TREAD: Token Routing for Efficient Architecture-agnostic Diffusion Training

–Supplementary Materials–

A. Implementation Details

A.1. Experimental Configuration

In contrast to DiT [48] and MDT [17], which leverage the ADM framework [8], our experimental approach is grounded in the formulation of EDM [32]. Specifically, we implement EDM's preconditioning through a σ-dependent skip connection, utilizing the standard parameter settings.

This approach eliminates the necessity to train ADM's noise covariance parameterization, as required by DiT. For the inference phase, we adopt the default temporal schedule defined by:

$$t_{i<N} = \left(t_{\text{max}}^{\frac{1}{\rho}} + \frac{i}{N-1} \left(t_{\text{min}}^{\frac{1}{\rho}} - t_{\text{max}}^{\frac{1}{\rho}} \right) \right)^{\rho}, \tag{6}$$

where the parameters are set to $N = 40$, $\rho = 7$, $t_{\text{max}} = 80$, and $t_{\text{min}} = 0.002$. Furthermore, we employ Heun's method as the ODE solver for the sampling process. This choice has been shown to achieve FID scores comparable to those obtained with 250 DDPM steps while significantly reducing the number of required steps [32, 75].

The noise distribution adheres to the EDM configuration, defined by:

$$\ln(p_\sigma) \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}), \tag{7}$$

with $P_{\text{mean}} = -1.2$ and $P_{\text{std}} = 1.2$. For detailed information, refer to the EDM paper [32].

A.2. Network Details

Parameter Comparison As previously discussed, our method does not require any modifications to the architecture itself, whereas other methods incorporate a predefined decoder head on top of the standard DiT structure. This introduces computational overhead that is particularly noticeable in smaller models. Since our method does not need these additional parameters, we reduce the computational cost associated with the decoder head. This is demonstrated in Table S7.

Model	# of Parameters (Millions)
DiT	675
DiT$_{\text{+TREAD}}$	675
MaskDiT	730
SD-DiT	740

Table S7. Comparison of the number of network parameters. MaskDiT and SD-DiT add a substantial number of parameters, approximately 10% of those in XL-sized DiT models. This additional parameter count is fixed across different model sizes [75, 76], which can slow down smaller models since the relative size of the added decoder components increases.

Diffusion-RWKV Setting. Due to the nature of RWKV and other state-space models (SSMs) [2, 19, 30], a row selection strategy is applied instead of a random selection. Additionally, we adhere to the DiT configuration in the RWKV setting. Nevertheless, we are able to improve upon our own Diffusion-RWKV [14] baseline using TREAD. The poor performance of our RWKV baseline can be attributed to the number of layers; our model consists of only 12 layers, whereas Fei et al. [14] recommends using 25 or even 49 layers.

Mixed-Precision. TREAD can be used successfully with `bf16`. However, it is noteworthy that when less computational blocks are available towards the end (like 1-3) might run into instabilities during training. We were able to mitigate this by keeping L_j in `fp32` during training when using a route $\mathbf{r}_{i \to j}$. The effect on iteration speed is minimal.

A.3. Hyperparameters

Throughout all of our experiments we use the same structure as DiT [48]. We use AdamW [38] and a constant learning rate of 1e-4, $(\beta_1, \beta_2) = (0.9, 0.999)$ and no weight decay. Furthermore, we train in `bf16`, precompute the data into lates using the Stable Diffusion VAE [54]. We use the `stabilityai/sd-vae-ft-ema` VAE checkpoint from huggingface.

	DiT-S	DiT-B	DiT-L	DiT-XL
Optimization				
Batch size	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW
LR	1e-4	1e-4	1e-4	1e-4
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Optimization - Finetune				
Batch size	-	-	-	1,024
Optimizer	-	-	-	AdamW
LR	-	-	-	1e-5
(β_1, β_2)	-	-	-	(0.9, 0.999)
Architecture				
Dim	384	768	1,024	1,152
Heads	6	12	16	16
Layers	12	12	24	28
TREAD				
Route	$\mathbf{r}_{2\to8}$	$\mathbf{r}_{2\to8}$	$\mathbf{r}_{2\to20}$	$\mathbf{r}_{2\to24}$
Selection Rate	0.5	0.5	0.5	0.5

Table S8. Hyperparameter setup for DiT variants.

A.4. Classifier-Free Guidance

TREAD does demonstrate superior performance for both unguided as well as guided generation on a DiT-B/2 as shown in Figure S9.

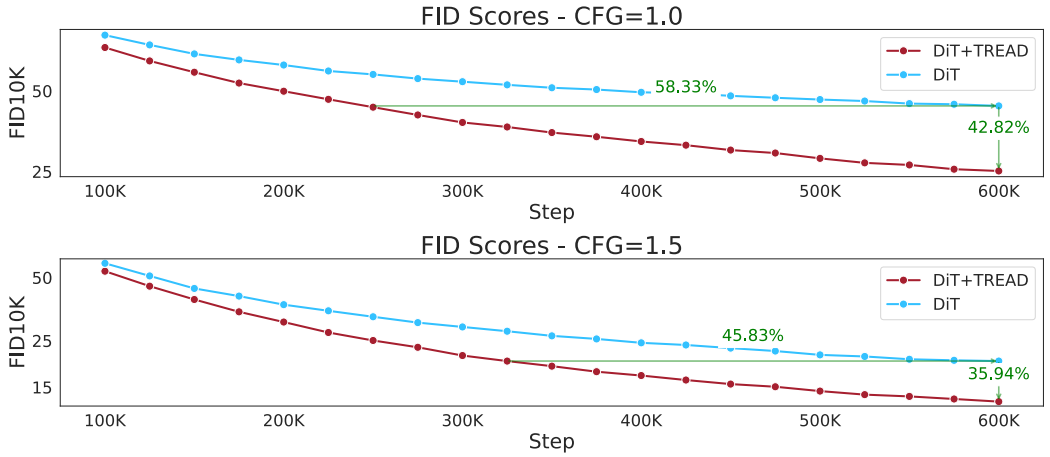

Figure S9. We compare FID@10K between a DiT-B/2 and DiT-B/2$_{+\textbf{TREAD}}$ with and without Classifier-free Guidance (CFG). TREAD outperforms the standard DiT approach, even without the need to finetune without routing.

B. Loss Curves and Routing induced Loss Gap

We provide loss curves in Figure S10 for better understanding of the interaction between loss, route length and FID. The loss gap to the baseline DiT can be explained using the increased difficulty during training which is induced by longer routes.

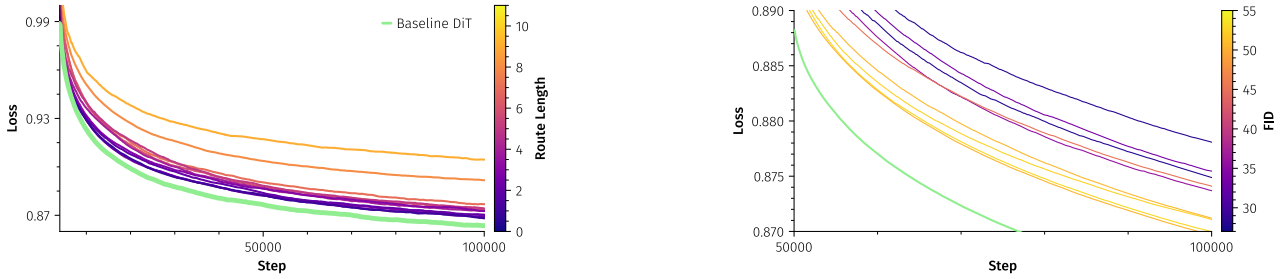

Figure S10. **The impact of the routing mechanism on the model can be estimated with a loss difference.** We provide loss curves between 0 to 100K iterations against route length (left) and a zoomed-in version against FID (right). It can be seen that route length correlates with increased loss difference from baseline as well as with final FID.

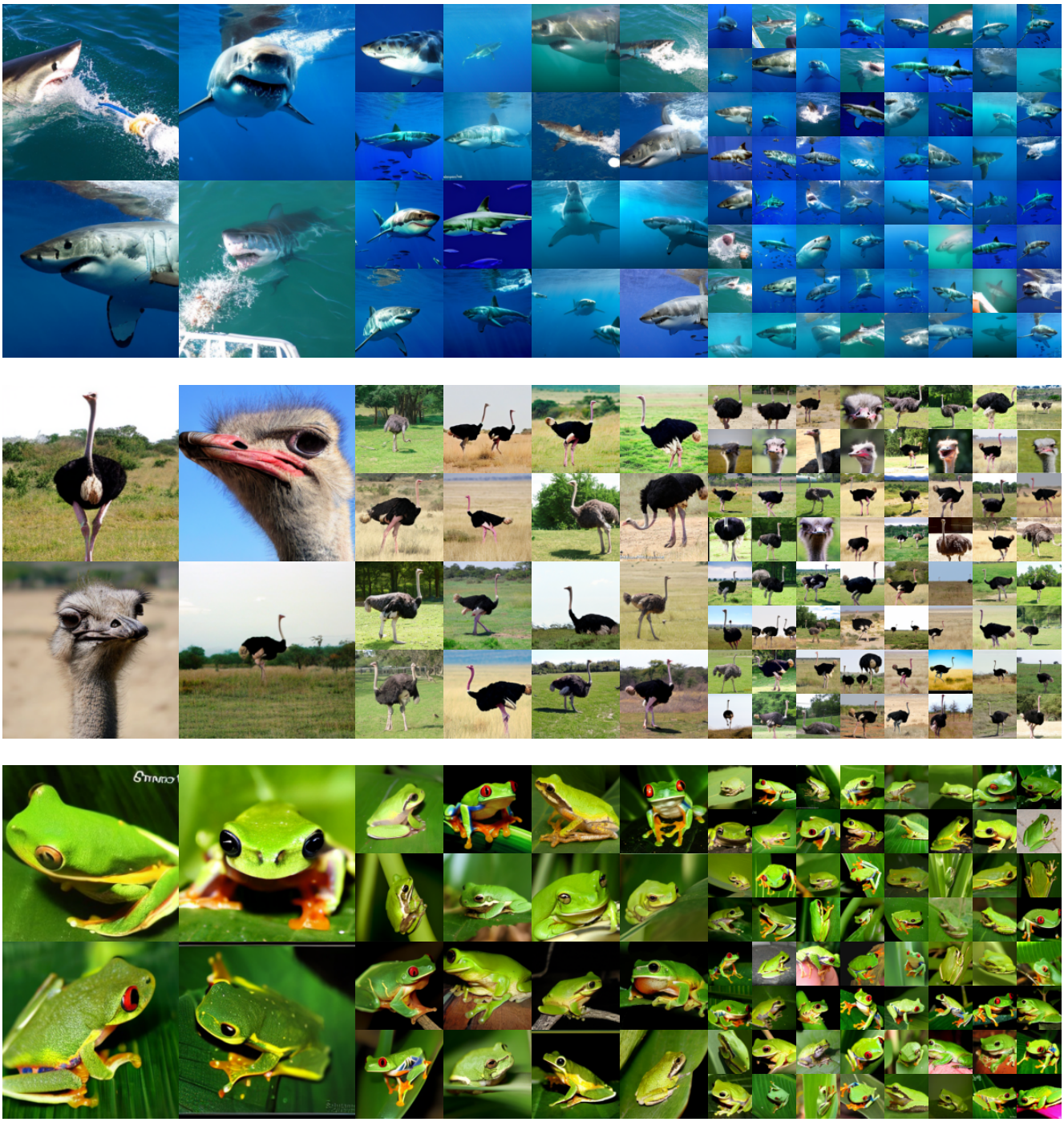

Figure S11. **Uncurated 256 × 256 samples** from DiT-XL/2$_{+\text{TREAD}}$ (F) with $\omega = 3.5$.

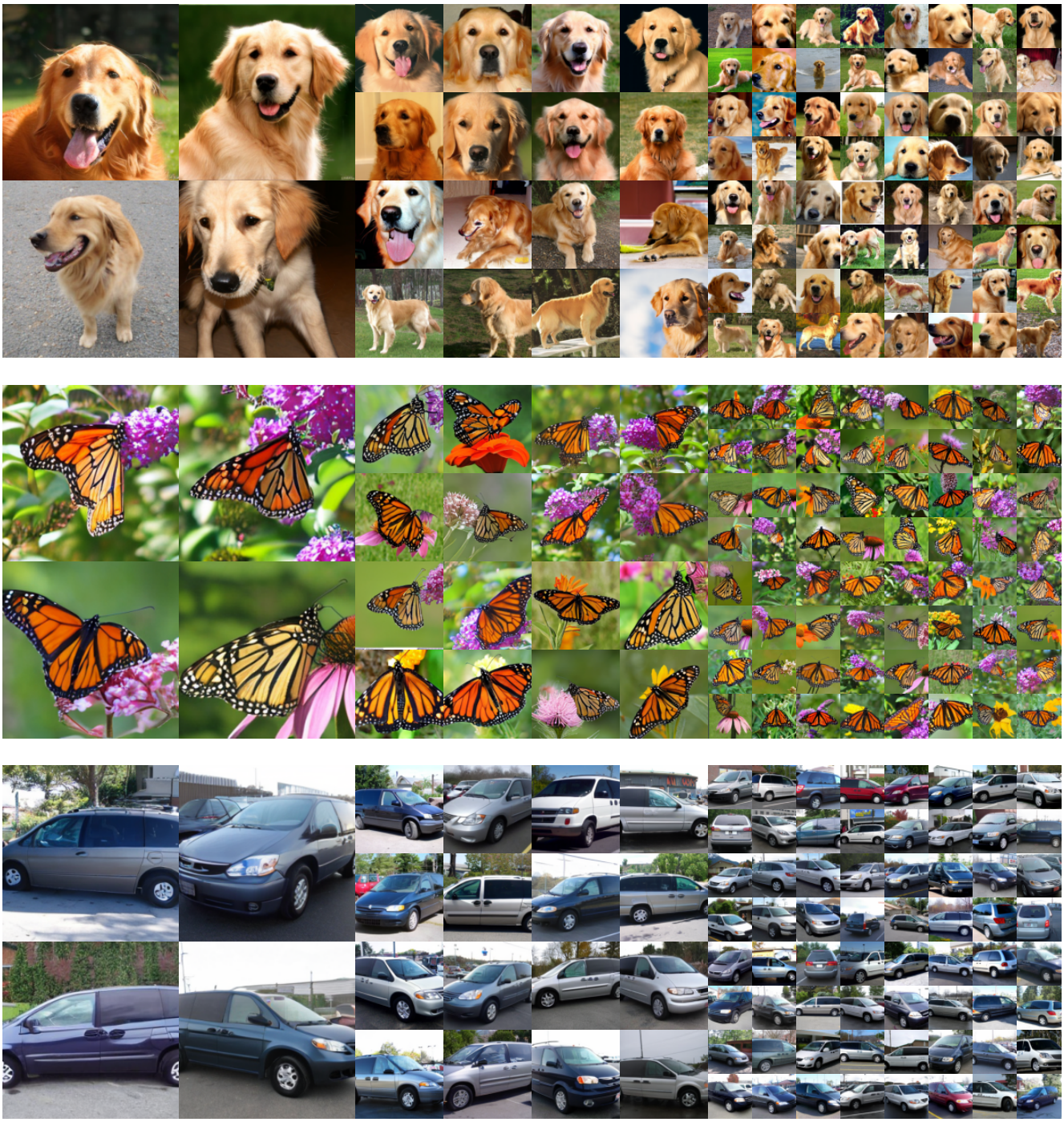

Figure S12. **Uncurated 256 × 256 samples** from DiT-XL/2$_{+\text{TREAD}}$ (F) with $\omega = 3.5$.

Figure S13. **Uncurated 256 × 256 samples** from DiT-XL/2$_{+\text{TREAD}}$ (F) with $\omega = 3.5$.

Figure S14. **Uncurated 256× 256 samples** from DiT-XL/2$_{+\text{TREAD}}$ (F) with $\omega = 3.5$.

Figure S15. **Uncurated 256 × 256 samples** from DiT-XL/2$_{+\text{TREAD}}$ (F) with $\omega = 3.5$.

Figure S16. **Uncurated 256 × 256 samples** from DiT-XL/2$_{+\text{TREAD}}$ (F) with $\omega = 3.5$.