TinyFusion: Diffusion Transformers Learned Shallow

Supplementary Material

1. Experimental Details

Models. Our experiments evaluate the effectiveness of three models: DiT-XL, MAR-Large, and SiT-XL. Diffusion Transformers (DiTs), inspired by Vision Transformer (ViT) principles, process spatial inputs as sequences of patches. The DiT-XL model features 28 transformer layers, a hidden size of 1152, 16 attention heads, and a 2×2 patch size. It employs adaptive layer normalization (AdaLN) to improve training stability, comprising 675 million parameters and trained for 1400 epochs. Masked Autoregressive models (MARs) are diffusion transformer variants tailored for autoregressive image generation. They utilize a continuous-valued diffusion loss framework to generate high-quality outputs without discrete tokenization. The MAR-Large model includes 32 transformer layers, a hidden size of 1024, 16 attention heads, and bidirectional attention. Like DiT, it incorporates AdaLN for stable training and effective token modeling, with 479 million parameters trained over 400 epochs. Finally, Scalable Interpolant Transformers (SiTs) extend the DiT framework by introducing a flow-based interpolant methodology, enabling more flexible bridging between data and noise distributions. While architecturally identical to DiT-XL, the SiT-XL model leverages this interpolant approach to facilitate modular experimentation with interpolant selection and sampling dynamics.

Datasets. We prepared the ImageNet 256×256 dataset by applying center cropping and adaptive resizing to maintain the original aspect ratio and minimize distortion. The images were then normalized to a mean of 0.5 and a standard deviation of 0.5. To augment the dataset, we applied random horizontal flipping with a probability of 0.5. To accelerate training without using Variational Autoencoder (VAE), we pre-extracted features from the images using a pre-trained VAE. The images were mapped to their latent representations, normalized, and the resulting feature arrays were saved for direct use during training.

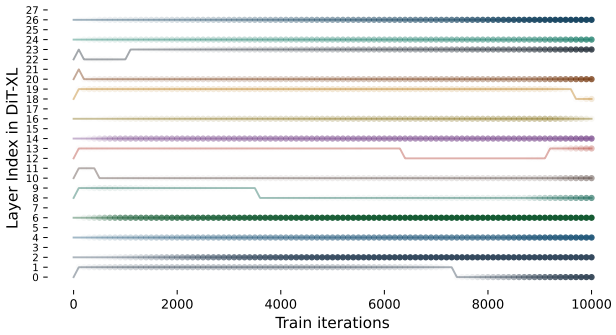

Figure 9. 1:2 Pruning Decisions

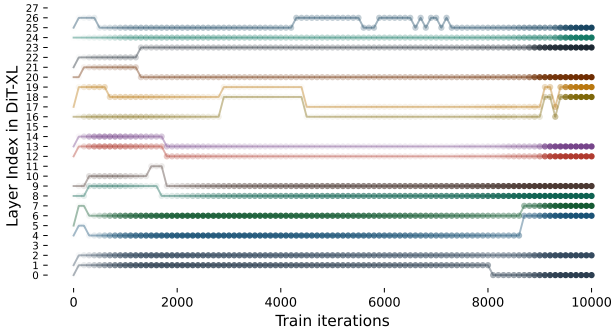

Figure 10. 2:4 Pruning Decisions

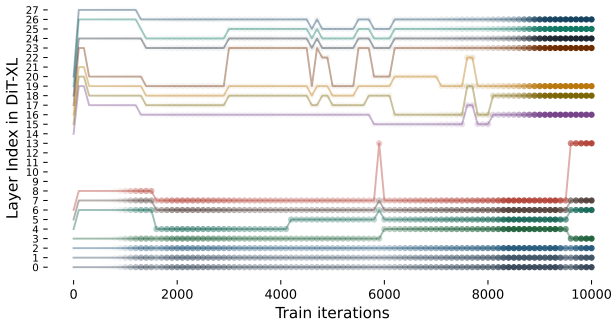

Figure 11. 7:14 Pruning Decisions

Training Details The training process began with obtaining pruned models using the proposed learnable pruning method as illustrated in Figure 12. Pruning decisions were made by a joint optimization of pruning and weight updates through LoRA with a block size. In practice, the block size is 2 for simplicity and the models were trained for 100 epochs, except for MAR, which was trained for 40 epochs. To enhance post-pruning performance, the Masked Knowledge Distillation (RepKD) method was employed during the recovery phase to transfer knowledge from teacher models to pruned student models. The RepKD approach aligns the output predictions and intermediate hidden states of the pruned and teacher models, with further details provided in the following section. Additionally, as Exponential Moving Averages (EMA) are updated and used during image generation, an excessively small learning rate can weaken EMA's effect, leading to suboptimal outcomes. To address this, a progressive learning rate scheduler was implemented to gradually halve the learning rate throughout training. The

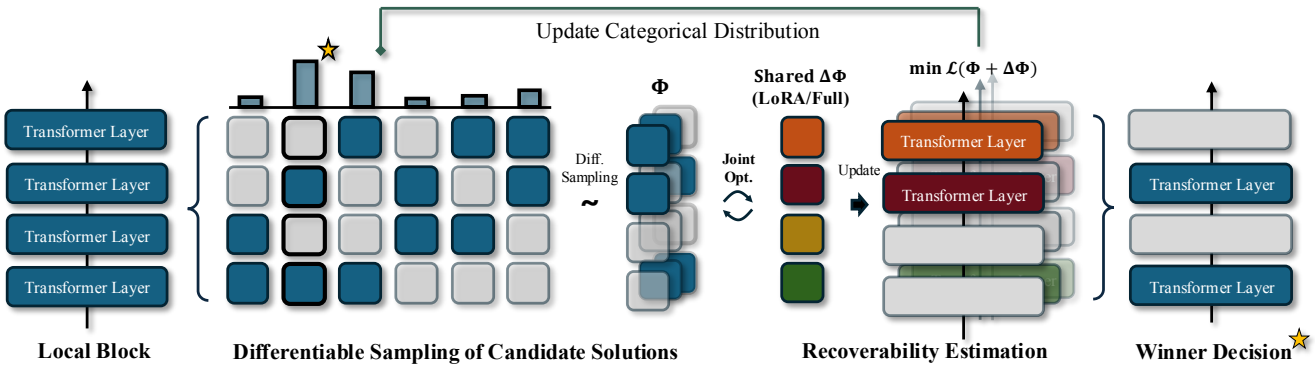

Figure 12. Learnable depth pruning on a local block

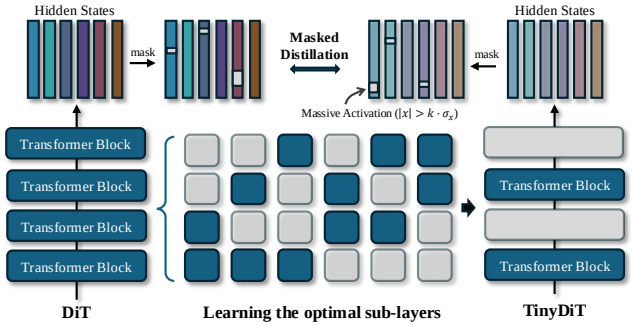

Figure 13. Masked knowledge distillation with 2:4 blocks.

details of each hyperparameter are provided in Table 6.

2. Visualization of Pruning Decisions

Figures 9, 10 and 11 visualize the dynamics of pruning decisions during training for the 1:2, 2:4, and 7:14 pruning schemes. Different divisions lead to varying search spaces, which in turn result in various solutions. For both the 1:2 and 2:4 schemes, good decisions can be learned in only one epoch, while the 7:14 scheme encounters optimization difficulty. This is due to the $\binom{14}{7}$=3,432 candidates, which is too huge and thus cannot be adequately sampled within a single epoch. Therefore, in practical applications, we use the 1:2 or 2:4 schemes for learnable layer pruning.

3. Details of Masked Knowledge Distillation

Training Loss. This work deploys a standard knowledge distillation to learn a good student model by mimicking the pre-trained teacher. The loss function is formalized as:

$$\mathcal{L} = \alpha_{\text{KD}} \cdot \mathcal{L}_{\text{KD}} + \alpha_{\text{Diff}} \cdot \mathcal{L}_{\text{Diff}} + \beta \cdot \mathcal{L}_{\text{Rep}} \tag{8}$$

Here, $\mathcal{L}$KD denotes the Mean Squared Error between the outputs of the student and teacher models. $\mathcal{L}$Diff represents the original pre-training loss function. Finally, $\mathcal{L}_{\text{Rep}}$ corresponds to the masked distillation loss applied to the hidden states, as illustrated in Figure 13, which encourages alignment between the intermediate representations of the pruned model and the original model. The corresponding hyperparameters α_{KD}, α_{Diff} and α_{Rep} can be found in Table 6.

Hidden State Alignment. The masked distillation loss $\mathcal{L}_{\text{Rep}}$ is critical for aligning the intermediate representations of the student and teacher models. During the recovery phase, each layer of the student model is designed to replicate the output hidden states of a corresponding two-layer local block from the teacher model. Depth pruning does not alter the internal dimensions of the layers, enabling direct alignment without additional projection layers. For models such as SiTs, where hidden state losses are more pronounced due to their unique interpolant-based architecture, a smaller coefficient β is applied to $\mathcal{L}_{\text{Rep}}$ to mitigate potential training instability. The gradual decrease in β throughout training further reduces the risk of negative impacts on convergence.

Iterative Pruning and Distillation. Table 7 assesses the effectiveness of iterative pruning and teacher selection strategies. To obtain a TinyDiT-D7, we can either directly prune a DiT-XL with 28 layers or craft a TinyDiT-D14 first and then iteratively produce the small models. To investigate the impact of teacher choice and the method for obtaining the initial weights of the student model, we derived the initial weights of TinyDiT-D7 by pruning both a pre-trained model and a crafted intermediate model. Subsequently, we used both the trained and crafted models as teachers for the pruned student models. Across four experimental settings, pruning and distilling using the crafted intermediate model yielded the best performance. Notably, models pruned from the crafted model outperformed those pruned from the pre-trained model regardless of the teacher model employed in the distillation process. We attribute this su-

Model	Optimizer	Cosine Sched.	Teacher	$\alpha_{\mathbf{KD}}$	$\alpha_{\mathbf{GT}}$	β	Grad. Clip	Pruning Configs
DiT-D19	AdamW(lr=2e-4, wd=0.0)	$\eta_{\min} = $ 1e-4	DiT-XL	0.9	0.1	1e-2 $\rightarrow$ 0	1.0	LoRA-1:2
DiT-D14	AdamW(lr=2e-4, wd=0.0	$\eta_{\min} = $ 1e-4	DiT-XL	0.9	0.1	1e-2 $\rightarrow$ 0	1.0	LoRA-1:2
DiT-D7	AdamW(lr=2e-4, wd=0.0)	$\eta_{\min} = $ 1e-4	DiT-D14	0.9	0.1	1e-2 $\rightarrow$ 0	1.0	LoRA-1:2
SiT-D14	AdamW(lr=2e-4, wd=0.0)	$\eta_{\min} = $ 1e-4	SiT-XL	0.9	0.1	2e-4 $\rightarrow$ 0	1.0	LoRA-1:2
MAR-D16	AdamW(lr=2e-4, wd=0.0)	$\eta_{\min} = $ 1e-4	MAR-Large	0.9	0.1	1e-2 $\rightarrow$ 0	1.0	LoRA-1:2

Table 6. Training details and hyper-parameters for mask training

Teacher Model	Pruned From	IS	FID	sFID	Prec.	Recall
DiT-XL/2	DiT-XL/2	29.46	56.18	26.03	0.43	0.51
DiT-XL/2	TinyDiT-D14	51.96	36.69	28.28	0.53	0.59
TinyDiT-D14	DiT-XL/2	28.30	58.73	29.53	0.41	0.50
TinyDiT-D14	TinyDiT-D14	57.97	32.47	26.05	0.55	0.60

Table 7. TinyDiT-D7 is pruned and distilled with different teacher models for 10k, sample steps is 64, original weights are used for sampling rather than EMA.

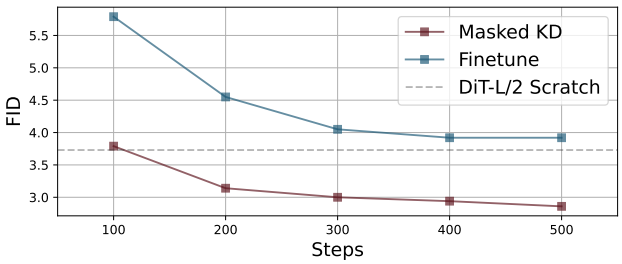

Figure 14. FID and training steps.

perior performance to two factors: first, the crafted model's structure is better adapted to knowledge distillation since it was trained using a distillation method; second, the reduced search space facilitates finding a more favorable initial state for the student model.

4. Analytical Experiments

Training Strategies Figure 14 illustrates the effectiveness of standard fine-tuning and knowledge distillation (KD), where we prune DiT-XL to 14 layers and then apply various fine-tuning methods. Figure 3 presents the FID scores across 100K to 500K steps. It is evident that the standard fine-tuning method allows TinyDiT-D14 to achieve performance comparable to DiT-L while offering faster inference. Additionally, we confirm the significant effectiveness of distillation, which enables the model to surpass DiT-L at just 100K steps and achieve better FID scores than the 500K standard fine-tuned TinyDiT-D14. This is because the distillation of hidden layers provides stronger supervision. Further increasing the training steps to 500K leads to significantly better results.

Learning Rate	IS	FID	sFID	Prec.	Recall
lr=2e-4	207.27	3.73	5.04	0.8127	0.5401
lr=1e-4	194.31	4.10	5.01	0.8053	0.5413
lr=5e-5	161.40	6.63	6.69	0.7419	0.5705

Table 8. The effect of Learning rato for TinyDiT-D14 finetuning w/o knowledge distillation

Learning Rate. We also search on some key hyperparameters such as learning rates in Table 8. We identify the effectiveness of lr=2e-4 and apply it to all models and experiments.

5. Visulization

Figure 15 and 16 showcase the generated images from TinySiT-D14 and TinyMAR-D16, which were compressed from the official checkpoints. These models were trained using only 7% and 10% of the original pre-training costs, respectively, and were distilled using the proposed masked knowledge distillation method. Despite compression, the models are capable of generating plausible results with only 50% of depth.

6. Limitations

In this work, we explore a learnable depth pruning method to accelerate diffusion transformer models for conditional image generation. As Diffusion Transformers have shown significant advancements in text-to-image generation, it is valuable to conduct a systematic analysis of the impact of layer removal within the text-to-image tasks. Additionally, there exist other interesting depth pruning strategies that need to be studied, such as more fine-grained pruning strategies that remove attention layers and MLP layers independently instead of removing entire transformer blocks. We leave these investigations for future work.

Figure 15. Generated images from TinySiT-D14

Figure 16. Generated images from TinyMAR-D16