# Exploring Diffusion Transformer Designs via Grafting

**Keshigeyan Chandrasegaran** [*1 2] **Michael Poli** [*1 2] **Daniel Y Fu** [3 4] **Dongjun Kim** [1]
**Lea M. Hadzic** [1] **Manling Li** [1 5] **Stefano Massaroli** [2] **Agrim Gupta** [6]
**Azalia Mirhoseini** [1] **Juan Carlos Niebles** [1 7 †] **Stefano Ermon** [1 †] **Li Fei-Fei** [1 †]

## Abstract

Model architecture design requires decisions such as selecting operators (e.g., attention, convolution) and configurations (e.g., depth, width). However, evaluating the impact of these decisions on model quality requires costly pretraining, limiting architectural investigation. Inspired by how new software is built on existing code, we ask: can new architecture designs be studied using pretrained models? We present *grafting*, a simple approach for editing pretrained diffusion transformers (DiTs) to materialize new architectures under small compute budgets. We study the impact of grafting on model quality using the DiT-XL/2 design. We develop a family of hybrid designs via grafting: replacing softmax attention with gated convolution, local, and linear attention; and MLPs with variable-width and convolutional variants. Notably, many hybrid designs achieve good quality (FID: 2.38–2.64 vs. 2.27 for DiT-XL/2) using $< 2\%$ pretraining compute. Next, we graft a text-to-image model (PixArt-$\Sigma$), achieving a 43% speedup with $< 2\%$ drop in GenEval score. Finally, we present a case study where we restructure DiT-XL/2 by converting every pair of sequential transformer blocks into parallel blocks via grafting, reducing model depth by 2x, achieving better quality (FID: 2.77) than models of comparable depth. Together, we show that new diffusion model designs can be explored by grafting pretrained DiTs, with edits ranging from operator replacement to architecture restructuring. Code and grafted models: grafting.stanford.edu.

[*]Equal contribution [†] Equal senior authorship [1]Stanford University [2]Liquid AI [3]UC San Diego [4]Together AI [5]Northwestern University [6]Google DeepMind [7]Salesforce Research. Correspondence to: <keshik@stanford.edu>, <poli@stanford.edu>.



## 1. Introduction

Model architecture design plays a central role in machine learning, alongside data, algorithms, systems, compute, and benchmarks. It entails several key decisions, including the choice of computational operators (e.g., attention, convolution) and structural configurations (e.g., model depth, width). Despite this, insight into architectures—what works and what doesn't—is difficult to obtain due to the prohibitive costs of training models from scratch, especially in today's foundation model era. As a result, studying new architectures remains a challenge, particularly for generative models. Much like how new software is built on existing code rather than written from scratch, we ask whether pretrained models can serve as scaffolds for studying new architectures. In this work, *we investigate architectural editing of pretrained models to study new architecture designs.* We focus on diffusion transformers (DiTs), a class of generative transformers widely used for image and video generation (Peebles & Xie, 2023; Brooks et al., 2024; Gupta et al., 2023).

A pretrained model implements a computational graph to perform tasks such as image or video generation. Given a new architectural idea and a pretrained model, we investigate whether the idea can be materialized by modifying the pretrained model's computational graph under small compute budgets. For example, one might hypothesize that a convolutional design could replace Multi-Head Attention (MHA) or Multi-Layer Perceptron (MLP) in a DiT. A simple way to materialize this idea is to replace MHA or MLP operators with a convolutional operator, while preserving model functionality. This raises two key questions: (Q1) *operator initialization*: How should a new operator be initialized before being integrated into the graph? (Q2) *error accumulation*: How can we mitigate error propagation as multiple operators are integrated into the graph?

To address these questions, we present **grafting**, a simple two-stage approach to architecture editing (Fig. 1). Grafting proceeds as follows: (i) *operator initialization via activation distillation*: This stage performs activation distillation, using regression to transfer the functionality of the original operator to the new operator. (ii) *lightweight finetuning*: This stage mitigates error propagation due to integrating

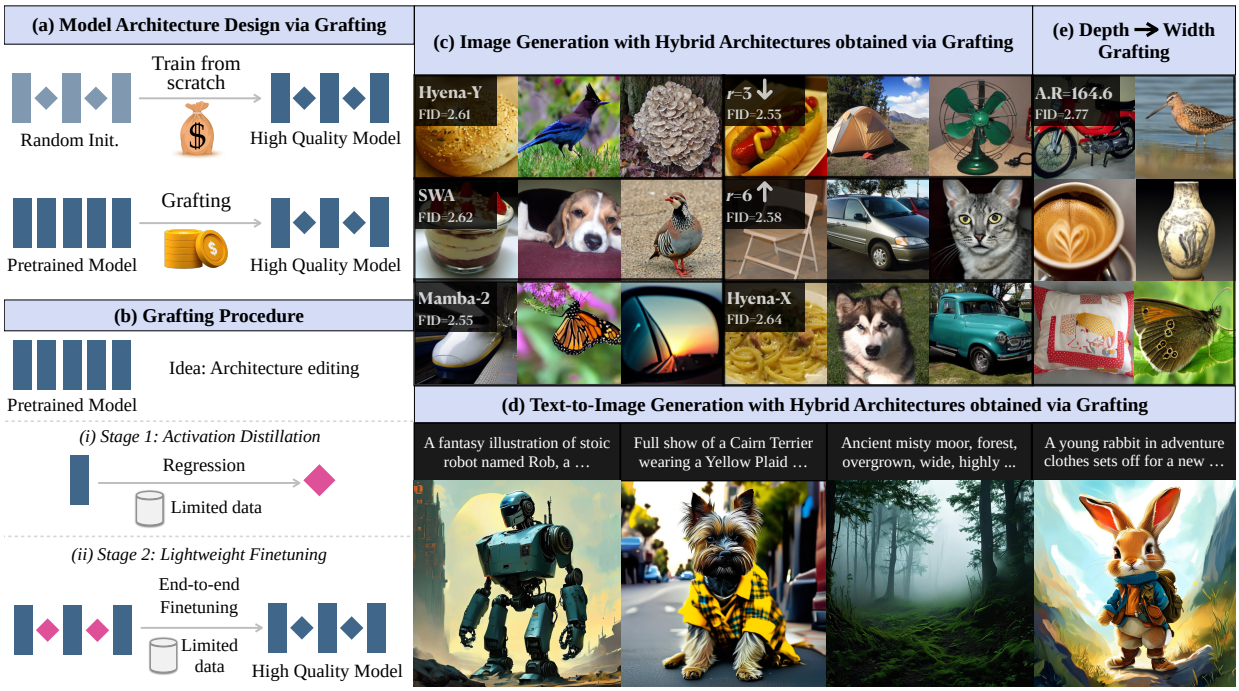


*Figure 1.* **Grafting overview.** *(a,b) Model architecture design via grafting.* Studying new model architecture designs requires costly pretraining. Grafting materializes new architectures by editing pretrained models under small compute budgets (Sec. 3). *(c) Class-conditional image generation.* Samples generated by hybrid architectures obtained via grafting (Sec. 4). *(d) High-resolution text-to-image generation.* Samples generated using a grafted PixArt-$\Sigma$ model (Sec. 5). *(e) Depth → width case study.* Samples generated using a model restructured via grafting (depth: 28 → 14) (Sec. 6).

multiple new operators by finetuning using limited data. We validate our grafting approach in increasingly challenging generative modeling setups:

**Result I: Grafting renders hybrid architectures with competitive quality for class-conditional image generation (Sec. 4).** For softmax attention, we explore several interleaved designs: local gated convolution (Hyena-SE, and our proposed Hyena-X/ Hyena-Y), local attention (sliding window), and linear attention (Mamba-2). For MLPs, we explore several interleaved designs: variable-width MLPs (width ratios=3, 6), and a convolutional variant (Hyena-X). Interestingly, FID scores across these hybrids range from 2.38 to 2.64 (DiT-XL/2 256x256: 2.27), showing that grafting can construct high-quality hybrids. Grafting is lightweight: each experiment completes in under 24 hours on 8×H100 GPUs, using $<$2% of pretraining compute.

**Result II: We construct efficient hybrid architectures for high-resolution text-to-image generation via grafting (Sec. 5).** We validate grafting in a challenging real-world setting: Text-to-image generation (2048x2048) using PixArt-$\Sigma$. This setting reflects key real-world challenges: it operates on long sequences (16,384 tokens), involves a multimodal setup with text conditioning, and lacks training data. We target self-attention layers for grafting, as they account for over 62% of generation latency and use 12k synthetic data. The grafted model achieves a 1.43× speedup with less than a 2% drop in GenEval score (47.78 vs. 49.75). These results show that grafting is effective for high-resolution, text-to-image DiTs and works under realistic constraints.

**Case Study: Converting model depth to width via grafting (Sec. 6).** Motivated by our MLP grafting results, we restructure DiT-XL/2 to trade depth for width by parallelizing pairs of transformer blocks, as modern GPUs favor parallel over sequential computation. This reduces model depth by 2× (28→14). The resulting model achieves FID=2.77, outperforming other models of comparable depth. To our knowledge, this is the *first attempt* to convert sequential transformer blocks into parallel in DiTs.

## 2. Prerequisites

**Diffusion models (DMs).** DMs generate data samples by iteratively denoising random noise. This sampling process inversely mirrors the forward data corruption mechanism: $\mathbf{z}_t = \alpha_t \mathbf{z} + \sigma_t \boldsymbol{\epsilon}$ where $\mathbf{z} = E(\mathbf{x}) \sim q(\mathbf{z})$ with $E$ representing a pretrained encoder and $\mathbf{x}$ the data variable. The noise term $\boldsymbol{\epsilon}$ follows the prior distribution $\mathcal{N}(0, I)$. The transition kernel from time 0 to $t$ is given by $q_t(\mathbf{z}_t|\mathbf{z}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{z}, \sigma_t^2 I)$. The choice of $\alpha_t$ and $\sigma_t$ defines the diffusion variant, including variance-preserving (Ho et al., 2020), or flow matching (Lipman et al., 2022). The training objective (Ho et al., 2020) is as follows:

$$\mathcal{L}_{DM}(\boldsymbol{\phi}) = \mathbb{E}_{q(t)q(\mathbf{z},\mathbf{c})\mathcal{N}(\boldsymbol{\epsilon};0,I)}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{z}_t, t, \mathbf{c})\|_2^2], \quad (1)$$

where $q(t)$ is the time sampling distribution, and $q(\mathbf{z}, \mathbf{c})$ is the joint distribution of latent $\mathbf{z}$ and condition $\mathbf{c}$.

**Diffusion transformers (DiTs).** DiTs model the diffusion process by patchifying the input—noised images or latent—into a sequence of 1D tokens with positional embeddings. These tokens are processed through transformer blocks comprising self-attention, feedforward layers, residual connections, and normalization layers. DiTs also incorporate conditioning signals, such as noise timestep ($t$), class labels ($c$), or natural language prompts, enabling controllable generation (Peebles & Xie, 2023; Chen et al., 2023).

## 3. Grafting Diffusion Transformers

### 3.1. Two-Stage Grafting Approach

Grafting aims to materialize new architectures by editing a pretrained model's computational graph, replacing existing operators with alternatives. This raises two core questions:

*(Q1) How should a new operator be initialized before being integrated into the computational graph?* **Stage 1: Activation distillation.** We cast initialization as a supervised regression task. Operators in a DiT block process $[B, N, D]$ inputs (batch, sequence, hidden) and output tensors of the same shape. Given a pretrained operator $f_\phi^l$ at layer $l$, we learn a new operator $g_\theta^l$ that approximates $f_\phi^l$ (Hinton et al., 2015). Since DiT activations are continuous and smooth, this can be posed as a regression problem:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{q(t)q(\mathbf{z},\mathbf{c})q_t(\mathbf{z}_t|\mathbf{z})}\left[\mathcal{L}_{\text{reg}}(g_{\boldsymbol{\theta}}^l(\mathbf{z}_t, t, \mathbf{c}), f_{\boldsymbol{\phi}}^l(\mathbf{z}_t, t, \mathbf{c}))\right] \quad (2)$$

where $q(\mathbf{z}, \mathbf{c})$ is the joint distribution of latent representation $\mathbf{z}$ and condition $\mathbf{c}$, $q(t)$ is the time sampling distribution, and $q_t(\mathbf{z}_t|\mathbf{z})$ is the transition kernel from time 0 to $t$. $\mathcal{L}_{reg}$ is a regression objective such as $L_2$. In practice, a good initialization requires as few as 8k samples.

*(Q2) How can we mitigate error propagation as multiple operators are integrated into the computational graph?* **Stage 2: Lightweight finetuning.** As more operators are replaced, initialization errors propagate, leading to deviations from the pretrained model's behavior. We apply end-to-end finetuning with limited data to mitigate cumulative errors from Stage 1. The fine-tuning objective is given in Equation 1. In practice, we find that competitive performance can be recovered using only 10% of the pretraining data.

**Self-grafting baseline.** As a control, self-grafting replaces each operator with a randomly initialized version, preserving the architecture while isolating the effect of grafting.

| | Ratio | IS ↑ | FID ↓ | sFID ↓ | Prec. ↑ | Rec. ↑ |
|---|---|---|---|---|---|---|
| Baseline | – | 278.20 | 2.27 | 4.60 | 0.83 | 0.57 |
| **MHA Grafting** | | | | | | |
| Random Init | 100% | 1.66 | 289.23 | 154.00 | 0.00 | 0.00 |
| Self-grafting | 100% | 287.81 | 2.49 | 4.71 | 0.83 | 0.56 |
| Hyena-SE | 50% | 274.73 | 2.73 | 5.05 | 0.82 | 0.56 |
| | 75% | 231.15 | 3.62 | 6.04 | 0.81 | 0.54 |
| | 100% | ✗ | ✗ | ✗ | ✗ | ✗ |
| Hyena-X | 50% | 273.30 | 2.74 | 5.03 | 0.83 | 0.56 |
| | 75% | 229.11 | 3.69 | 6.10 | 0.81 | 0.53 |
| | 100% | ✗ | ✗ | ✗ | ✗ | ✗ |
| Hyena-Y | 50% | 273.37 | 2.72 | 5.02 | 0.83 | 0.55 |
| | 75% | 228.99 | 3.66 | 5.95 | 0.81 | 0.53 |
| | 100% | ✗ | ✗ | ✗ | ✗ | ✗ |
| SWA | 50% | 280.62 | 2.67 | 4.90 | 0.83 | 0.56 |
| | 75% | 249.99 | 3.09 | 5.54 | 0.82 | 0.55 |
| | 100% | ✗ | ✗ | ✗ | ✗ | ✗ |
| Mamba-2 | 50% | 285.08 | 2.65 | 4.84 | 0.83 | 0.55 |
| | 75% | 257.66 | 3.02 | 5.48 | 0.82 | 0.53 |
| | 100% | ✗ | ✗ | ✗ | ✗ | ✗ |
| **MLP Grafting** | | | | | | |
| Random Init | 100% | 1.27 | 314.72 | 204.99 | 0.00 | 0.00 |
| Self-grafting | 100% | 277.72 | 2.54 | 4.52 | 0.83 | 0.57 |
| Width=3 | 50% | 272.14 | 2.53 | 4.51 | 0.83 | 0.57 |
| | 75% | 279.72 | 2.61 | 4.61 | 0.83 | 0.56 |
| | 100% | 252.11 | 2.66 | 4.57 | 0.81 | 0.57 |
| Width=6 | 50% | 278.00 | 2.38 | 4.50 | 0.83 | 0.58 |
| | 75% | 277.94 | 2.37 | 4.48 | 0.82 | 0.58 |
| | 100% | 276.86 | 2.42 | 4.50 | 0.82 | 0.58 |
| Hyena-X | 50% | 265.60 | 2.64 | 4.66 | 0.83 | 0.56 |
| | 75% | 226.13 | 3.26 | 4.79 | 0.81 | 0.55 |
| | 100% | ✗ | ✗ | ✗ | ✗ | ✗ |

*Table 1.* **Generation quality when replacing MHA and MLP operators in DiT-XL/2 with efficient alternatives via grafting.** For each alternative, key architectural parameters: kernel size $K = 4$ (Hyena variants), window size $w = 4$ (SWA), state size $d_s = 64$, expand factor $E = 2$ (Mamba-2) and MLP expansion ratio is denoted by $r$. DiT-XL/2 uses 16 attention heads and $r = 4$. The best-performing setup for each alternative is highlighted

## 4. Experiments I: Hybrid Architectures via Grafting

**Design space and experiment setup.** The space of architectural alternatives is vast, raising a practical question: how should we define a representative set of designs to explore under grafting? We focus on two sources of motivation: empirical observations and prior architectural work. First, our attention locality analysis (see Supp. A) reveals that several MHA operators in DiTs exhibit strong local interactions. This suggests that local operators—such as convolutions or sliding-window attention—may serve as effective replacements for global attention. For MLPs, we draw inspiration from prior work on efficient architectures (Fu et al., 2023; Komatsuzaki et al., 2023; Kaplan et al., 2020). To organize

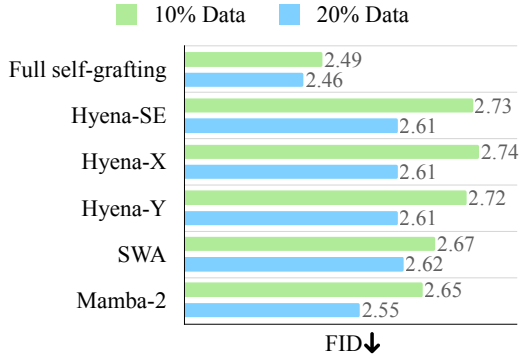


*Figure 2.* Ablation studies analyzing the effect of data scale for MHA operator replacements.

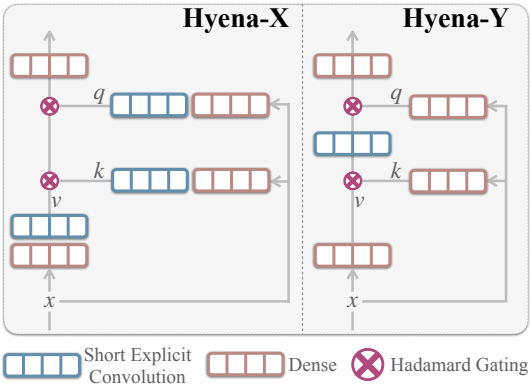


*Figure 3.* Our proposed Hyena-X/ Hyena-Y, efficient local gated convolution operators used as drop-in replacements for MHA.

this exploration, we define four design axes: (1) which operator to replace (e.g., MHA, MLP); (2) what to replace it with (e.g., convolutions); (3) how to select layers for replacement (e.g., all layers); and (4) replacement ratio (full vs. partial). As the design space grows exponentially with depth ($2^L$ for $L$ layers), we apply simple heuristics such as full and interleaved replacement for layer selection strategies—motivated by striped transformer designs (Poli et al., 2024; Ku et al., 2025).

**We introduce Hyena-X and Hyena-Y—two efficient gated convolution operators designed as drop-in replacements for MHA.** While our study includes several off-the-shelf alternatives, we also contribute new operator designs motivated by attention locality analysis. This allows us to test novel architectural ideas via grafting, broadening our study. Both Hyena-X and Hyena-Y are local gated convolutions composed of dense, short causal depth-wise 1D convolutions. Fig. 3 illustrates their structure. We also adapt Hyena-X as an MLP alternative by applying it along the channel dimension. Hyena-X and Hyena-Y scale linearly with sequence length, compared to the quadratic scaling of MHA. Operator details are provided in the Supp.

**Results.** We report results in Tab. 1.

**(i) MHA results.** Replacing MHA operators in DiT-XL/2 via grafting yields strong quality-efficiency tradeoffs. We discuss our key insights below:

- *Surprising effectiveness of operators with smaller receptive fields under interleaved grafting*. Our findings highlight that at 50% interleaved replacement, several alternatives—including SWA, Hyena-X/Y, and Mamba-2—consistently achieve FID scores within 0.5 of the baseline (2.27).
- *Replacement strategy: Interleaved vs. Full.* Performance generally declines when increasing interleaved replacement from 50% to 75%. However, SWA remains effective at 75% interleaved replacement (FID=3.09). At 100% replacement, performance sharply degrades (all FIDs $> 50$). This trend aligns with our locality analysis, indicating that only a subset of layers are local and amenable to grafting.
- *Ablations on data scale (Fig. 2).* Under 50% MHA replacement: Increasing fine-tuning data from 10% to 20% improves FID across all variants (e.g., Hyena-X: 2.74→2.61; SWA: 2.67→2.62; Mamba-2: 2.65→2.55).

**(ii) MLP results.** Replacing MLP operators via grafting is effective. We discuss our key insights below:

- *Variable expansion ratio MLPs are effective under full replacement.* MLP alternatives with expansion ratio $r$=3 and $r$=6 demonstrate good quality under all replacement ratios. Even under full (100%) replacement, both variants maintain good performance, with $r$=3 achieving FID=2.66. This highlights that MLP width is a robust dimension for grafting.
- *Convolutional alternatives.* Hyena-X which combines dense and local channel mixing, performs competitively at 50% replacement (FID=2.63) but degrades at higher ratios, suggesting that such operators are only effective at moderate ratios.

*Takeaway 1: Grafting is effective for constructing efficient hybrid architectures with good generative quality under small compute budgets. Interleaved designs are particularly effective.*

## 5. Experiments II: Grafting Text-to-Image Diffusion Transformers

**Motivation.** We apply grafting to a more challenging setting: text-to-image generation (2048×2048) with PixArt-Σ (Chen et al., 2024). This presents three challenges: (1) long sequences (16,384 tokens), (2) a multimodal setup with text conditioning, and (3) lack of publicly available training

| Model | Ratio | Obj(1) | Obj(2) | Count | Colors | Pos | ColorAttr | Overall ↑ | Latency (ms) ↓ |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | - | 81.45 | 61.62 | 46.25 | 77.13 | 10.75 | 21.50 | 49.75 | 235.46 |
| Hyena-X | 29% | 80.31 | 59.34 | 49.69 | 68.62 | 11.50 | 18.75 | 48.04 | 194.95 (1.21×) |
| Hyena-X | 50% | 80.00 | 57.07 | 48.13 | 70.74 | 11.25 | 19.50 | **47.78** | **164.58** (1.43×) |

*Table 2.* **GenEval results and inference latency for PixArt-Σ and the grafted variants.** The 50% grafted model achieves a 1.43× speedup while retaining strong text-image alignment (GenEval overall score: 47.78 vs. 49.75). Latency is measured for a single forward pass on an Nvidia H100 (batch size=2).

| Method | Depth | A.R | Iters | IS ↑ | FID ↓ | sFID ↓ | Prec. ↑ | Recall ↑ | Speedup ↑ | Params ↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| DiT-L/2 (Peebles & Xie, 2023) | 24 | 42.7 | 1,000K | 196.26 | 3.73 | 4.62 | 0.82 | 0.54 | — | 458M |
| U-ViT-L (Bao et al., 2023) | 21 | 48.8 | 300K | 221.29 | 3.44 | 6.58 | 0.83 | 0.52 | — | 287M |
| DiT-B/2 (Peebles & Xie, 2023) | 12 | 64.0 | 1000K | 119.63 | 10.12 | 5.39 | 0.73 | 0.55 | — | 130M |
| BK-SDM (Kim et al., 2024) | 14 | 82.3 | 100K | 141.18 | 7.43 | 6.09 | 0.75 | 0.55 | 2× | 340M |
| TinyDiT-D14 (Fang et al., 2024) | 14 | 82.3 | 500K | 198.85 | 3.92 | 5.69 | 0.78 | 0.58 | 2× | 340M |
| TinyDiT-D14 w/ MKD (Fang et al., 2024) | 14 | 82.3 | 500K | 234.50 | 2.86 | 4.75 | 0.82 | 0.55 | 2× | 340M |
| DiT-XL/2 (Peebles & Xie, 2023) | 28 | 41.4 | 7,000K | 278.20 | 2.27 | 4.60 | 0.83 | 0.57 | 1× | 675M |
| **Grafting (Ours)** | 14 | 164.6 | 100K | 231.91 | 3.12 | 4.71 | 0.82 | 0.55 | 2×¶ | 712M |
| **Grafting (Ours)** | 14 | 164.6 | 230K | 251.77 | 2.77 | 4.87 | 0.82 | 0.56 | 2×¶ | 712M |

*Table 3.* **Generative quality vs. model depth.** We report generative quality metrics (IS, FID, sFID, Precision, and Recall). A.R. (Aspect Ratio) is defined as model width divided by depth (e.g., 1152/14 = 82.3). Parameters (Params) are reported in millions. For pruning and grafting setups, we report speedup with respect to DiT-XL/2 (depth=28). Off-the-shelf DiT-L/2, U-ViT-L, and DiT-B/2 scores, along with pruning baselines (BK-SDM, TinyDiT-D14, and TinyDiT-D14 w/ MKD), are sourced from (Fang et al., 2024). MKD refers to Masked Knowledge Distillation, a recovery method used in TinyDiT (Fang et al., 2024). ¶ Speedup is measured for a single forward pass on an Nvidia H100 (batch size=2). More details are provided in Sec. D. Our grafted models achieve better generative quality at depth=14, surpassing baselines in FID, IS, Precision, and Recall.

data. These factors make PixArt-Σ a representative setting for evaluating grafting under real-world constraints.

**Experiment setup.** We replace self-attention layers (MHA) in PixArt-Σ with Hyena-X via grafting, as MHA accounts for over 62% of generation latency. Hyena-X was chosen based on its good performance in the ImageNet setup, achieving FID 2.61 with 20% data. Interleaved grafting is applied at layers 8, 10, 12, 14, 16, 18, and 20–27; empirically, we found that layers 20–27 can be grafted without significant quality drop. We follow the two-stage grafting procedure. *Stage 1 (activation distillation):* 8k uncurated synthetic image-text pairs (from a total of 12k) are used to initialize Hyena-X blocks. *Stage 2 (finetuning):* The full 12k dataset is used to finetune the grafted model using LoRA (rank=64).

**Results.** The grafted model achieves a 1.43× speedup in wall-clock time, with a small drop in GenEval score (47.78 vs. 49.75). Attribute-specific metrics remain comparable, and qualitative samples show good alignment and quality. Some localized artifacts are observed in textured regions likely due to LoRA's adaptation capacity and low-quality synthetic data (see failure cases in Supp E.2).

*Takeaway 2: We graft high-resolution text-to-image DiTs, constructing hybrid architectures with meaningful speedups and minimal quality drop.*

## 6. Case Study: Converting Model Depth to Width via Grafting

**Can we rewire two sequential transformer blocks to run in parallel?** Our MLP grafting results showed that MLPs are amenable to grafting, even at 100% replacement with an expansion ratio of $r = 6$, demonstrating that wider computation within an operator is feasible. This success, combined with the fact that modern GPUs favor parallel over sequential computation, motivates a broader question: can we convert deeper, sequential DiT computations into wider, parallel ones via grafting while maintaining quality? To explore this, we rewire DiT-XL/2 by parallelizing every pair of sequential transformer blocks—each pair receives the same input, and their outputs are merged via a linear projection. This reduces model depth by 2× (28 → 14) with a 6% increase in parameters (See Fig. 4).

**We trade depth for width via grafting, achieving better quality than models of comparable depth.** We report results in Tab. 3. To contextualize this, we compare against two categories: (i) DiTs trained from scratch at lower depth, and (ii) pruning methods (Kim et al., 2024; Fang et al., 2024). Our grafted model achieves better generative quality compared to these baselines.

*Takeaway 3: Grafting enables architectural restructuring at the transformer block level, allowing model depth to be traded for width.*

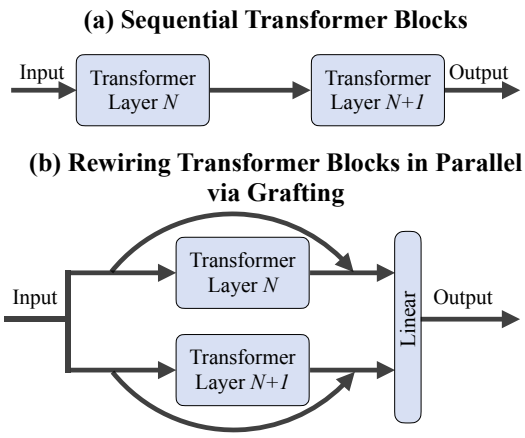


*Figure 4.* Convert model depth → width via grafting: (a) Two sequential transformer layers. (b) Rewiring in parallel via grafting (includes skip connections).

## 7. Related Work

**Diffusion model architectures.** Recently, many architectural innovations have been proposed for diffusion models for image and video generation (Xie et al., 2024; Yan et al., 2024; Fei et al., 2024a; Hu et al., 2024; Teng et al., 2024; Zhu et al., 2024; Seawead et al., 2025; Xing et al., 2024; Gao et al., 2024; Wang et al., 2024a). Many recent works focus on improving the attention mechanism in diffusion models to enhance efficiency and scalability. One major direction is the use of modern linear attention variants, such as DiffuSSM (Yan et al., 2024), DiS (Fei et al., 2024a), Zigma (Hu et al., 2024), DiM (Teng et al., 2024), and DIG (Zhu et al., 2024). Recently, text-to-image diffusion models such as SANA (Xie et al., 2024) have also adapted linear attention variants to support high-resolution generation. Another recent direction explores the mixture-of-experts (MoE) idea. DiT-MoE (Fei et al., 2024b) introduces sparse diffusion transformers with shared expert routing and expert-level balance loss, enabling efficient scaling to 16.5B parameters while achieving competitive performance. We note that methods like STAR (Thomas et al., 2025) have also successfully discovered architectures via evolutionary methods for autoregressive language modeling. While effective, these approaches require training from scratch, making such studies expensive and inaccessible to practitioners. In contrast, grafting focuses on architecture editing of pretrained models to materialize new architectures under small compute budgets.

**Architectural editing of pretrained generative models.** Another line of work focuses on linearizing large language models by replacing softmax attention with efficient operators, such as linear attention (Zhang et al., 2025; Wang et al., 2024b; Bick et al., 2024). Similar ideas have also been adopted for diffusion models in (Liu et al., 2024a;b; Becker et al., 2025), though these works focus only on ultra-high-resolution settings. These prior efforts typically focus on replacing a single operator type (primarily attention) or are limited to specific application domains. Grafting presents a more general and comprehensive approach for architectural editing. It extends beyond single-operator replacement to enable modifying multiple operator types, exploring diverse architectural alternatives (e.g., both MHA and MLP replacements), and restructuring architectures (e.g., converting model depth to width). Recently, FFN Fusion (Bercovich et al., 2025) explored parallelizing transformer blocks in LLMs, aiming to reduce sequential computation.

## 8. Discussion

We introduced *grafting*, a simple approach for materializing new architectures by editing pretrained diffusion transformers. We applied it to construct hybrid models by replacing self-attention and MLPs with efficient alternatives, achieving competitive quality (FID 2.38–2.64 vs. 2.27 baseline) using small compute budgets. We further grafted a high-resolution text-to-image model (PixArt-$\Sigma$), achieving a 43% speedup with less than 2% drop in GenEval score. Finally, we used grafting to restructure DiT-XL/2 by converting sequential computation into parallel, reducing model depth by 2×, achieving better quality (FID 2.77) among 14-layer DiTs. These results position grafting as a lightweight approach for architectural exploration under small compute budgets. We hope that our insights and results will encourage the community to explore new architecture designs, with grafting as a practical tool.

**Applications and future work.** Grafting holds promise for diverse applications where efficiency is important. This includes adapting models from low-resolution to high-resolution settings, extending capabilities from short-form video understanding/generation to long-form (Chandrasegaran et al., 2024; Chen et al., 2025), or improving user experience in interactive applications like image editing where even modest gains (e.g., 10% speedup) are highly valued. Code and grafted models: grafting.stanford.edu.

## Acknowledgments

We thank Liquid AI for sponsoring compute for this project. We also thank Armin W. Thomas, Garyk Brixi, Kyle Sargent, Karthik Dharmarajan, Stephen Tian, Cristobal Eyzaguirre, and Aryaman Arora for their feedback on the manuscript.

## References


Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., and Zhu, J. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023.

Becker, P., Mehrotra, A., Chavhan, R., Chadwick, M., Morreale, L., Noroozi, M., Ramos, A. G., and Bhattacharya, S. Edit: Efficient diffusion transformers with linear compressed attention. *arXiv preprint arXiv:2503.16726*, 2025.

Bercovich, A., Dabbah, M., Puny, O., Galil, I., Geifman, A., Geifman, Y., Golan, I., Karpas, E., Levy, I., Moshe, Z., et al. Ffn fusion: Rethinking sequential computation in large language models. *arXiv preprint arXiv:2503.18908*, 2025.

Bick, A., Li, K., Xing, E., Kolter, J. Z., and Gu, A. Transformers to ssms: Distilling quadratic knowledge to subquadratic models. *Advances in Neural Information Processing Systems*, 37:31788–31812, 2024.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Chandrasegaran, K., Gupta, A., Hadzic, L. M., Kota, T., He, J., Eyzaguirre, C., Durante, Z., Li, M., Wu, J., and Li, F.-F. Hourvideo: 1-hour video-language understanding. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Chen, G., Li, Z., Wang, S., Jiang, J., Liu, Y., Lu, L., Huang, D.-A., Byeon, W., Le, M., Rintamaki, T., et al. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. *arXiv preprint arXiv:2504.15271*, 2025.

Chen, J., Jincheng, Y., Chongjian, G., Yao, L., Xie, E., Wang, Z., Kwok, J., Luo, P., Lu, H., and Li, Z. Pixart-$\alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2023.

Chen, J., Ge, C., Xie, E., Wu, Y., Yao, L., Ren, X., Wang, Z., Luo, P., Lu, H., and Li, Z. Pixart-$\sum$: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024.

Fang, G., Li, K., Ma, X., and Wang, X. Tinyfusion: Diffusion transformers learned shallow. *arXiv preprint arXiv:2412.01199*, 2024.

Fei, Z., Fan, M., Yu, C., and Huang, J. Scalable diffusion models with state space backbone. *arXiv preprint arXiv:2402.05608*, 2024a.

Fei, Z., Fan, M., Yu, C., Li, D., and Huang, J. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024b.

Fu, D., Arora, S., Grogan, J., Johnson, I., Eyuboglu, E. S., Thomas, A., Spector, B., Poli, M., Rudra, A., and Ré, C. Monarch mixer: A simple sub-quadratic gemm-based architecture. *Advances in Neural Information Processing Systems*, 36:77546–77603, 2023.

Gao, Y., Huang, J., Sun, X., Jie, Z., Zhong, Y., and Ma, L. Matten: Video generation with mamba-attention. *arXiv preprint arXiv:2405.03025*, 2024.

Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Fei-Fei, L., Essa, I., Jiang, L., and Lezama, J. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hu, V. T., Baumann, S. A., Gui, M., Grebenkova, O., Ma, P., Fischer, J., and Ommer, B. Zigma: Zigzag mamba diffusion model. *arXiv preprint arXiv:2403.13802*, 2024.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kim, B.-K., Song, H.-K., Castells, T., and Choi, S. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pp. 381–399. Springer, 2024.

Komatsuzaki, A., Puigcerver, J., Lee-Thorp, J., Ruiz, C. R., Mustafa, B., Ainslie, J., Tay, Y., Dehghani, M., and Houlsby, N. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations*, 2023.

Ku, J., Nguyen, E., Romero, D. W., Brixi, G., Yang, B., Vorontsov, A., Taghibakhshi, A., Lu, A. X., Burke, D. P., Brockman, G., et al. Systems and algorithms for convolutional multi-hybrid language models at scale. *arXiv preprint arXiv:2503.01868*, 2025.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, S., Tan, Z., and Wang, X. Clear: Conv-like linearization revs pre-trained diffusion transformers up. *arXiv preprint arXiv:2412.16112*, 2024a.

Liu, S., Yu, W., Tan, Z., and Wang, X. Linfusion: 1 gpu, 1 minute, 16k image. *arXiv preprint arXiv:2409.02097*, 2024b.

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

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Poli, M., Thomas, A. W., Nguyen, E., Ponnusamy, P., Deiseroth, B., Kersting, K., Suzuki, T., Hie, B., Ermon, S., Ré, C., et al. Mechanistic design and scaling of hybrid architectures. *arXiv preprint arXiv:2403.17844*, 2024.

Seawead, T., Yang, C., Lin, Z., Zhao, Y., Lin, S., Ma, Z., Guo, H., Chen, H., Qi, L., Wang, S., Cheng, F., Zeng, F. Z. X., Yang, Z., Kong, F., Qing, Z., Xiao, F., Wei, M., Hoang, T., Zhang, S., Zhu, P., Zhao, Q., Yan, J., Gui, L., Bi, S., Li, J., Ren, Y., Wang, R., Li, H., Xiao, X., Liu, S., Ling, F., Zhang, H., Wei, H., Kuang, H., Duncan, J., Zhang, J., Zheng, J., Sun, L., Zhang, M., Sun, R., Zhuang, X., Li, X., Xia, X., Chi, X., Peng, Y., Wang, Y., Wang, Y., Zhao, Z., Chen, Z., Song, Z., Yang, Z., Feng, J., Yang, J., and Jiang, L. Seaweed-7b: Cost-effective training of video generation foundation model. 2025. URL https://api.semanticscholar.org/CorpusID:277740920.

Teng, Y., Wu, Y., Shi, H., Ning, X., Dai, G., Wang, Y., Li, Z., and Liu, X. Dim: Diffusion mamba for efficient high-resolution image synthesis. *arXiv preprint arXiv:2405.14224*, 2024.

Thomas, A. W., Parnichkun, R., Amini, A., Massaroli, S., and Poli, M. Star: Synthesis of tailored architectures. In *The Thirteenth International Conference on Learning Representations*, 2025.

Wang, H., Ma, C.-Y., Liu, Y.-C., Hou, J., Xu, T., Wang, J., Juefei-Xu, F., Luo, Y., Zhang, P., Hou, T., et al. Lingen: Towards high-resolution minute-length text-to-video generation with linear computational complexity. *arXiv preprint arXiv:2412.09856*, 2024a.

Wang, J., Paliotta, D., May, A., Rush, A. M., and Dao, T. The mamba in the llama: Distilling and accelerating hybrid models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=uAzhODjALU.

Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Xing, Z., Feng, Q., Chen, H., Dai, Q., Hu, H., Xu, H., Wu, Z., and Jiang, Y.-G. A survey on video diffusion models. *ACM Comput. Surv.*, 57(2), November 2024. ISSN 0360-0300. doi: 10.1145/3696415. URL https://doi.org/10.1145/3696415.

Yan, J. N., Gu, J., and Rush, A. M. Diffusion models without attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8239–8249, 2024.

Zhang, M., Arora, S., Chalamala, R., Spector, B. F., Wu, A., Ramesh, K., Singhal, A., and Re, C. Lolcats: On low-rank linearizing of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhu, L., Huang, Z., Liao, B., Liew, J. H., Yan, H., Feng, J., and Wang, X. Dig: Scalable and efficient diffusion models with gated linear attention. *arXiv preprint arXiv:2405.18428*, 2024.

## Supplementary Material



## A. Locality Analysis of Self-attention

MHA scales quadratically with sequence length, making it a computational bottleneck. A natural idea is to replace it with local operators, such as convolution or local attention. However, this will fail if the model relies on long-range dependencies: for example, replacing all MHA operators in DiT-XL/2 with a sliding window attention degrades FID from 2.27 to 199.3. To guide grafting, we quantify attention locality using a simple band-$k$ metric. Given an attention matrix $A \in \mathbb{R}^{N\times N}$, we define a bi-directional band indicator matrix $B_k \in \mathbb{R}^{N\times N}$ as:

$$(B_k)_{i,j} = \begin{cases} 1, & \text{if } |i-j| \le k \\ 0, & \text{otherwise} \end{cases}$$

Then, locality within a band of size $k$ is computed as:

$$L_k = \frac{1}{N}\sum_{i,j} A_{i,j}(B_k)_{i,j} \tag{3}$$

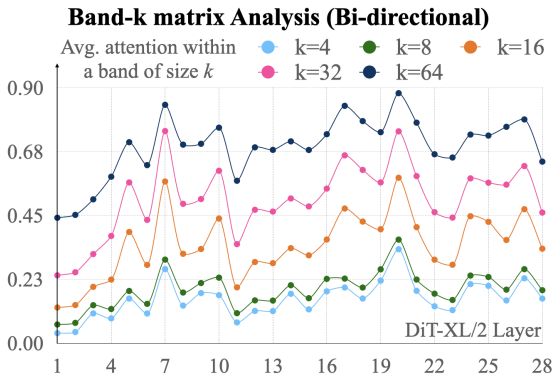


*Figure A.1.* **Locality of self-attention in DiT-XL/2 quantified using band-$k$ metric.** We plot band-$k$ values for all 28 layers of DiT-XL/2, averaged over timesteps and samples. At $k$=32, 15 out of 28 layers exhibit values exceeding 0.5, indicating that several MHA operators model local interactions.

We compute $L_k$ for all 28 MHA operators in DiT-XL/2 using 50-step DDIM sampling on 250 ImageNet samples (sequence length 256, cfg scale 1.5), averaging across timesteps and samples. As shown in Fig. A.1, MHA is largely local: for $k$=32, 15 layers attend more than 50% within the band. Early layers are less local. Our analysis provides guidance for replacing MHA with efficient alternatives.

## B. Standard Deviation of Experiments

To compute variance associated with our reported results, we repeat two representative experiments—MHA (Hyena-Y) and MLP (width=6)—using three different random seeds (`seed = 0, 200, 300`). We follow the exact grafting setup used in the main paper for these experiments. We report the mean and standard deviation of IS, FID, sFID, Precision and Recall in Tab. B.1. We observe that the standard deviations are within an acceptable range.

| Setup | IS | FID | sFID | Precision | Recall |
|---|---|---|---|---|---|
| **MHA/ Hyena-Y** | $273.19 \pm 0.46$ | $2.73 \pm 0.01$ | $5.06 \pm 0.04$ | $0.83 \pm 0.00$ | $0.55 \pm 0.00$ |
| **MLP/ higher width** ($r = 6$) | $277.91 \pm 0.95$ | $2.41 \pm 0.01$ | $4.48 \pm 0.02$ | $0.82 \pm 0.00$ | $0.58 \pm 0.00$ |

*Table B.1.* Mean and standard deviation of IS, FID, sFID, Precision and Recall calculated for three runs with different seeds (`0,200,300`).

# C. Hybrid Architecture Experiments: Additional details

## C.1. Experiment details and additional samples

We provide all hyperparameters used for these experiments in Tab. C.1. To ensure a fair comparison, we used identical hyper-parameters across every hybrid experiment. Each hybrid experiment requires less than 24 hours of wall clock time on 8xH100 GPUs. We include additional qualitative samples generated using our hybrid architectures obtained via grafting in Fig. C.1.

| **Stage 1: Activation Distillation** | |
|---|---|
| Initial Learning Rate | $1 \times 10^{-3}$ |
| Weight Decay | 0 |
| Epochs | 200 |
| Batch Size | 64 |
| Clip Norm | 10.0 |
| Optimizer | AdamW ($betas = (0.9, 0.999)$) |
| Loss Function | L1 (MHA), L2 (MLP) |
| **Stage 2: Lightweight Finetuning** | |
| Initial Learning Rate | $1 \times 10^{-4}$ |
| Weight Decay | $5 \times 10^{-5}$ |
| Iterations | 50,000 (100 epochs) |
| Batch Size | 256 |
| Optimizer | AdamW ($betas = (0.9, 0.999)$) |
| Scheduler | Linear Warmup over 1K steps, then constant lr |
| Training Data | 10% of ImageNet-1K (128k samples) |

*Table C.1.* Hyperparameters for MHA/MLP grafting experiments using DiT-XL/2 (ImageNet-1K).

## C.2. Modulated Regression Targets for MHA

For Stage 1, we explored a modulation-aware regression variant for MHA experiments that incorporates the learned scalar (`gate_msa`) applied to the attention output. In the standard setup, we regress from input $x$ to the raw output of the attention block $y = \texttt{MHA}(\cdot)$. In the modulation-aware formulation, the target becomes $y = \texttt{gate\_msa} \odot \texttt{MHA}(\cdot)$. Tab. C.2 compares these two variants with L1 and L2 loss. Modulation-aware regression increases target scale, which adversely affects L2 loss performance due to its sensitivity to large values. L1 performs similarly in both settings. We adopted the standard (modulation-agnostic) formulation for all experiments for simplicity.

# D. Depth to Width Grafting Experiments: Additional details

We provide all hyperparameters in Tab. D.1 and additional samples in Fig. D.1. As one can observe, these grafted models produce realistic samples.

# E. Text-to-Image Generation Experiments: Additional details

## E.1. Experiment details

We provide all hyperparameters used in our PixArt-$\Sigma$ grafting experiments in Tab. E.1.

| | Modulation-aware | IS | FID | sFID | Precision | Recall |
|---|---|---|---|---|---|---|
| Baseline | | 278.20 | 2.27 | 4.60 | 0.83 | 0.57 |
| L2 | ✓ | 246.17 | 3.00 | 7.11 | 0.79 | 0.58 |
| | ✗ | 269.31 | 2.58 | 5.75 | 0.82 | 0.58 |
| L1 | ✓ | 272.86 | 2.51 | **5.29** | 0.82 | 0.57 |
| | ✗ | **273.03** | **2.51** | 5.48 | **0.83** | **0.58** |

*Table C.2.* Comparison of modulation-aware and standard regression targets for Stage 1. The modulation-aware setup includes the learned scalar (`gate_msa`) as a multiplicative factor in the regression target. L2 loss is sensitive to the amplified target scale and performs worse, while L1 loss remains robust and performs similarly in both cases. We adopt the standard formulation by default.


*Figure C.1.* **Figure B.1**: Additional generated samples from grafted DiT-XL/2 models using various architectural edits. Each row corresponds to a different hybrid variant. FID scores (*lower is better*, ImageNet-1K 256×256) are shown in parentheses. *MHA variants (top 4 rows)*: Hyena-X (2.61), Hyena-Y (2.61), SWA (2.62), Mamba-2 (2.55) *MLP variants (bottom 3 rows)*: Lower width (2.53), Higher width (2.38), Hyena-X MLP (2.64). These results highlight the flexibility of grafting in constructing high-quality hybrid architectures by replacing MHA and MLP operators.

*Figure D.1.* **Figure B.2**: Depth-to-width grafting. Samples from a DiT-XL/2 model in which every pair of transformer blocks is merged into a parallel block, effectively reducing depth by 2× while increasing width (FID=2.77).

| Stage 1: Activation Distillation | |
|---|---|
| Initial Learning Rate | $1 \times 10^{-4}$ |
| Regression Objective | L1 |
| Epochs | 200 |
| Batch Size | 64 |
| Optimizer | AdamW ($betas = (0.9, 0.999)$) |
| **Stage 2: Lightweight Finetuning** | |
| Learning Rate | $5 \times 10^{-5}$ |
| Weight Decay | 0 |
| Iterations | 360k (360 epochs) |
| Batch Size | 256 |
| Optimizer | AdamW ($betas = (0.9, 0.95)$) |
| Scheduler | Warmup over 1K steps, then half every 100k steps |
| Training Data | 20% of ImageNet-1K (256k) |

*Table D.1.* Hyperparameters for depth-to-width grafting experiments using DiT-XL/2 (ImageNet-1K).

### E.2. Generated samples and failure cases

We show additional high-resolution samples generated by the grafted PixArt-Σ model in Fig. E.1, illustrating the model's ability to preserve generative quality across diverse prompts despite substantial architectural edits.

Figure E.2 illustrates two types of failure modes observed in grafted PixArt-Σ outputs. Each column pair shows the output of PixArt-Σ (left) and the grafted model (right) for the same text prompt. In the top row, the original model generates images that are reasonably aligned with the prompts, while the grafted model fails to preserve this alignment—indicating limitations during the LoRA-based finetuning stage. In the bottom row, the synthetic supervision itself is of low quality, resulting in poor outputs from both the original and grafted models. To better understand this issue, Figure E.3 presents additional examples of low-quality synthetic data produced by PixArt-Σ and used for grafting. These samples often exhibit artifacts and unrealistic physics. While synthetic data enables low-cost adaptation, these results highlight the importance of improved data curation and filtering to avoid propagating errors during the grafting process.


*Figure E.1.* Additional 2048x2048 samples generated using our grafted PixArt-Σ model.

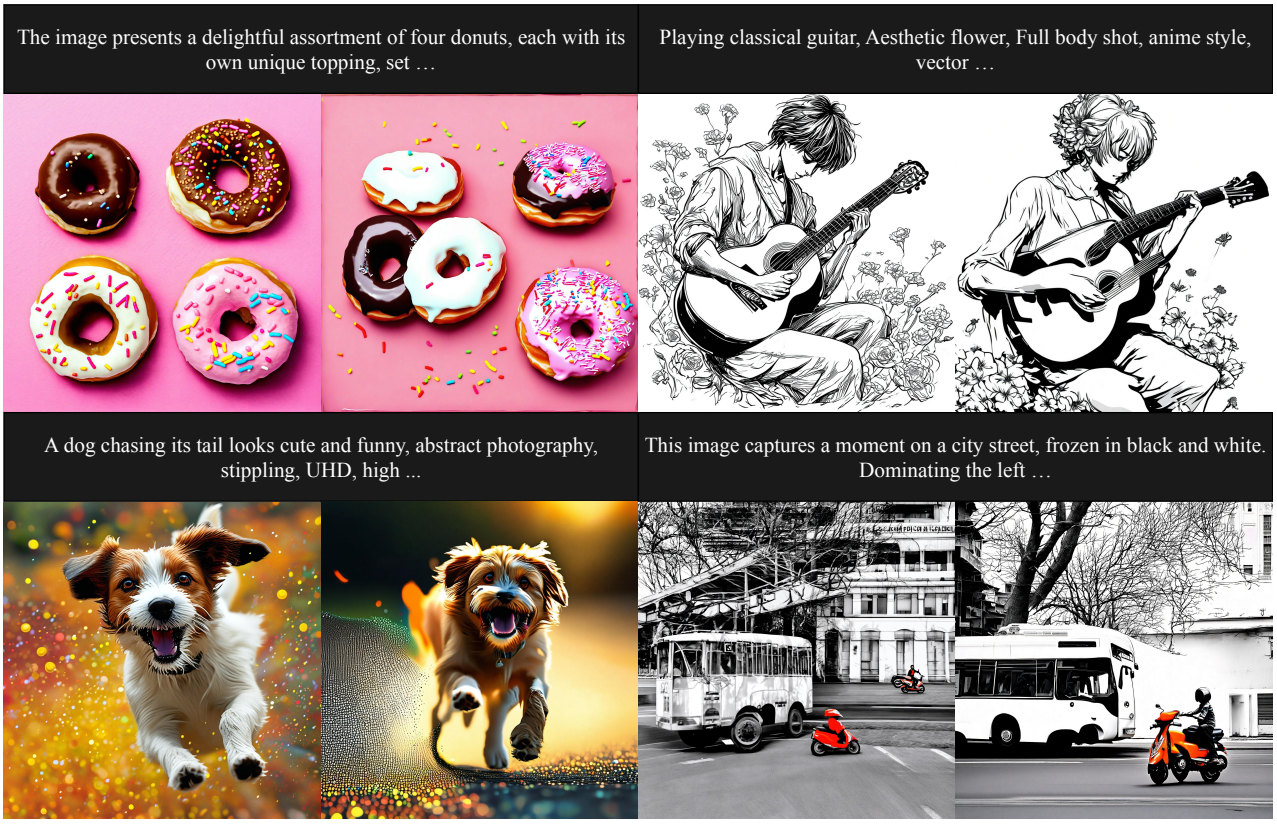


*Figure E.2.* **Failure cases in text-to-image generation.** Each column pair shows outputs from PixArt-Σ (left) and the grafted model (right) for the same prompt. In the top row, the prompt specifies four donuts with unique toppings and a full-body anime-style character playing classical guitar. The grafted outputs deviate from these prompts—showing incorrect object counts (e.g., five donuts) and degraded structure (e.g., distorted hands), reflecting text-image misalignment and visual artifacts introduced during grafting. In the bottom row, the supervision itself is poor: prompts such as a dog chasing its tail in UHD stippling style and a black-and-white street photo are not faithfully captured by either model. These examples highlight challenges arising both from LORA finetuning and low-quality synthetic data.

## F. Hyena-X and Hyena-Y operators: Additional details

Informed by our band-k analysis of MHA operators, we introduce a collection of efficient operators designed to exploit the locality in attention matrices. Given an input $x \in \mathbb{R}^{\ell \times d}$, a generic Hyena operator performs the following transformation:

$$
\begin{aligned}
q_s^c &= \sum_{s'} T_{ss'}^c \sum_{c'} x_{s'}^{c'} W^{c'c} \\
k_s^c &= \sum_{s'} H_{ss'}^c \sum_{c'} x_{s'}^{c'} U^{c'c} \\
v_s^c &= \sum_{s'} K_{ss'}^c \sum_{c'} x_{s'}^{c'} P^{c'c} \\
y_s^c &= \sum_{c'} \sum_{s'} (q_s^{c'} G_{ss'}^{c'} k_{s'}^{c'} v_{s'}^{c'}) M^{c'c}
\end{aligned}
$$

where $W, U, P, M \in \mathbb{R}^{d \times d}$ are parametrized as dense or low-rank matrices, and $T, H, K, G \in \mathbb{R}^{\ell \times \ell}$ are Toeplitz matrices corresponding to convolutions with the filters $h_T, h_H, h_K, h_G$, respectively. In the original formulation (Poli et al., 2023), the filters $h_T, h_H, h_K$ are short and explicitly parametrized, whereas $h_G$ is implicitly parametrized.

We build on this formulation and propose `Hyena-X` and `Hyena-Y`, two Hyena operators designed for grafting. `Hyena-X`

*Figure E.3.* **Examples of low-quality samples generated by PixArt-$\Sigma$ used for grafting**. These images contain unrealistic features, inconsistent physics, and visual artifacts. Their presence in the grafting dataset can degrade downstream generation quality, highlighting the importance of data curation when using synthetic data.

removes the implicit convolution entirely by setting $G = I$. In contrast, `Hyena-Y` introduces two changes: (i) it removes all three featurizer convolutions ($T$, $H$, $K$), and (ii) replaces the implicit long convolution in $G$ with a short, explicit convolution. This modified structure preserves local inductive bias while significantly reducing computational cost. An illustration is provided in main paper. These operators allows us to realize speedups across a range of inputs resolutions: both `Hyena-X` and `Hyena-Y` are faster than Mamba-2 operators on all input sequence lengths, including lower resolution regimes.

## G. FLOPs expressions

Notations are provided in Tab. G.1.

### G.1. MHA

- **Input projections (Q, K, V)**: $6LD^2$
- **Softmax attention computation**: $4L^2D + 2HL^2$
- **Output projection**: $2LD^2$

### G.2. SWA

- **Input projections (Q, K, V)**: $6LD^2$
- **Sliding window attention (Bidirectional)**: $4L(2w + 1)D + 2HL(2w + 1)$
- **Output projection**: $2LD^2$

### G.3. Hyena-SE

- **Input projections**: $6LD^2$
- **Featurizer**: $3LDK \times 2$

| Stage 1: Activation Distillation | |
|---|---|
| Initial Learning Rate | $1 \times 10^{-4}$ |
| Weight Decay | $1 \times 10^{-5}$ |
| Epochs | 100 |
| Clip Norm Value | 0.1 (Layers 20-27), 0.01 (Other layers) |
| Batch Size | 16 |
| Optimizer | AdamW |
| Scheduler | Reduce lr by 0.5 at epochs = 50, 100, 150 |
| **Stage 2: Lightweight Finetuning** | |
| Initial Learning Rate | $1 \times 10^{-5}$ |
| Weight Decay | 0 |
| Iterations | 18,000 |
| Batch Size | 64 (with gradient accumulation) |
| Optimizer | AdamW |
| Scheduler | linear warmup (500 steps), then constant lr |
| LoRA rank | 64 |

*Table E.1.* Hyperparameters used for PixArt-$\Sigma$ grafting experiments.

- **Inner filter convolution**: $LDK \times 2$
- **gates**: $LD \times 2$
- **Output projection**: $2LD^2$

### G.4. Hyena-X

- **Input projections**: $6LD^2$
- **Featurizer**: $3LDK \times 2$
- **gates**: $LD \times 2$
- **Output projection**: $2LD^2$

### G.5. Hyena-Y

- **Input projections**: $6LD^2$
- **Inner filter convolution**: $LDK \times 2$
- **gates**: $LD \times 2$
- **Output projection**: $2LD^2$

### G.6. Hyena-X (MLP)

- **Dense input projections**: $6LD^2r$
- **Featurizer**: $3LDK \times 2$
- **Gates**: $LD \times 2$
- **Dense output projections**: $2LD^2r$

| Symbol | Description |
|---|---|
| $L$ | Sequence length |
| $D$ | Hidden dimension |
| $H$ | Number of attention heads |
| $K$ | Kernel size for convolutions |
| $w$ | Window size for sliding window attention |
| $r$ | MLP expansion ratio |
| $E$ | Expansion factor in Mamba-2 |
| $d_{\text{state}}$ | State size in Mamba-2 |

*Table G.1.* Notation used in FLOPs expressions.

| **MHA Alternatives** | **Δ FLOPs**$_{op.}$ | **Δ FLOPs**$_{feat.}$ | **Δ Params** |
|---|---|---|---|
| Softmax Attention ($H$=16) | — | — | — |
| Hyena-SE ($K$=4) | -99.03% | +0.26% | +0.43% |
| Hyena-X ($K$=4) | -99.81% | +0.26% | +0.33% |
| Hyena-Y ($K$=4) | -99.03% | 0.00% | +0.11% |
| SWA ($w$=4) | -96.48% | 0.00% | 0.00% |
| Mamba-2 ($d_s$=64, $E$=2) | -75.17% | +155.77% | +56.05% |
| **MLP Alternatives** | | | |
| MLP ($r = 4$) | — | — | — |
| Lower width ($r$=3) | -25.00% | — | -25.00% |
| Larger width ($r$=6) | +50.00% | — | +49.99% |
| Hyena-X ($r$=2, $K$=4) | +0.14% | — | +0.03% |

*Table G.2.* Relative changes in FLOPs and parameters for all MHA and MLP alternatives, compared to the baseline DiT-XL/2 operator (MHA with $H$=16 and MLP with $r$=4). For MHA, total cost is split into ΔFLOPs$_{\text{op.}}$ (softmax attention, gating) and ΔFLOPs$_{\text{feat.}}$ (QKV/output projections, convolutions). We do not use this decomposition for MLP variants. Mamba-2 incurs higher ΔFLOPs$_{\text{feat.}}$ due to additional projections. We use sequence length $N$=256 and hidden dimension $D$=1152. Operator-specific notation (e.g., $K$, $r$, $H$, $w$, $d_s$, $E$) is defined in Tab G.1.

### G.7. Mamba-2

- **Projections**: $8LD^2E$
- **Short convolution**: $6LDE$
- **Featurization**: $2LDE(1 + 2d_{\text{state}}) + 2LDE$
- **Associative scan**: $2LDEd_{\text{state}}$
- **Output layer**: $2LD^2E$