ViT^3: Unlocking Test-Time Training in Vision

Supplementary Material

1. Contribution and Limitation

In this paper, we systematically study the design space of Test-Time Training (TTT), shedding some light on the design principles for effective visual TTT and possible future directions. Our main contributions are as follows:

- We present a systematic empirical study of Test-Time Training designs for vision, covering inner training regimes (loss function, learning rate, batch size, epochs) and inner model design (architecture and model size).
- We offer six practical insights for building effective yet efficient TTT module, providing detailed analyses of the TTT design space. Our analyses also reveal several valuable future research directions for TTT models.
- We build the Vision Test-Time Training (ViT^3) model, a simple TTT architecture that implements these insights. With $\mathcal{O}(N)$ complexity, ViT^3 achieves competitive results across image classification, image generation, object detection, and semantic segmentation, serving as a strong baseline for future research on visual TTT models.

However, we note that there are other design choices that we have not covered (e.g., inner optimizer, inner data augmentation, Transformer inner model, etc.), and this paper is not exhaustive. Exploring these axes is left to future work.

2. Inner Training Loss

Consider a mini-batch of target value tokens and model predictions $V_{\mathcal{B}}, \hat{V}_{\mathcal{B}} \in \mathbb{R}^{B\times d}$, where B denotes the batch size. We denote the i-th token (row) by $V_i, \hat{V}_i \in \mathbb{R}^{1\times d}$.

For each loss function considered in Tab. 1 of the main paper, we provide the explicit formula and compute the mixed second derivative $\frac{\partial^2 \mathcal{L}}{\partial V_{ij}\partial \hat{V}_{ij}}$.

(1) Dot Product Loss.

$$\mathcal{L} = -\frac{1}{B\sqrt{d}}\sum_{i=1}^{B}\hat{V}_i V_i^\top. \tag{1}$$

The mixed second derivative is:

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial V_{ij}\partial \hat{V}_{ij}} &= \frac{\partial}{\partial V_{ij}}\left(\frac{\partial \mathcal{L}}{\partial \hat{V}_{ij}}\right)\\ &= \frac{\partial}{\partial V_{ij}}\left(-\frac{1}{B\sqrt{d}}V_{ij}\right)\\ &= -\frac{1}{B\sqrt{d}}.\end{aligned} \tag{2}$$

(2) MSE (L2) Loss.

$$\mathcal{L} = \frac{1}{2B\sqrt{d}}\sum_{i=1}^{B}(\hat{V}_i - V_i)(\hat{V}_i - V_i)^\top. \tag{3}$$

The mixed second derivative is:

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial V_{ij}\partial \hat{V}_{ij}} &= \frac{\partial}{\partial V_{ij}}\left(\frac{\partial \mathcal{L}}{\partial \hat{V}_{ij}}\right)\\ &= \frac{\partial}{\partial V_{ij}}\left(\frac{1}{B\sqrt{d}}(\hat{V}_{ij} - V_{ij})\right)\\ &= -\frac{1}{B\sqrt{d}}.\end{aligned} \tag{4}$$

(3) RMSE Loss.

$$\mathcal{L} = \sqrt{\frac{1}{B\sqrt{d}}\sum_{i=1}^{B}(\hat{V}_i - V_i)(\hat{V}_i - V_i)^\top}. \tag{5}$$

The mixed second derivative is:

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial V_{ij}\partial \hat{V}_{ij}} &= \frac{\partial}{\partial V_{ij}}\left(\frac{\partial \mathcal{L}}{\partial \hat{V}_{ij}}\right)\\ &= \frac{\partial}{\partial V_{ij}}\left(\frac{1}{B\sqrt{d}\sqrt{S}}(\hat{V}_{ij} - V_{ij})\right)\\ &= -\frac{1}{B\sqrt{d}\sqrt{S}} + \frac{1}{B^2 d S^{3/2}}(\hat{V}_{ij} - V_{ij})^2,\\ S &= \frac{1}{B\sqrt{d}}\sum_{i=1}^{B}(\hat{V}_i - V_i)(\hat{V}_i - V_i)^\top.\end{aligned} \tag{6}$$

(4) MAE (L1) Loss.

$$\mathcal{L} = \frac{1}{B\sqrt{d}}\sum_{i=1}^{B}\|\hat{V}_i - V_i\|_1, \tag{7}$$

The mixed second derivative is:

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial V_{ij}\partial \hat{V}_{ij}} &= \frac{\partial}{\partial V_{ij}}\left(\frac{\partial \mathcal{L}}{\partial \hat{V}_{ij}}\right)\\ &= \frac{\partial}{\partial V_{ij}}\left(\frac{1}{B\sqrt{d}}\operatorname{sign}(\hat{V}_{ij} - V_{ij})\right)\\ &= 0 \quad \text{if } \hat{V}_{ij} \neq V_{ij}.\end{aligned} \tag{8}$$

(5) Smooth L1 loss.

$$\begin{aligned}\mathcal{L} &= \frac{1}{B\sqrt{d}}\sum_{i=1}^{B}\sum_{j=1}^{d}\ell(\hat{V}_{ij} - V_{ij}),\\ \ell(x) &= \begin{cases}\frac{1}{2}x^2 & \text{if } |x| < 1\\ |x| - \frac{1}{2} & \text{otherwise}\end{cases}.\end{aligned} \tag{9}$$

The mixed second derivative is:

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial V_{ij} \partial \hat{V}_{ij}} &= \frac{\partial}{\partial V_{ij}} \left(\frac{\partial \mathcal{L}}{\partial \hat{V}_{ij}} \right) \\ &= \frac{\partial}{\partial V_{ij}} \left(\frac{1}{B\sqrt{d}} \ell'(\hat{V}_{ij} - V_{ij}) \right) \\ &= -\frac{1}{B\sqrt{d}} \ell''(\hat{V}_{ij} - V_{ij}) \\ &= -\frac{1}{B\sqrt{d}} \times \begin{cases} 1 & \text{if } |\hat{V}_{ij} - V_{ij}| < 1 \\ 0 & \text{if } |\hat{V}_{ij} - V_{ij}| > 1 \end{cases}.\end{aligned} \tag{10}$$

Notably, the $1/\sqrt{d}$ scaling used above is consistent with the scaled dot product attention convention [7]. As analyzed in the main paper, losses with vanishing (or piecewise-vanishing) mixed second derivatives — in particular MAE (almost everywhere zero) and Smooth L1 in its linear region — hinder the learning of outer model parameter W_V matrix and therefore leads to lower performance.

3. Model Architecture

As discussed in the main paper, we present a plug-in visual TTT block based on our findings. Specifically, for inner training, we use a single epoch of full-batch gradient descent with learning rate 1.0, optimizing a dot-product loss. The inner model comprises a simplified gated linear unit $\mathcal{F}_1 = \mathrm{FC}(x) \odot \mathrm{SiLU}(\mathrm{FC}(x))$ and a depthwise convolution $\mathcal{F}_2 = \mathrm{DWConv}(x)$. The gated linear unit doubles the capacity of a naive $d \times d$ linear state while remaining easy to optimize, whereas the depthwise convolution offers a natural integration of local and global information. Within each TTT block, we use $\mathcal{F}_2$ in a single attention head and instantiate the remaining heads with $\mathcal{F}_1$.

We build two model families with this TTT block, ViT^3 (non-hierarchical) and H-ViT^3 (hierarchical 4-stage), and adapt our approach to diffusion image Transformers (DiT) [6] for generative tasks. The architectures are provided in Tab. 1, Tab. 2, and Tab. 3. To introduce positional information, we employ conditional positional encodings [1], which is widely adopted by modern vision Transformers [2, 8, 9], linear attention [3] and Mamba models [4, 5, 10]. Since our method benefits from $\mathcal{O}(N)$ complexity, we directly process the high-resolution feature map with a global receptive field.

	ViT^3-T	ViT^3-S	ViT^3-B
Backbone	Patch $\downarrow 16$ $\mathrm{B}(192, 6) \times 12$	Patch $\downarrow 16$ $\mathrm{B}(384, 6) \times 12$	Patch $\downarrow 16$ $\mathrm{B}(768, 12) \times 12$
Classifier	Global Average Pooling, Linear		

Table 1. Architectures of ViT^3 model series. Patch "$\downarrow n$" indicates the patch size is n. "$\mathrm{B}(C, H)$" represents one building block with embedding dimension C and H attention heads.

	Size	H-ViT^3-T	H-ViT^3-S	H-ViT^3-B
Stage1	$\frac{H}{4} \times \frac{W}{4}$	Stem $\downarrow 4$ $\mathrm{B}(64, 2) \times 1$	Stem $\downarrow 4$ $\mathrm{B}(64, 2) \times 2$	Stem $\downarrow 4$ $\mathrm{B}(96, 3) \times 2$
Stage2	$\frac{H}{8} \times \frac{W}{8}$	Down $\downarrow 2$ $\mathrm{B}(128, 4) \times 3$	Down $\downarrow 2$ $\mathrm{B}(128, 4) \times 6$	Down $\downarrow 2$ $\mathrm{B}(192, 6) \times 6$
Stage3	$\frac{H}{16} \times \frac{W}{16}$	Down $\downarrow 2$ $\mathrm{B}(320, 10) \times 9$	Down $\downarrow 2$ $\mathrm{B}(320, 10) \times 18$	Down $\downarrow 2$ $\mathrm{B}(448, 14) \times 18$
Stage4	$\frac{H}{32} \times \frac{W}{32}$	Down $\downarrow 2$ $\mathrm{B}(512, 16) \times 4$	Down $\downarrow 2$ $\mathrm{B}(512, 16) \times 8$	Down $\downarrow 2$ $\mathrm{B}(640, 20) \times 8$
Classifier		Global Average Pooling, Linear		

Table 2. Architectures of H-ViT^3 model series. "$\downarrow n$" indicates the downsampling ratio is n. "$\mathrm{B}(C, H)$" represents one building block with embedding dimension C and H attention heads.

	DiT^3-S/8	DiT^3-S/4	DiT^3-S/2
Backbone	Patch $\downarrow 8$ $\mathrm{B}(384, 6) \times 12$	Patch $\downarrow 4$ $\mathrm{B}(384, 6) \times 12$	Patch $\downarrow 2$ $\mathrm{B}(384, 6) \times 12$
	DiT^3-B/8	DiT^3-B/4	DiT^3-B/2
Backbone	Patch $\downarrow 8$ $\mathrm{B}(768, 12) \times 12$	Patch $\downarrow 4$ $\mathrm{B}(768, 12) \times 12$	Patch $\downarrow 2$ $\mathrm{B}(768, 12) \times 12$

Table 3. Architectures of DiT^3 model series. Patch "$\downarrow n$" indicates the patch size is n. "$\mathrm{B}(C, H)$" represents one building block with embedding dimension C and H attention heads.

References

[1] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *ICLR*, 2023. 2

[2] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, 2022. 2

[3] Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify mamba in vision: A linear attention perspective. In *NeurIPS*, 2024. 2

[4] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. In *ECCVW*, 2024. 2

[5] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. In *NeurIPS*, 2024. 2

[6] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-

reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[8] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Dat++: Spatially dynamic vision transformer with deformable attention. *arXiv preprint arXiv:2309.01430*, 2023. 2

[9] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson WH Lau. Biformer: Vision transformer with bi-level routing attention. In *CVPR*, 2023. 2

[10] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: efficient visual representation learning with bidirectional state space model. In *ICML*, 2024. 2