$0.109_{\pm 0.014}$	$6.70_{\pm 0.21}$	$0.162_{\pm 0.033}$	$0.308_{\pm 0.120}$	$0.021_{\pm 0.011}$	$0.67_{\pm 0.22}$
tiled M=64	$56.8_{\pm 2.2}$	$0.041_{\pm 0.002}$	$7.65_{\pm 0.09}$	$0.070_{\pm 0.011}$	$0.396_{\pm 0.010}$	$0.006_{\pm 0.003}$	$0.28_{\pm 0.05}$

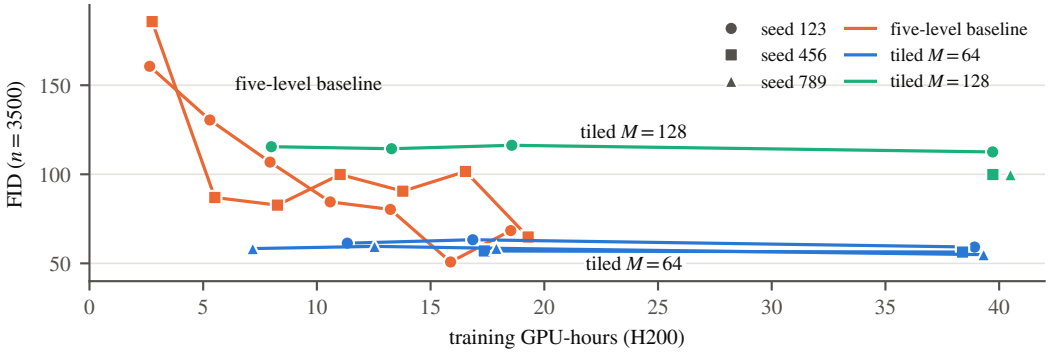


Figure 2: FID against training GPU-hours at 512 (n=3500; H200, pro rata for intermediate checkpoints), joined within each training seed; each tiled checkpoint uses its own seed's base. Runs end at 350k steps; the residual flows and the seed-789 base at M=64 trained without compilation. Values are in Tables 7 and 9.

Baselines. The full-frame baselines are the RGB U-Net family of the base flow. The two-level baselines, at C=64 (4.89 M parameters), C=80 (7.63 M, matched to the tiled model's 7.64 M) and C=96 (10.98 M), have a receptive field 87 pixels wide, 17% of the frame width, and fail on global structure: no checkpoint of the C=80 model scores below 261.9, while a perfect 32×32 base upsampled to 512 already scores 224.9 (Appendix I). The five-level baseline, with channel multipliers $(1, 2, 2, 2, 2, 2)$ and 7.40 M parameters, sees 791 to 823 pixels, wider than the frame, and otherwise follows the C=80 recipe (Appendix E). We train it with two seeds and evaluate every 50k checkpoint.

Quality. The best five-level checkpoints score 50.79 (300k) and 64.81 (350k), and all three tiled seeds at 350k fall between them (54.89 to 59.15, mean 56.8 ± 2.2 against 57.8), so the two match on FID; KID is close as well (0.046 at the best checkpoints against 0.041), while the tiled model's diversity stays higher (7.65 against 6.18). At the final checkpoint the tiled model is lower in all six seed pairs (56.8 against 66.6 ± 2.5), but after 100k steps the baseline's FID moves by up to 37 between adjacent checkpoints and neither model uses weight averaging, so with two baseline seeds we do not read this as a win. On diversity, downsampled edge, low-resolution MMD and color at both resolutions every tiled seed beats all fourteen five-level checkpoints (Table 8), four of these five already hold for the seed-123 and seed-456 bases alone (Appendix L), and one checkpoint beats every tiled seed on FID and KID, four on the radial spectrum and thirteen on native 512 edge.

Compute and memory. In Figure 2 the tiled model at M=64 reaches 58.3 by 7.2 GPU-hours and stays between 54.9 and 63.3, while the baseline stays above 80 until 15.9 GPU-hours and falls below 65 at only two of its fourteen checkpoints. At the baseline's full budget the two are level: the tiled seeds at 150k steps (16.9 to 17.9 GPU-hours) score 63.3, 57.0 and 58.4, between the baseline's best checkpoints and below its final ones (68.4 and 64.8 at 18.5 and 19.3). Training the tiled model

Table 3: Ablations of the design choices, one variable at a time, at a matched step budget; see the text for the three rows that also change the parameter count, the batch or the base. The reference run† is one evaluation shared by the halo, tile and width rows, not three measurements. The stitching row uses the 350k checkpoint and the rest are 100k runs, so they are not directly comparable. The h=0 run also switches to central stitching, since no overlap exists at h=0, so it varies two things.

variable	settings and FID	FID effect
stitching (350k)	Hann 112.56, central crop 112.26	none detected
halo	h=0 113.24, h=16 114.36†, h=32 112.98	none detected
tile size	T=64 110.92, 128 114.36†, 256 111.42	none detected
width	C=32 114.96, 48 114.36†, 64 110.32	none detected
position channels	zeroed 113.08	none detected
local base crop	zeroed 223.60	+109 FID
base resolution	M=64 61.33	−53 FID

on to 350k costs on average 2.06× the baseline's GPU-hours, an upper bound (Appendix F), for a further 2.8 FID. Per image of area, 16 tiles, the tiled step needs more memory at the training batch (4.06 against 3.53 GiB in fp32, Appendix H); what tiling bounds is the per-step floor of Section 4.1.

The M=128 configuration. At M=128 the base flow is the two-level U-Net, whose 87-pixel receptive field spans 68% of the frame width, and the tiled model's samples show distorted global structure (Appendix T). There the tiled model trails the five-level baseline at 350k (104.1 ± 7.3) and at 8.0, 13.3 and 18.6 GPU-hours (Appendix E); at M=64 the receptive field is wider than the base. Base resolution also changes how much of the image the base must model, so for the tiled model we read receptive-field coverage as consistent with the result, not as its demonstrated cause.

5.3 Which design choices carry the result

The method combines established ingredients; we ask which of them the result depends on. We ablate each at M=128 over 100k residual steps with one base checkpoint and seed shared (stitching uses the 350k checkpoint), so only the named variable moves, with four exceptions: the h=0 run also stitches centrally, the width rows change the residual flow's parameter count, the T=256 row trained at half the batch because the full batch did not fit, and the base-resolution row trains a new base. The three-seed standard deviation of 7.34 FID at M=128 comes from retraining both stages and overstates the noise here, where every row shares a base. Excluding the base-crop run, the eight base-sharing runs span 4.64 FID, and a second residual seed on a shared base, at M=64, moves FID by 3.7, so we treat differences below roughly 10 FID as undetermined. At this budget the whole residual improves FID by 13.0 over the base alone, so these rows rule out only large effects.

Only two variables move FID: zeroing the local base crop costs 109 FID, and coarsening the base to M=64 improves it by 53. Stitching, halo, tile size, position channels and width stay below the 10 FID threshold, width spanning 4.64 FID over a 4.0× range of residual parameters.

Stitching and the halo change the seam instead. On the 350k checkpoint central stitching raises the boundary L_1 (Appendix M) by 41% (0.01814 against 0.01284), and the h=0 run, which also stitches centrally, raises it by 73% (0.02216 against 0.01279) with FID unchanged; the other halo, tile, width and position rows lower it by at most 15%.

5.4 Base resolution

Base resolution is the only change in Table 3 that detectably improves FID. At M=64 the same 7,639,110 parameters reach FID 61.33 at 100k against 114.36 at M=128, and at 350k, over three retrains of both stages, 56.8 ± 2.2 against 104.1 ± 7.3. Diversity rises (7.65 against 6.70), so the gain is not collapse, but native 512 edge is worse on average (0.396 ± 0.010 against 0.308 ± 0.120).

A coarser base leaves more of the image to the tile-local residual flow, and the residual's measured contribution grows accordingly. Against each seed's own base-only control the residual improves FID at M=128 by 14.76, 23.28 and 28.34 ($t(2)$=5.59, p=0.031, 95% interval $[5.1, 39.2]$); at M=64

it improves FID by 106.04 and 100.77 on seeds 123 and 456, four to seven times what it adds at $M{=}128$ on those seeds. An oracle base, obtained by pushing real images through the decomposition, scores 83.98 at $M{=}128$ and 137.04 at $M{=}64$, against a real-versus-real FID of 9.53. At $M{=}64$ the tiled model scores below that oracle, so the residual adds detail beyond an exact base at that bandwidth, and at $M{=}128$ it does not (Appendix I). A run on AFHQ, at $M{=}128$ against a two-level baseline, is in Appendix J.

6 LIMITATIONS

Low-resolution superiority does not hold. At $64{\times}64$ the two-stage Laplacian residual recipe loses to full-frame RGB flow matching on edge quality, balanced score, FID and KID (Section 5.1), so the contribution is specific to the high-resolution regime.

Metric, claim, and seed scope. FID and KID at $n{=}3500$ order models within this study and are not comparable to published FID-50k numbers; the claim and non-claim summary is in Appendix A. The comparison with the five-level baseline rests on two baseline seeds and three tiled seeds. The baseline's FID moves by up to 37 between adjacent checkpoints, its best checkpoints and our choice of $M{=}64$ were both made on the evaluation split, and no model uses weight averaging. One baseline checkpoint scores better FID and KID than every tiled seed, and thirteen of the fourteen score better native 512 edge statistics. Nearest-neighbor distances show no tail of copies for either model (Appendix E). Absolute quality is weak: the tiled model reaches FID 56.8 with 7.6 M parameters and 30,000 training images, level with a full-frame model of the same size, and a competitive generator would need larger models and more data.

Cost, and where tiling does not help. A full tiled run at $M{=}64$ used on average $2.06\times$ the five-level baseline's GPU-hours, an upper bound, since the residual flows and the seed-789 base trained without compilation while the baseline was compiled (Appendix F). At $M{=}64$ the tiled model is level with the baseline at its full budget and leads between 7 and 15 GPU-hours, while at $M{=}128$ it trails the five-level baseline at every budget (Figure 2). At 512, where we evaluate quality, tiling needs more memory per image in training and, against the two-level model, in sampling (Section 4.2); the memory advantage is the per-step training floor and, at sampling, a bound that needs capped tiles per call, both claims about larger N.

Resolution and external baselines. In a 50k-step pilot at 1024 on 5,000 images, with $M{=}256$, $T{=}256$, $h{=}32$ and a $C{=}32$ residual flow, the residual does not help: FID 225.67 against 215.97 for the base alone. Whether that is a budget effect or a property of the geometry at that scale is open. Full 1024 training, conditional generation and downstream applications are out of scope, as are external baselines: a pixel-space diffusion transformer scaled to our budget and a latent diffusion model at matched memory are the fair comparisons, and under-tuned reimplementations would misrepresent them.

Datasets and long-range structure. FFHQ and AFHQ are both aligned single-object datasets. The residual flow sees only a $(T{+}2h)^2$ crop, so long-range structure must come from the base, and scenes with strong long-range structure, such as urban, indoor, multi-object or text-bearing images, may expose that limit (Appendix U).

7 CONCLUSION

A shared tile-local residual bounds the memory floor of a training step by tile geometry rather than image size, and the gap to a full-frame step widens with resolution: $10.7\times$ at 512 and $166.9\times$ at 2048. At 512 and base resolution 64 the tiled model matches the best checkpoints of a full-frame baseline whose receptive field is wider than the frame, also at equal training compute, and leads it between 7 and 15 GPU-hours. Ablation at $M{=}128$ locates FID in two places, local base conditioning and base resolution; stitching, halo, tile size, position channels and width show no FID effect we can detect, and of these only stitching and the halo raise the seam metric.

AI USE STATEMENT

We used generative AI tools for four tasks with required disclosure: implementing methods, designing or giving feedback on experiments, proposing or refining hypotheses, and interpreting results, and not for the others (synthetic data, theory or conceptual frameworks, mathematical claims and proofs, translation, dataset cleaning, qualitative analysis). Among tasks with recommended disclosure, we used them for code, figures, experimental parameters, drafting, editing and structuring the paper, literature search and summaries, and reference formatting.

We have reviewed all AI-assisted work. The research question, the claims, and the decisions about what to run and what to report are the authors' own. AI-assisted code was tested, and every number in this paper was produced by the experiments described and checked against the stored results before being written down. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The authors have read and follow the ICLR Code of Ethics. This work trains unconditional generators of human faces on FFHQ (Karras et al., 2019) and of animal faces on AFHQ (Choi et al., 2020), used under their non-commercial terms (Appendix V). FFHQ consists of Flickr photographs that their uploaders published under permissive licenses, and it inherits the biases of that website (Karras et al., 2019), which a generator trained on it can reproduce. Its documentation records no consent from the people shown, offers removal to the photos' owners on request, and asks that the dataset not be used to develop face recognition, which this work does not do. Nearest-neighbor checks against the training set find no evidence of copied training images (Appendix E), and the models are small and produce visibly distorted faces (Appendix T). Methods that make high-resolution generation cheaper can also make deceptive images of people cheaper to produce; this paper does not target photorealism and introduces no capability specific to impersonation. The work involves no human-subjects research.

REPRODUCIBILITY STATEMENT

Section 3 specifies the decomposition, tiling, conditioning and stitching, and Appendix K the sampler. The evaluation protocol of the 512 comparison is fixed: $n{=}3500$ generated samples against the $3{,}500$-image held-out validation split at sampling seed 123, with metric definitions at the start of Section 5; appendices that use another protocol state it. Appendix D lists the training data, batch sizes and steps of the 512 comparison, Section 5.3 the changes made by each ablation, and Appendix F their measured wall-clock. Every model trains with AdamW at a constant learning rate of $2{\times}10^{-4}$ and samples with 50 Euler steps, 50 for the base and 50 for the residual in the tiled model, except where fewer residual steps are marked (Appendices B, Q and R). Appendix E specifies the five-level baseline, its receptive-field measurement and every evaluated checkpoint. The memory floors follow the microbenchmark protocol of Appendix H. The ablation protocol, including the shared base checkpoint and the threshold used to read its rows, is in Section 5.3, and the seed aggregation in Table 2 and Appendix L. We will release the code, configurations and evaluation scripts.

REFERENCES

Shinei Arakawa, Hideki Tsunashima, Daichi Horita, Keitaro Tanaka, and Shigeo Morishima. Memory efficient diffusion probabilistic models via patch-based generation. *arXiv preprint arXiv:2304.07087*, 2023.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In *International Conference on Machine Learning (ICML)*, pp. 1737–1752, 2023.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, 2018.

Peter J. Burt and Edward H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z. Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *International Conference on Machine Learning (ICML)*, pp. 9550–9575, 2024.

Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Zheng Ding, Mengqi Zhang, Jiajun Wu, and Zhuowen Tu. Patched denoising diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Josh Susskind, and Navdeep Jaitly. Matryoshka diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

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 (NeurIPS)*, 2017.

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning (ICML)*, pp. 13213–13232, 2023.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *arXiv preprint arXiv:2502.17437*, 2025.

Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. COCO-GAN: Generation by parts via conditional coordinating. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Chieh Hubert Lin, Hsin-Ying Lee, Yen-Chi Cheng, Sergey Tulyakov, and Ming-Hsuan Yang. InfinityGAN: Towards infinite-pixel image synthesis. In *International Conference on Learning Representations (ICLR)*, 2022.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Hao Phung, Quan Dao, and Anh Tran. Wavelet diffusion models are fast and scalable image generators. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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.

Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch Diffusion: Faster and more data-efficient training of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zhuoyi Yang, Heyang Jiang, Wenyi Hong, Jiayan Teng, Wendi Zheng, Yuxiao Dong, Ming Ding, and Jie Tang. Inf-DiT: Upsampling any-resolution image with memory-efficient diffusion transformer. In *European Conference on Computer Vision (ECCV)*, 2024.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT: Pixel diffusion transformers for image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14273–14282, 2026.

Zelin Zhao, Petr Molodyk, Haotian Xue, and Yongxin Chen. LapFlow: Laplacian multi-scale flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2026.

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

A Claim and non-claim summary

What we claim. (a) A VAE-free, exact-transform, image-space formulation of high-resolution generation with a shared, tile-local residual generator. (b) A training-step memory floor set by tile geometry rather than image size: $10.7\times$ below the two-level full-frame model at 512, $41.9\times$ at 1024 and $166.9\times$ at 2048 (Table 1; $11.0\times$ to $171.6\times$ against the five-level model), and residual-stage sampling memory that grows $1.82\times$ over a $16\times$ increase in pixels at fixed tile geometry (Section 4.2). (c) An ablation result at $M{=}128$: among the design choices, local base conditioning and base resolution change FID, while stitching, halo, tile size, position channels and width show no FID effect we can detect; stitching and the halo improve seam continuity (Section 5.3). (d) At 512 and $M{=}64$, with matched data and parameters: FID level with the best checkpoints of a five-level full-frame baseline whose receptive field is wider than the frame; better diversity, low-resolution, downsampled-edge and color statistics than every one of its fourteen checkpoints; FID level with its best checkpoints at equal training compute, and lower FID from 7 to 15 GPU-hours (Table 2, Figure 2). (e) A $64{\times}64$ negative result: full-frame RGB flow matching beats our two-stage recipe on edge quality, balanced score, FID and KID (Section 5.1).

What we do not claim. (a) State-of-the-art or competitive absolute quality at any resolution; FID and KID are computed at $n{=}3500$ with torchvision Inception weights and order models within this evaluation only. (b) An advantage over Laplacian multi-scale flow matching (Zhao et al., 2026), pixel-space transformers (Crowson et al., 2024; Yu et al., 2026) or latent diffusion (Rombach et al., 2022); we run no side-by-side comparison with any of them. (c) Low-resolution superiority. (d) Training compute or throughput savings: at 512 the two-level full-frame model trains faster at matched memory, and tiling needs more single-image sampling memory. (e) A gain in native 512 edge quality, or in FID and KID over the five-level baseline's best checkpoints. (f) Quality at 1024

or above; our only 1024 quality run is a 50k-step pilot in which the residual flow does not help. (g) Evidence from the duplicate-rate metric, which flags 92% of held-out real images. (h) That tiling improves FID at the five-level baseline's full budget, where the two are level, or at $M{=}128$, where the tiled model trails that baseline. (i) That receptive-field coverage is the demonstrated cause of the gap between $M{=}64$ and $M{=}128$. (j) That the FID nulls of the ablations, measured at $M{=}128$ where the whole residual improves FID by 13, exclude small effects or carry over to $M{=}64$.

B HELD-OUT $n{=}7000$ VALIDATION AND TEST FID

As a robustness check, we recompute FID using $n{=}7000$ generated samples against the union of our held-out FFHQ-512 validation and test splits ($3{,}500 + 3{,}500 = 7{,}000$ images, both disjoint from the training split by construction). This check covers only the two-level $C{=}64$ and $C{=}96$ baselines and the tiled model at $M{=}128$; the five-level baseline, the two-level $C{=}80$ model and the tiled model at $M{=}64$ of Section 5.2 were not rescored on the test split. The main text uses the $n{=}3500$ validation FID. These values remain within-evaluation ordering metrics and are not comparable to published FID-50k benchmarks.

Table 4: Held-out $n{=}7000$ validation and test FID robustness check. Reference images are the union of our held-out FFHQ-512 validation and test splits (7,000 images total, both disjoint from the training split). NFE is the number of residual Euler steps (function evaluations); the base flow always takes 50. Δ is the $n{=}7000$ FID minus the $n{=}3500$ FID, computed before rounding. The ordering matches the main $n{=}3500$ validation evaluation.

configuration	FID $n{=}3500$	FID $n{=}7000$	Δ
two-level $C{=}64$, seed 123	322.04	320.31	-1.74
two-level $C{=}64$, seed 456	303.44	300.38	-3.06
two-level $C{=}64$, seed 789	271.17	268.08	-3.09
two-level $C{=}96$, seed 999	279.60	277.22	-2.39
tiled $M{=}128$, seed 123, NFE$=$50	112.56	107.79	-4.77
tiled $M{=}128$, seed 123, NFE$=$30	112.23	107.46	-4.76
tiled $M{=}128$, seed 456, NFE$=$50	99.87	96.15	-3.71
tiled $M{=}128$, seed 789, NFE$=$50	99.81	96.08	-3.73

Doubling both the generated and the reference set lowers every FID by 1.7 to 4.8 points, consistent with reduced small-sample bias. The ranking of all eight rows is unchanged, the FID separation between the three-seed means of the two-level $C{=}64$ baseline and the tiled model stays at 10.2 pooled standard deviations (10.3 at $n{=}3500$), and NFE$=$30 still scores 0.33 FID below NFE$=$50.

C FFHQ-64 HEAD-TO-HEAD: FULL TABLE

Table 5: FFHQ-64 head-to-head against 2,048 real images. Edge, lowres, radial and balanced are means over three sampling seeds of 4,096 samples each (lowres uses 512 samples and 512 real images); FID and KID use 2,048 samples from one sampling seed. Lower is better for all columns. The two C=96 rows are the best of 36 checkpoints (four training seeds, every 25k steps from 150k to 350k). Both two-stage rows pair a C=64 base flow at 250k steps with the same residual flow, trained for 50k steps on 4,096 images conditioned on their real low-pass images; the retrained-base row trains the base again from a new initialization. Full-frame RGB flow matching beats the two-stage recipe on edge, balanced score, FID and KID; the original-base two-stage row is better on the radial spectrum, and on lowres against C=96. The balanced score is $\mathrm{edge}/0.0929 + 0.5\,\mathrm{lowres}/0.0064 + 0.5\,\mathrm{radial}/0.0076 + 0.25\,\mathrm{color}/0.518 + 0.25\,\mathrm{sat}/0.0028$, where sat, the fraction of output values outside $[0, 1]$, is 0 for every row because samples are clipped before scoring.

configuration	edge	lowres	radial	balanced	FID	KID
RGB C=96 (best edge: seed 123, 250k steps)	0.0483	0.0070	0.0066	1.701	–	–
RGB C=96 (best balanced: seed 789, 200k steps)	0.0727	0.0044	0.0040	1.547	31.28	0.0177
RGB C=64 (final: seed 999, 300k steps)	0.0796	0.0027	0.0047	1.508	31.03	0.0153
Two-stage, original base (250k steps)	0.0996	0.0031	0.0033	1.722	38.40	0.0195
Two-stage, retrained base (seed 123, 250k steps)	0.0827	0.0075	0.0090	2.340	37.86	0.0180

D PER-CONFIGURATION PARAMETER AND COMPUTE BUDGET

Table 6: Per-configuration parameter and compute budget for the 512 comparison. Every model trains on the same 30,000 images, the first 30,000 of our 63,000-image FFHQ training split, which is disjoint from the 3,500 validation and 3,500 test images.

configuration	params	batch	steps	samples seen
two-level C=64	4.89 M	24	350k	8.4 M images
two-level C=80	7.63 M	24	350k	8.4 M images
two-level C=96, seed 999	10.98 M	16	350k	5.6 M images
five-level C=64	7.40 M	24	350k	8.4 M images
base flow, M=128 or 64, C=64	4.89 M	256	100k	25.6 M images
residual flow, C=48, plus base	7.64 M total	256	350k	89.6 M tiles
				(5.6 M image-equivalents, 16 tiles each)

The two-level C=80 model, evaluated under the protocol of Table 2 at every 50k checkpoint, scores FID 321.71, 281.72, 287.13, 302.27, 278.63, 261.88 and 298.43 from 50k to 350k. Table 7 lists every checkpoint of the five-level baseline.

E THE FIVE-LEVEL FULL-FRAME BASELINE

Architecture and training. The five-level baseline is the RGB U-Net family of the base flow and the two-level baselines, with five resolution levels instead of two. Channel multipliers $(1, 2, 2, 2, 2, 2)$ on a base width of 64 give 64 channels at full resolution and 128 below it, a 16×16 bottleneck at 512, and 7,404,675 parameters, 3% fewer than the tiled model's 7,639,110. Data, optimizer, learning rate, batch (24), steps (350k), bf16 autocast and compilation are those of the two-level C=80 run; the base width, the depth and the channel schedule differ. We train two seeds, 123 and 456.

Receptive field. We measure the receptive field of one output pixel by back-propagating from it to the input with GroupNorm removed, since GroupNorm statistics couple every pixel. The two-level U-Net sees 87 pixels at every position: 17% of a 512 frame, 68% of the 128 base at M=128, and more than the 64 base at M=64. The five-level U-Net sees 791 to 823 pixels depending on the

position modulo 32, wider than the 512 frame, although no pixel near a border reaches the opposite border.

Trajectories. Table 7 lists every 50k checkpoint of both seeds, and Table 8 gives all nine metrics at each. After 100k steps adjacent checkpoints differ by up to 36.8 FID, and the best checkpoints are 300k for seed 123 (50.79) and 350k for seed 456 (64.81). Neither the baseline nor the tiled model uses weight averaging. For the tiled model at $M{=}128$ (seed 123) the four checkpoints we evaluated score 115.49, 114.36, 116.28 and 112.56 at 50k, 100k, 150k and 350k, costing 8.00, 13.28, 18.57 and 39.71 GPU-hours; at $M{=}64$ (seed 123) the three span 59.15 to 63.26 (Table 9).

Table 7: The five-level baseline at every 50k checkpoint, $n{=}3500$, seeds 123 and 456. GPU-hours are measured H200 wall-clock of each full run, costed pro rata. Lower is better except for div, on which real validation images score 8.81.

	FID		KID		div		GPU-h	
steps	123	456	123	456	123	456	123	456
50k	160.63	185.74	0.210	0.249	5.08	5.38	2.65	2.76
100k	130.49	86.98	0.154	0.093	4.49	5.19	5.29	5.51
150k	106.85	82.68	0.123	0.081	4.59	5.34	7.94	8.27
200k	84.51	99.88	0.081	0.115	4.76	4.58	10.59	11.02
250k	80.27	90.51	0.085	0.094	5.06	4.65	13.23	13.78
300k	50.79	101.59	0.037	0.113	7.09	4.91	15.88	16.53
350k	68.40	64.81	0.054	0.055	4.80	5.27	18.52	19.29

Table 8: All nine metrics at every checkpoint of the five-level baseline, $n{=}3500$. Lower is better except for div. Every tiled $M{=}64$ seed at 350k (Table 15) beats all fourteen rows on div, edge_{64}, lowres_{512}, color_{512} and color_{64}; one row beats every tiled seed on FID and KID, four on radial_{512} and thirteen on edge_{512}.

checkpoint	FID	KID	div	edge_{64}	edge_{512}	lowres_{512}	color_{512}	color_{64}	radial_{512}
seed 123, 50k	160.63	0.210	5.08	0.182	0.127	0.0439	1.30	1.77	0.0041
seed 123, 100k	130.49	0.154	4.49	0.193	0.211	0.0470	1.49	2.09	0.0046
seed 123, 150k	106.85	0.123	4.59	0.262	0.344	0.0465	1.47	1.72	0.0035
seed 123, 200k	84.51	0.081	4.76	0.198	0.266	0.0314	1.43	1.83	0.0050
seed 123, 250k	80.27	0.085	5.06	0.162	0.109	0.0412	1.14	1.70	0.0040
seed 123, 300k	50.79	0.037	7.09	0.095	0.365	0.0287	0.57	0.71	0.0018
seed 123, 350k	68.40	0.054	4.80	0.171	0.415	0.1080	1.34	2.00	0.0076
seed 456, 50k	185.74	0.249	5.38	0.223	0.069	0.0344	1.10	1.51	0.0014
seed 456, 100k	86.98	0.093	5.19	0.191	0.265	0.0622	1.28	1.56	0.0009
seed 456, 150k	82.68	0.081	5.34	0.150	0.220	0.0271	1.17	1.53	0.0045
seed 456, 200k	99.88	0.115	4.58	0.184	0.135	0.0380	1.35	1.81	0.0035
seed 456, 250k	90.51	0.094	4.65	0.225	0.131	0.0524	1.45	1.83	0.0006
seed 456, 300k	101.59	0.113	4.91	0.203	0.306	0.0469	1.42	1.95	0.0048
seed 456, 350k	64.81	0.055	5.27	0.170	0.323	0.0343	1.20	1.58	0.0056

Nearest training images. For each generated sample we find the nearest of the 30,000 training images in the $16{\times}16$ feature space of the evaluation code. Held-out validation images give the reference: median distance 4.09, first percentile 2.51, minimum 1.93. The five-level baseline's closest samples sit about as far from the training set as the closest real images (first percentile 2.47 to 2.81 over its fourteen checkpoints, minimum 2.03 to 2.41), so there is no tail of copies. Its median is lower (3.15 to 3.63) because its samples vary less (diversity 4.49 to 7.09), and it is highest at the checkpoint with the highest diversity (300k, seed 123). The tiled model at $M{=}64$ has median 3.83 to 4.03, first percentile 2.64 to 2.80 and minimum 1.88 to 2.36 over its three seeds. At this resolution the distances follow the base: the seed-123 base alone gives 3.78, 2.61 and 1.82, and adding its residual moves them to 3.83, 2.64 and 1.88. By eye, the nearest training images of the first eight samples of each model at 350k, found with $64{\times}64$ features, show different people.

Table 9: Every checkpoint of the tiled model evaluated with its own seed's 100k-step base, with its training cost, n=3500. GPU-hours are measured H200 wall-clock of the base run plus the residual training up to that checkpoint, pro rata within the run that produced it; at M=64, seeds 123 and 456 trained the residual in two runs, to 100k and on to 350k. These are the tiled points of Figure 2.

checkpoint	GPU-h	FID	KID	div
M=64, seed 123, 100k	11.34	61.33	0.045	7.59
M=64, seed 123, 150k	16.85	63.26	0.047	7.60
M=64, seed 123, 350k	38.92	59.15	0.041	7.59
M=64, seed 456, 150k	17.36	56.96	0.043	7.76
M=64, seed 456, 350k	38.39	56.33	0.043	7.76
M=64, seed 789, 50k	7.18	58.30	0.043	7.61
M=64, seed 789, 100k	12.54	59.53	0.044	7.61
M=64, seed 789, 150k	17.89	58.45	0.044	7.60
M=64, seed 789, 350k	39.32	54.89	0.039	7.61
M=128, seed 123, 50k	8.00	115.49	0.129	6.91
M=128, seed 123, 100k	13.28	114.36	0.127	6.91
M=128, seed 123, 150k	18.57	116.28	0.131	6.90
M=128, seed 123, 350k	39.71	112.56	0.125	6.90
M=128, seed 456, 350k	39.72	99.87	0.102	6.70
M=128, seed 789, 350k	40.50	99.81	0.099	6.49

F Compute resources for the 512 comparison

Every training run used a single NVIDIA H200. The n=3500 evaluation of the eight rows of Table 4 and the n=7000 robustness check (Appendix B) ran on a single NVIDIA H100 80 GB; every other evaluation ran on a single H200. Wall-clock is read from each run's recorded timing summary or, for evaluation jobs, from the scheduler. Every residual flow, the seed-789 base flow at M=64 and the two-level C=96 model trained without compilation; every other training run was compiled (Appendix S), so the GPU-hour ratios below overstate the cost of the tiled path.

A tiled run's training cost is its 100k base run plus its 350k residual run. At M=64 this is 38.92, 38.39 and 39.32 H200-hours for the three seeds, against 18.52 and 19.29 for the five-level baseline at the same 350k steps, a ratio of 2.06× on the means; at M=128 it is 39.71, 39.72 and 40.50 against 18.33, 18.49 and 18.34 for the two-level C=64 baseline, 2.17× on the means. Both ratios are upper bounds for the reason above.

Not included in the table: the AFHQ runs (17.2 H200-hours of training), the 1024 pilot (5.2), evaluations not listed above, unreported and failed runs, the feasibility pilot (Appendix G), the FFHQ-64 study (Appendix C), and the microbenchmark grid (Appendix H).

G Feasibility pilot: quality and seam metrics

A 50k-step pilot at FFHQ-512 on 5,000 training images compared five configurations: a full-frame RGB reference, bases at $M \in \{128, 256\}$, and a tiled residual paired with each base. The configuration of the ablations (M=128, T=128, h=16, C=48) reached the lowest edge and balanced scores (0.111 and 6.15, against 0.328 and 11.56 for the RGB reference at the same 50k steps). The pilot is not a quality result, since the RGB reference at 50k is far from convergence; its T, h and C carried over to the 350k runs, and M=64 comes from the ablation of Section 5.3. We do not read the sign of the tile-boundary edge excess in Table 11 as evidence of absent seams (Appendix M).

H Microbenchmark protocol and selected memory cells

Protocol. For every (kind, image size N, base resolution M or tile size T, halo h, base channels C, batch size B) configuration, we allocate a synthetic fp32 input batch of the right shape, run two warm-up forward+backward passes on the live model, and time five further iterations with `torch.cuda.Event`. Peak memory is read from `torch.cuda.max_memory_allocated` after a per-configuration reset of the peak-memory counter. Out-of-memory errors are caught and

Table 10: Wall-clock compute in hours per training run, from recorded run summaries, and per evaluation job, from the scheduler; each block names its GPU. Every residual flow trained without compilation; of the other runs, only the two marked did. Subtotals and the total are summed at full precision, so they can differ from sums of the rounded entries. The n=7000 job held two H100s and computed on one.

configuration	wall-clock (h)
Training (H200):	
two-level C=64, seed 123, 350k	18.3
two-level C=64, seed 456, 350k	18.5
two-level C=64, seed 789, 350k	18.3
two-level C=96, seed 999, 350k (uncompiled)	40.5
base flow, M=128, seed 123, 100k	2.7
base flow, M=128, seed 456, 100k	2.7
base flow, M=128, seed 789, 100k	2.7
residual flow, M=128, seed 123, 350k	37.0
residual flow, M=128, seed 456, 350k	37.0
residual flow, M=128, seed 789, 350k	37.8
two-level C=80, seed 123, 350k	30.9
ablation and M=64 runs of Table 3 and Section 5.2 (12 runs)	136.1
training subtotal (H200)	382.6
Training, five-level baseline and M=64 seeds (H200):	
five-level, seed 123, 350k	18.5
five-level, seed 456, 350k	19.3
base flow, M=64, seed 456, 100k	0.7
base flow, M=64, seed 789, 100k (uncompiled)	1.8
residual flow, M=64, seed 456, 100k to 350k	26.3
residual flow, M=64, seed 789, 350k	37.5
subtotal (H200)	104.2
Evaluation (H100 80 GB):	
n=3500 evaluation, the 8 rows of Table 4	10.3
n=7000 robustness check, the same 8 rows	23.7
evaluation subtotal (H100)	33.9
Evaluation (H200):	
five-level baseline, 14 checkpoints (5 jobs)	17.4
M=64 seeds 456 and 789 at 350k	2.4
M=64, 5 checkpoints from 50k to 150k (5 jobs)	6.8
evaluation subtotal (H200)	26.6
total in this table	547.2

Table 11: 50k-step feasibility pilot at FFHQ-512. Quality metrics on 64 samples per configuration, downsampled to 64×64. Seam metrics on the 512 output: seam L_1 is the boundary L_1, seam-grad jump is the seam-gradient jump, and edge excess is the tile-boundary edge excess (same metrics as Table 17, abbreviated for table width). This is a feasibility pilot, not a final quality result; RGB C=64 is severely undertrained at 50k steps for N=512.

configuration	edge$_{64}$	balanced	seam L_1	seam-grad jump	edge excess
RGB full-frame, C=64	0.328	11.56	–	–	–
Base flow, M=128, C=64	0.128	6.35	–	–	–
Base flow, M=256, C=64	0.147	8.62	–	–	–
Tiled, M=128, T=128, h=16, C=48	0.111	6.15	0.0105	-9.2×10^{-5}	−0.0083
Tiled, M=256, otherwise as above	0.145	8.58	0.0096	3.3×10^{-5}	−0.0092

recorded as out-of-memory cells rather than aborting the sweep. The four kinds are the full-frame RGB baseline at resolution N, the base flow at M, a full-frame residual U-Net at N (not scalable, included to show that the tile-locality, not the residual-modeling idea, is what saves memory), and

the tiled residual flow (the proposed model on $(T + 2h)\times(T + 2h)$ inputs). Table 12 lists selected cells of a 114-configuration grid; the floors of Table 1 and the values quoted below come from separate runs of the same protocol.

Table 12: Peak GPU memory on an H200 (selected microbenchmark configurations), at N=512 and batch 8 unless marked. Values from the 114-configuration grid; the last row sums two separately measured peaks. The full-frame residual U-Net matches the full-frame RGB peak memory at the same setting, indicating that residual modeling alone does not save memory; the tile-locality of the residual is what does. A batch of 8 counts 8 images for the full-frame entries and 8 tiles for the tiled ones, half an image at T=128 and two images at T=256. They are therefore not comparable to each other across model kinds; the resolution-scaling statement is Table 1, which compares one image against one tile.

configuration	peak memory
RGB full-frame, C=64, B=8	27.55 GiB
RGB full-frame, C=96, B=8	43.11 GiB
RGB full-frame, N=1024, C=64, B=1	13.82 GiB
RGB full-frame, C=64, B=4	13.82 GiB
Full-frame residual U-Net, C=64, B=4	13.85 GiB
Base flow, M=128, C=64, B=8 (smallest)	1.80 GiB
Base flow, M=256, C=64, B=8	6.95 GiB
Tiled residual, T=128, h=16, C=32, B=8 (smallest)	1.45 GiB
Tiled residual, T=128, h=16, C=48, B=8	2.12 GiB
Tiled residual, T=256, h=32, C=48, B=8	8.19 GiB
Base M=128 + tiled residual (T=128, h=16, C=32), B=8 each	3.25 GiB

Optimizer state. Each timed step applies a stateless update (SGD with zero learning rate), so the floors in Table 1 exclude optimizer state. Adam's two moment buffers would add 0.02 GiB to the tiled floor and 0.04 GiB to the full-frame one, giving ratios of 10.2×, 39.6× and 157.3× at 512, 1024 and 2048 instead of 10.7×, 41.9× and 166.9×.

Five-level full-frame model. The five-level baseline of Section 5.2 (7.40 M parameters) needs 3.63, 14.21 and 56.56 GiB for one image at 512, 1024 and 2048, about 3% above the two-level C=64 model, giving ratios of 11.0×, 43.1× and 171.6× against one tile (10.5×, 40.7× and 161.7× with Adam state). At its training batch of 24 it peaks at 84.79 GiB, 3.53 GiB per image, and that batch does not fit on the H200 at 1024 or 2048; the tiled residual flow at its training batch of 256 tiles peaks at 64.90 GiB, 4.06 GiB per 16 tiles, one image of area.

Where the out-of-memory regime begins. A full-frame RGB C=64 step at N=1024 and batch 1 fits in 13.82 GiB, so at 1024 tiling adds memory headroom but is not needed. The regime where tiling is required begins at 2048: a full-frame step needs 55.02 GiB for one image, beyond a 40 GB accelerator, and 109.96 GiB at batch 2, beyond an 80 GB one (Section 4.1).

I AN ORACLE FOR THE BASE STAGE

Pushing real images through our own decomposition and keeping only the base reconstruction gives an oracle base: the output of a base flow that reproduced the real base distribution exactly. It scores FID 83.98 at M=128, 137.04 at M=64 and 224.94 at M=32, against a real-versus-real FID of 9.53. This is a control, not a bound: a base flow that departs from the real base distribution could in principle score lower. Even the M=32 oracle scores better than every two-level full-frame baseline, whose best checkpoint is 261.88, so those baselines fail on global structure; the five-level baseline, at 57.8 over its best checkpoints, is not limited this way. Our base-only control at M=128 (127.32) sits 43 FID above its oracle, so the base flow, not the lowpass, limits it. At M=64 the tiled model (56.8 ± 2.2) scores below that resolution's oracle, so the residual adds detail beyond an exact base at that bandwidth, whereas at M=128 (104.1 ± 7.3) it does not.

J A SECOND DATASET: AFHQ AT 512

On AFHQ (Choi et al., 2020) at 512, all three classes, trained on 14,000 of the 14,630 training images and evaluated with $n{=}1500$ samples against the release's 1,500-image test set (500 per class) at sampling seed 123, the tiled model at $M{=}128$ reaches FID 119.97 against 217.45 for the two-level RGB $C{=}64$ model, the same ordering as for those two configurations on FFHQ; we did not train a five-level baseline on AFHQ. The native-detail metrics run the other way: edge_{512} is 0.3614 against 0.1175 and the seam-gradient jump is positive. The comparison is not parameter-matched (7.64 M against 4.89 M), the tiled model pairs a 100k-step residual flow with a 50k-step base against 100k steps for the baseline, and $n{=}1500$ makes these values comparable within AFHQ only.

K SAMPLING PROCEDURE

Algorithm 1 gives the sampler. The base flow models $2B_M - 1$ and the residual flow models $(A_N - \mu_A)/\sigma_A$, where μ_A and σ_A are the scalar mean and standard deviation of A_N over the training images. The base and residual flows start from Gaussian noise and take n_{base} and n_{res} Euler steps, 50 each unless stated otherwise. Tiles do not interact before the stitch, so in practice the tile loop is batched, K effective tiles per call to the residual flow (Section 3.5).

Algorithm 1 Sampling at resolution N with base M, tile T, halo h

$\hat{B}_M \leftarrow \text{clamp}\big((1 + \text{euler}(\text{base_flow}; n_{\text{base}}))/2, 0, 1\big)$ ▷ from $M{\times}M$ noise
$\hat{B}_N \leftarrow (\text{us}_2)^k(\hat{B}_M)$
for each tile $i = 1, \ldots, (N/T)^2$ **do**
  $\hat{B}^{(i)}_{\text{ctx}} \leftarrow \text{extract_effective}(\hat{B}_N, i)$ ▷ reflection padding at the border
  $\text{pos}^{(i)} \leftarrow (c_y^{(i)}/N, c_x^{(i)}/N, T/N)$
  $\hat{A}^{(i)} \leftarrow \mu_A + \sigma_A \,\text{euler}(\text{residual_flow} \mid \hat{B}^{(i)}_{\text{ctx}}, \text{pos}^{(i)}; n_{\text{res}})$ ▷ own noise per tile
end for
$\hat{A}_N \leftarrow \text{stitch_windowed}(\{\hat{A}^{(i)}\}, \text{Hann})$
return $\text{clamp}(\hat{B}_N + \hat{A}_N, 0, 1)$

L SEED-AGGREGATED METRICS

Table 13 lists nine metrics over the three retrains of the two-level $C{=}64$ baseline and the tiled model at $M{=}128$; on all but native-512 edge the means differ by more than one pooled standard deviation.

Table 13: Nine metrics at 350k over three training seeds (123, 456, 789); each tiled seed retrains both stages. Separation is the absolute difference of the means in pooled standard deviations. Arrows give the direction of improvement. The baseline here is the two-level $C{=}64$ model (4.89 M parameters), not the five-level baseline of Section 5.2, and the tiled model uses $M{=}128$.

metric	tiled $M{=}128$	two-level $C{=}64$	separation
FID ↓	104.1 ± 7.3	298.9 ± 25.7	10.3
KID ↓	0.109 ± 0.014	0.381 ± 0.039	9.2
div ↑	6.70 ± 0.21	4.79 ± 0.33	7.0
color_{512} ↓	0.67 ± 0.22	1.52 ± 0.17	4.4
edge_{64} ↓	0.162 ± 0.033	0.282 ± 0.022	4.3
color_{64} ↓	0.97 ± 0.13	2.13 ± 0.38	4.1
radial_{512} ↓	0.0015 ± 0.0012	0.0060 ± 0.0020	2.7
lowres_{512} ↓	0.021 ± 0.011	0.105 ± 0.074	1.6
edge_{512} ↓	0.308 ± 0.120	0.330 ± 0.147	0.2

Table 14 gives the same nine metrics for the comparison of Table 2: the tiled model at $M{=}64$ over three retrains of both stages against the five-level baseline over two seeds, at 350k and at each seed's best checkpoint. With two baseline seeds we report means and standard deviations only.

Table 14: Nine metrics for the tiled model at $M{=}64$ (350k, seeds 123, 456, 789) and the five-level baseline (seeds 123, 456), at 350k and at each seed's best checkpoint by FID (300k for seed 123, 350k for seed 456). Arrows give the direction of improvement.

metric	tiled $M{=}64$	five-level, 350k	five-level, best
FID ↓	56.8 ± 2.2	66.6 ± 2.5	57.8 ± 9.9
KID ↓	0.041 ± 0.002	0.054 ± 0.001	0.046 ± 0.013
div ↑	7.65 ± 0.09	5.04 ± 0.33	6.18 ± 1.29
color_{512} ↓	0.28 ± 0.05	1.27 ± 0.10	0.89 ± 0.45
edge_{64} ↓	0.070 ± 0.011	0.171 ± 0.001	0.132 ± 0.053
color_{64} ↓	0.52 ± 0.12	1.79 ± 0.29	1.15 ± 0.62
radial_{512} ↓	0.0036 ± 0.0008	0.0066 ± 0.0014	0.0037 ± 0.0027
lowres_{512} ↓	0.006 ± 0.003	0.071 ± 0.052	0.031 ± 0.004
edge_{512} ↓	0.396 ± 0.010	0.369 ± 0.065	0.344 ± 0.030

Table 15 lists the tiled model per seed at 350k and the base flows evaluated alone. On diversity, low-resolution MMD and color at both resolutions the $M{=}64$ bases of seeds 123 and 456 alone already beat every five-level checkpoint (Table 8); downsampled edge needs the residual (0.179 and 0.116 for these bases alone, 0.095 for the best checkpoint, 0.060 to 0.082 for the tiled model).

Table 15: Per-seed values of the tiled model at 350k and of its 100k-step base flow alone, $n{=}3500$; the $M{=}64$ base of seed 789 was not evaluated alone. Lower is better except for div.

configuration	FID	KID	div	edge_{64}	edge_{512}	lowres_{512}	color_{512}	color_{64}	radial_{512}
tiled $M{=}64$, seed 123	59.15	0.041	7.59	0.082	0.407	0.0044	0.34	0.51	0.0039
tiled $M{=}64$, seed 456	56.33	0.043	7.76	0.060	0.389	0.0038	0.24	0.41	0.0028
tiled $M{=}64$, seed 789	54.89	0.039	7.61	0.069	0.391	0.0090	0.26	0.65	0.0042
base alone $M{=}64$, seed 123	165.19	0.192	7.52	0.179	1.071	0.0052	0.44	0.62	0.0086
base alone $M{=}64$, seed 456	157.10	0.183	7.68	0.116	1.021	0.0045	0.37	0.54	0.0081
tiled $M{=}128$, seed 123	112.56	0.125	6.90	0.166	0.211	0.0145	0.46	0.82	0.0004
tiled $M{=}128$, seed 456	99.87	0.102	6.70	0.127	0.272	0.0145	0.64	0.99	0.0014
tiled $M{=}128$, seed 789	99.81	0.099	6.49	0.192	0.442	0.0331	0.90	1.09	0.0028
base alone $M{=}128$, seed 123	127.32	0.136	6.88	0.173	0.744	0.0149	0.56	0.88	0.0041
base alone $M{=}128$, seed 456	123.15	0.130	6.68	0.142	0.788	0.0147	0.72	1.04	0.0047
base alone $M{=}128$, seed 789	128.15	0.133	6.48	0.200	0.856	0.0332	0.95	1.13	0.0055

Table 16 gives the two-level rows of Table 2 per seed.

Table 16: Per-seed values of the two-level baselines of Table 2, $n{=}3500$: $C{=}64$ at 350k and $C{=}80$ at its best checkpoint by FID. Lower is better except for div.

configuration	FID	KID	div	edge_{64}	edge_{512}	lowres_{512}	color_{512}	color_{64}	radial_{512}
two-level $C{=}64$, seed 123, 350k	322.04	0.417	4.52	0.302	0.470	0.1905	1.66	2.52	0.0082
two-level $C{=}64$, seed 456, 350k	303.44	0.387	4.69	0.258	0.342	0.0667	1.57	2.10	0.0054
two-level $C{=}64$, seed 789, 350k	271.17	0.339	5.15	0.287	0.177	0.0578	1.34	1.76	0.0043
two-level $C{=}80$, seed 123, 300k	261.88	0.327	5.58	0.250	0.188	0.0777	1.23	1.59	0.0002

M SEAM METRICS AT THE FINAL CHECKPOINT

Table 17 reports the seam metrics per training seed for both base resolutions. Boundary L_1 is the mean absolute difference between the two pixel rows, or columns, on either side of each interior central-tile boundary. The seam-gradient jump subtracts from it the mean absolute difference of the adjacent pixel pairs just inside each tile, so it is positive when a boundary is rougher than its surroundings. The tile-boundary edge excess subtracts the boundary L_1 of real validation images on the same grid.

Table 17: Seam metrics of the tiled model at the 350k checkpoint, three seeds at each base resolution, on the n=3500 samples of Table 2. We do not read the negative tile-boundary edge excess as evidence of absent seams: central stitching at M=128, with a seam-gradient jump of $+307\times10^{-5}$, still has an edge excess of -0.0015, and a variance deficit in the overlap band would also lower it. Appendix N tests for that deficit at M=128.

metric	M=64, seeds 123, 456, 789	M=128, seeds 123, 456, 789
boundary L_1	0.0108, 0.0102, 0.0104	0.0128, 0.0109, 0.0090
seam-gradient jump	-4.2, -4.9, -0.4 ($\times10^{-5}$)	-15.5, -7.8, -7.1 ($\times10^{-5}$)
tile-boundary edge excess	-0.0088, -0.0094, -0.0092	-0.0068, -0.0087, -0.0106

N Overlap-band variance

Averaging independently sampled overlapping tiles would shrink variance in the overlap band, which would be a distributional artifact rather than a seam. We measure it for the tiled model at M=128. If the tiles were independent, our window would scale the per-pixel standard deviation by $\sqrt{\sum w^2}/\sum w$, which averages 0.871 over the 34% of pixels inside an overlap and falls to 0.502 where four tiles meet. For the statistic we measure, the standard deviation in columns within h of a vertical seam relative to columns at least $2h$ away, the prediction is 0.878. Measured over 128 samples, that ratio is 1.011, so the pooled statistic shows no deficit.

Independence does not hold. The residual flow is conditioned on a crop of the shared upsampled base, so overlapping tiles receive nearly the same context. Measured Pearson correlation between the two tiles sharing a boundary, over 192 pairs, is 0.807, which lifts the equal-weight prediction from 0.707 to 0.951. That is still below the measured 1.011, so correlation explains the direction but not all of the margin. We measured one geometry, and the standard deviation is pooled across samples carrying different base images, which would dilute a conditional deficit.

O Fusion rule and noise sharing

At M=128 (seed 123, 350k steps) we cross when the tiles are fused, once after the last solver step or after every step, with how the initial residual noise is drawn, independently per tile or as crops of one $N\times N$ field shared by overlapping tiles. Per-step fusion with shared noise is the rule of MultiDiffusion (Bar-Tal et al., 2023). Table 18 lists the four cells. Their FIDs span 1.77, below the 10 FID threshold of Section 5.3. Per-step fusion with independent noise has the lowest boundary L_1, but its first fusion averages independently drawn noise. Measured as in Appendix N, its ratio of overlap-band to interior standard deviation is 0.975, against 1.011 for the default sampler, so its lower L_1 is not evidence of a cleaner seam.

Table 18: Fusion rule crossed with noise sharing at M=128, seed 123, 350k steps, n=3500. The first row is the default sampler.

fusion	initial noise	FID	boundary L_1
once, after the last step	per tile	112.56	0.0128
once, after the last step	shared field	113.53	0.0154
every step	shared field (MultiDiffusion)	113.45	0.0153
every step	per tile	111.76	0.0102

P Matched-memory training throughput

We complement the memory floor of Table 1 with a matched-memory throughput sweep. At each memory band we compare how many image-equivalents per second the two models train on an H200 under bf16 autocast, `torch.compile(mode="reduce-overhead")` and AdamW, timing the forward, backward and optimizer steps on synthetic inputs. Each configuration runs three times, each run in its own process so that compilation sees a single batch size, with 5 warm-up and 20

timed steps. Peak memory is from the second and third runs, which reuse cached compiled kernels; the first run, which compiles them, peaks 5.5 to 7.4% higher. For the tile path, an image-equivalent is 16 tiles (at N=512, T=128), so a tile entry of B=128 means a batch of 128 tiles, equivalent to 8 full images.

Table 19: Matched-memory throughput at N=512 on an H200 (bf16 + compile). Throughput is given in image-equivalents per second, the median of three runs. Each cell shows the batch size and the throughput at that operating point. Peak memory is 9.46, 18.86, 37.65 and 91.67 GiB for the full-frame cells and 11.20, 22.35, 44.65 and 89.26 GiB for the tiled cells.

memory band	full-frame, C=64	tiled, C=48	full-frame faster by
∼10 GiB	B=8: 125.5	B=128: 113.7	10%
∼20 GiB	B=16: 130.0	B=256: 113.6	14%
∼40 GiB	B=32: 130.9	B=512: 113.8	15%
∼90 GiB	B=78: 129.2	B=1024: 98.8	31%

Reading. The full-frame C=64 model trains at 125.5 to 162.2 images per second at every batch size we measured, from B=8 to 78, and is fastest at B=48 and 64. The tiled C=48 model trains at 113.6 to 113.8 image-equivalents per second from B=128 to 512 and at 98.8 at B=1024, so larger batches do not raise its throughput. Part of the tile cost is halo work: each 160×160 input covers 1.56 times the pixels of its 128×128 tile.

At 512 on an H200 the systems advantage is the per-step memory floor, not training throughput. Fewer residual solver steps, sparse tile routing or hierarchical residuals might lower sampling cost; Appendix Q tests each on the pilot checkpoint.

Q Efficiency levers tested on the pilot checkpoint

Using the 50k pilot residual flow with the 100k-step base flow at M=128, on 64 FFHQ-512 validation images, we tested four ways to reduce sampling cost. Cutting residual solver steps from 50 to 30, with the base solver at 50, gave a 1.59× speedup at 7.5% relative edge degradation, and 20 steps gave 2.27× at 19.0%. Caching base-conditioning features made a synthetic training step 5 to 16% slower, depending on the encoder; its effect on sampling was not measured. Routing only the half of the tiles with the most residual energy would halve the residual pass, but a practical router based on base-image gradients selected tiles too noisily to approach that oracle. A hierarchical residual pyramid passed a latency and energy feasibility check, and its quality is untested. Appendix R repeats the solver sweep at the 350k checkpoint.

R Residual solver steps and evaluation noise

Table 20 sweeps the residual solver of the tiled model at M=128 (seed 123, 350k) with the base solver held at 50 Euler steps. FID improves slightly as residual steps are removed while native edge quality degrades, edge_{512} rising 24% from NFE=50 to NFE=10. FID resizes images to 299×299 before the Inception network and so underweights the native detail the residual produces; fewer steps most likely give smoother output that FID rewards. We therefore keep NFE=50 as the quality reference and do not read the curve as a free speedup. A sweep on the pilot checkpoint, with the other efficiency levers we tested, is in Appendix Q.

Table 20: Residual solver steps at the 350k checkpoint, n=3500, with the base solver fixed at 50 Euler steps. Time is amortized sampling wall-clock per image. NFE=50 and 30 were timed on an H100 in the evaluation behind the M=128 row of Table 2, NFE=20 and 10 on an H200, where the same checkpoint took 0.532 s at NFE=50 in a separate evaluation; each speedup is computed on one GPU type.

residual NFE	FID	edge$_{512}$	boundary L_1	time (s/img)	GPU	speedup
50	112.56	0.211	0.0128	0.596	H100	1.00×
30	112.23	0.222	0.0124	0.373	H100	1.60×
20	111.81	0.234	0.0120	0.237	H200	2.25×
10	110.62	0.261	0.0112	0.138	H200	3.86×

To check whether the NFE=30 against NFE=50 gap exceeds evaluation noise, a 200-replicate paired bootstrap resamples the 3,500 generated samples with replacement, applying one index set to both settings, against a fixed reference pool. The point difference is -0.330 FID; the bootstrap standard deviation of the difference is 0.031 and its 95% interval $[-0.396, -0.274]$ does not cross zero. The paired difference is resolved this finely because most sampling noise is shared by the two settings; a single evaluation's own bootstrap spread is about 1.1 FID. These intervals describe sampling variability against one fixed reference; they are not confidence intervals on population FID. Both spreads are well below the three-seed standard deviation of 7.34 FID at M=128 and below the 2.17 at M=64, so retraining, not evaluation, is the main source of uncertainty.

S Implementation notes for the 512 comparison

The residual flow trains on crops of the real upsampled base B_N and never loads the base flow, so the two stages train independently and meet only at sampling, where the residual sees crops of the generated base; we use no conditioning augmentation. Training uses bf16 autocast and precomputed pinned tile pools with asynchronous host-to-device transfers. The full-frame and base trainers are compiled with `torch.compile`; the residual flows trained without compilation, averaging 381 ms per step at N=512, T=128, C=48 over the 350k-step run at M=128. The seed-789 base at M=64 also trained without compilation (1.82 against 0.74 H200-hours for the compiled seed-123 base), as did the two-level C=96 run. The pipeline is documented for reproducibility and is not a contribution of the method.

T QUALITATIVE COMPARISON AT 512

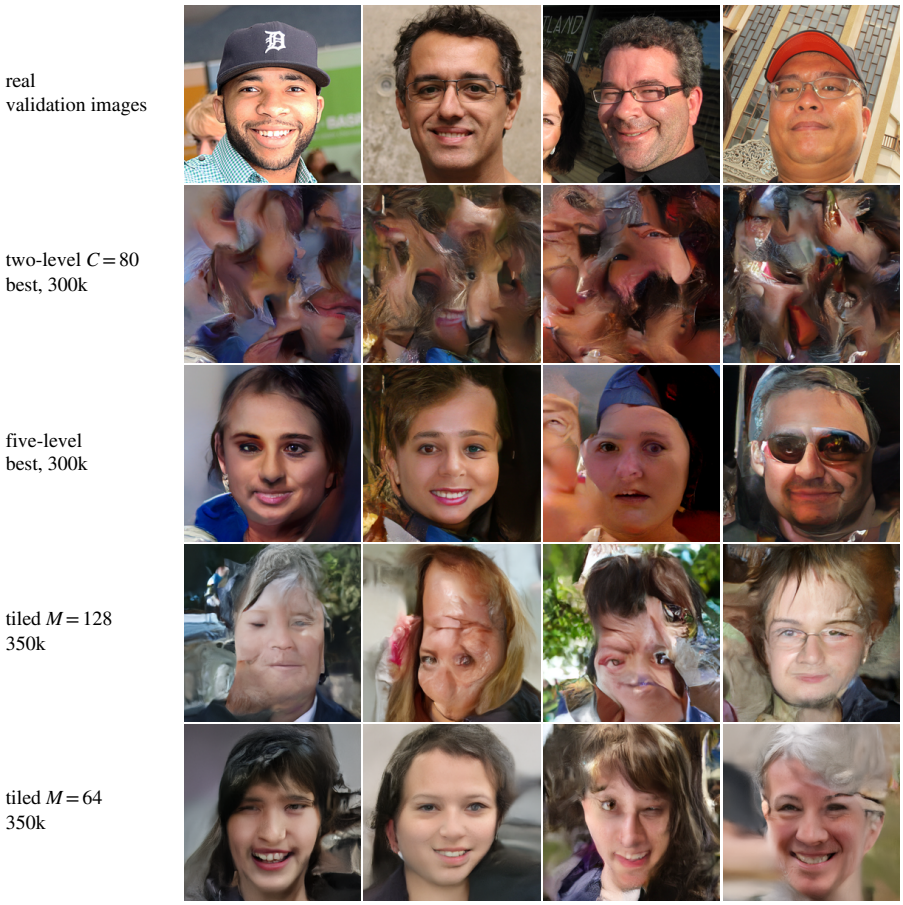


Figure 3: Samples at 512, the first four of the n=3500 used for the metrics of Table 2, uncurated, all at sampling seed 123 and training seed 123. Rows: the first four real validation images; the two-level C=80 baseline and the five-level baseline, each at its best checkpoint (300k); the tiled model at M=128 and at M=64, both at 350k. The generated rows score FID 261.9, 50.8, 112.6 and 59.1. The two full-frame models start from the same 512×512 noise, so their columns correspond; the tiled models start from $M \times M$ base noise and per-tile residual noise, so their columns do not correspond to the other rows.

Figure 4 shows failure cases of the tiled model at M=128 and training seed 123, the run with the highest boundary L_1 in Table 17. They are selected by that metric rather than curated: the 16 samples with the highest boundary L_1 out of 96 uncurated samples from the 350k checkpoint (mean 0.0131, standard deviation 0.0031, maximum 0.0237), and a close-up of the tile boundaries of the worst one. The failures are distorted faces and texture, consistent with an FID of 112.56; we see no discontinuity at tile boundaries.



Figure 4: Worst cases by boundary L_1 at $M{=}128$, 350k checkpoint, training and sampling seed 123. Left: the 16 of 96 uncurated samples with the highest boundary L_1, from 0.0237 to 0.0163, in decreasing order row by row. Right: $96{\times}96$ crops of the worst sample centered on the nine points where four tiles meet; ticks outside each crop mark the tile boundaries.

U LIMITATIONS: EXTENDED NOTES

Full $1024{\times}1024$ training is future work. At 1024 a full-frame step still fits on an H200 (Appendix H), so the regime where tiling is required is not reached at that size. A 50k-step pilot at 1024 on 5,000 images, with $M{=}256$, $T{=}256$, $h{=}32$ and a $C{=}32$ residual flow, evaluated under the same protocol, gives FID 225.67 for the tiled model against 215.97 for the base alone, so at that budget the residual flow does not help; whether this is a budget effect or a property of the geometry at that scale we do not know.

Duplicate-rate metric. Our evaluation code also computes a duplicate rate: the fraction of samples whose nearest training image at $16{\times}16$ lies closer than the first percentile of distances between random pairs of training images. Among 30,000 training images almost every face has such a neighbor, and 92% of the held-out validation images are flagged. We therefore do not use this metric; Appendix E compares nearest-neighbor distances against a real-image reference instead.

No text or class conditioning. The present recipe trains an unconditional generator. Conditioning could be added at the base flow, the residual flow, or both, but is left to future work. We do not claim text-to-image performance.

Tiled residual generation has local-context limitations. The shared residual evaluator sees only a $(T{+}2h){\times}(T{+}2h)$ context plus a normalized position vector. Long-range coherence in the residual must therefore come from the upsampled base B_N that the tile is conditioned on. If the base does not encode a particular long-range structure, the residual cannot recover it from local information alone. We have not measured long-range coherence directly; a dataset with strong long-range structure (for example urban, indoor, multi-object or text-bearing images) might expose this limitation.

Sampling time is not optimized. The tiled model at $M{=}128$ takes 0.596 s per image at NFE=50 and 0.373 s at NFE=30 on an H100, amortized over the $n{=}3500$ evaluation; at $M{=}64$ it takes 0.50

to 0.51 s at NFE$=50$ on an H200. Distillation, fewer-step solvers and batching tiles across images could lower this further and are out of scope.

Medical, safety, and other downstream applications. The exact-transform formulation and its deterministic reconstruction have adjacent applications (e.g. medical imaging, where a learned-decoder dependence is undesirable). We do not include those experiments here and do not claim downstream task performance. The paper is a generative modeling result.

V LICENSES FOR EXISTING ASSETS

This section lists the datasets and pretrained weights used in this paper with their licenses or terms of use, taken from each project's license file or, where there is none, from its project page. All assets are used within their stated terms; none is redistributed commercially.

Table 21: Licenses and terms of use for existing assets: datasets used for training and evaluation, and pretrained weights used as feature extractors for FID and KID.

Asset	License or terms of use	URL
Datasets		
FFHQ (Karras et al., 2019), used at 64, 512 and 1024	The dataset (metadata, download script and documentation) is released by NVIDIA under CC BY-NC-SA 4.0; each image keeps its Flickr license (CC BY 2.0, CC BY-NC 2.0, Public Domain Mark 1.0, CC0 1.0 or U.S. Government Works), all of which allow non-commercial use	`https://github.com/NVlabs/ffhq-dataset`
AFHQ (Choi et al., 2020)	Creative Commons BY-NC 4.0, NAVER Corporation	`https://github.com/clovaai/stargan-v2`
Pretrained model weights		
Inception-v3 (IMAGENET1K_V1 weights) for FID and KID features (Szegedy et al., 2016)	The torchvision code is BSD-3-Clause; torchvision states that its pretrained weights may carry terms from their training data. Used only as a frozen feature extractor	`https://github.com/pytorch/vision`

The real images in Figures 1 and 3 are FFHQ images 46951, 34069, 49282 and 24342, resized from 1024 to 512; the FFHQ metadata lists the Flickr author and license of each. No third-party checkpoints or code are redistributed with this submission.