ThinkingViT: Matryoshka Thinking Vision Transformer for Elastic Inference

Supplementary Material

A. Adaptive inference performance of *ThinkingViT* under varying entropy thresholds

Table 3 presents detailed results of *ThinkingViT*'s adaptive inference behavior under different entropy thresholds. We report accuracy, number of parameters, throughput, total compute (in GMACs), and the ratio of inputs that proceed to the second stage of inference of *ThinkingViT* 3H $\rightarrow$ 6H.

B. Results on ImageNet variants

Figure 11 and 12 show the performance of *ThinkingViT* on ImageNet variants. Note that the GMACs for ImageNet-V2 and ImageNet-R are reported in Figure 4b and Figure 4c, respectively.

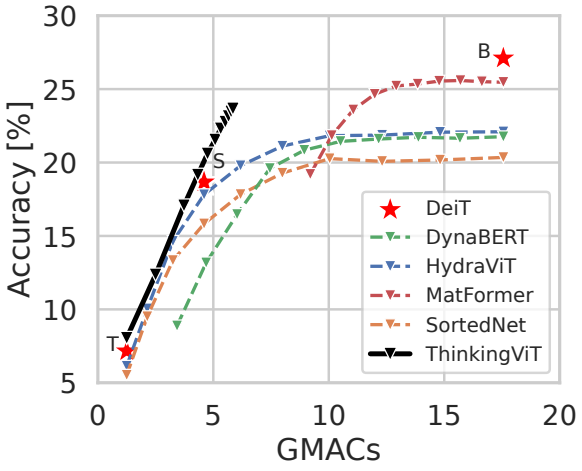

Figure 11. Full results of *ThinkingViT* and baselines in terms of GMACs on ImageNet-A.

C. Comparison with BranchyNet

Table 2 compares *ThinkingViT* with BranchyNet, a standard early exit model [41], applied to DeiT [43] that matches its GMACs and throughput. The baseline consists of 24 layers: the first 12 use 3 attention heads (3H) and the remaining 12 use 6 heads (6H). Similar to *ThinkingViT*, the exits are jointly trained together. At the 3H point, *ThinkingViT* records slightly lower accuracy because the first three heads must generate representations that remain compatible with the later expansion to six heads. Additionally, their weights must be trained in a way that allows the weights of the subsequent three heads to build upon them.

After expanding to 6 attention heads, *ThinkingViT* reaches 81.44% top-1 accuracy, improving by 3.37 p.p. upon the early exit baseline, which attains 78.08%. This improvement likely stems from *ThinkingViT* increasing the number of active attention heads within a fixed 12-layer backbone, allowing gradients to pass through fewer layers (12 layers compared to 24 layers) and reducing some of the optimization challenges associated with deeper networks. These findings indicate that allocating compute along the width dimension by gradually increasing attention heads can be more effective than depth-based early exit strategies used in prior work. Further, since it adjusts model width rather than depth, *ThinkingViT* is complementary to early-exit strategies and can be combined with them to further expand deployment flexibility.

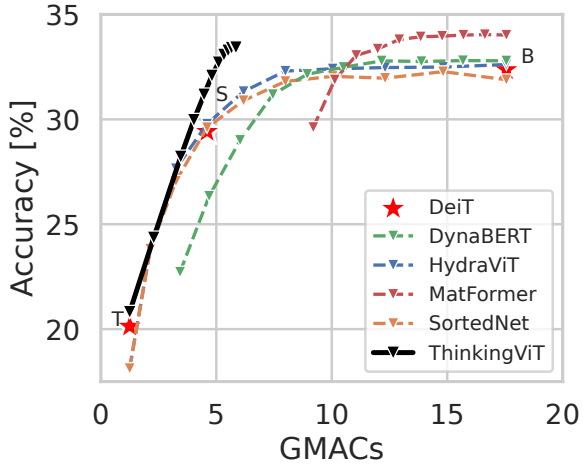

Figure 12. Full results of *ThinkingViT* and baselines in terms of GMACs on ImageNet-Sketch.

D. Ablation of different token recycling strategies

In Table 4, we compare several design variants for recycling information from the first round of inference to get better performance in the second round. We conduct these experiments on *ThinkingViT* 3H $\rightarrow$ 6H. In the *Layerwise Activation Snapshots* variant, we cache the hidden states at each of the 12 layers from the first round and feed them into the corresponding layers in the second round. However, this approach performs suboptimally, likely because it forces all intermediate representations from the first round to be reused in the second round, which makes it harder for the tokens as they need to work for both rounds. In the *Memory Tokens* design, we introduce a set of learnable memory tokens [7] that, in the first round, store helpful information for the

Table 2. Comparison of *ThinkingViT* and BranchyNet on DeiT.

Model	GMACs	Throughput [#]	Params [M]	Accuracy [%]
DeiT-Tiny	1.25	10047.6	5.7	72.2
DeiT-Small	4.6	4603.6	22.1	79.9
BranchyNet 3H	1.25	10047.6	27.8	74.51
BranchyNet 3H → 6H	5.85	3157.1	27.8	78.08
***ThinkingViT* 3H**	1.25	10047.6	22.1	73.58
***ThinkingViT* 3H → 6H**	5.85	3157.1	22.1	**81.44**

Table 3. Performance metrics of *ThinkingViT* 3H → 6H across different entropy thresholds

Entropy Threshold	Accuracy	Throughput [#/s]	Params [M]	GMACs	Second Round Call Ratio [%]
0	81.444	3157.09	22.01	5.85	100.0
0.1	81.440	3347.69	22.01	5.47	91.7
0.3	81.438	3955.05	22.01	4.50	70.58
0.5	81.386	4380.71	22.01	3.98	59.29
0.7	81.230	4807.04	22.01	3.55	49.95
0.9	80.714	5342.47	22.01	3.11	40.36
1.1	79.990	5918.90	22.01	2.72	31.97
1.3	79.114	6535.13	22.01	2.38	24.63
1.5	77.936	7201.46	22.01	2.08	18.11
1.7	76.766	7944.38	22.01	1.81	12.13
2	74.736	9203.90	22.01	1.44	4.20
2.5	73.580	10047.60	22.01	1.25	0.0

second round. We also experiment with hybrid strategies that combine activation snapshots with Memory Tokens or introduce a fresh `[CLS]` token in the second round. We additionally evaluated the use of KV-cache reuse [45], originally introduced for generative models, and include the results in Table 4. Ultimately, we find that the simplest strategy, *Final-Layer Token Recycling*, which reuses the features from the last layer of the first round, on average, achieves the best performance. This indicates that the output of the first round already captures sufficient high-level information to guide the second round effectively, while being comparatively simple to implement. It is important to note that except the KV-cache experiment, all results were conducted during the early experimental phase of *ThinkingViT*, and due to training resource constraints, they were not trained with the full joint training strategy described in Section 3.3, but rather with the stochastic training method introduced in [14]. As a result, their accuracies are slightly lower than those of the final model reported in Section 4.

E. Ablation of different fusing strategies

Table 5 compares six fusion strategies that recycle the tokens produced in the first round, denoted as z_1, by incorporating them into the initial patch embeddings of the second round, $\mathcal{E}_2$, on *ThinkingViT* with a configuration of $3H \rightarrow 6H$. Each variant forms the second-round input as a linear blend, where α is a learnable scalar. The strategies differ in how the two embeddings are projected to the same dimensional space. We consider two projection strategies: (I) a parameter-free approach that either repeats the embedding or pads the lower-dimensional embeddings with zeros, and (II) a learnable linear projection. Empirically, the learnable projection achieves the best overall accuracy by having the highest accuracy in the second round while maintaining performance comparable to the strongest model configuration in the first round. In addition, we evaluate initializing $\alpha = 1$ and observe that initializing with zero consistently leads to the best performance. This initialization allows the model to begin training without relying on information from the first round, thereby enabling it to learn the optimal degree of reuse. For example, in *ThinkingViT* configured with 3H → 6H heads, the learned value of α converges to -0.19.

F. Prediction dynamics across two inference rounds on ImageNet-1K

Figure 13 presents prediction dynamics across two inference rounds of *ThinkingViT* 3H → 6H on the ImageNet-1K validation set. A small portion of samples, approximately 2%, were correctly classified in the first round but misclassified

Table 4. Comparison of design variants for conditioning the second round of inference on the first.

Design Variant	First Thought Acc. [%]	Second Thought Acc. [%]
DeiT Baseline (Tiny → Small)	72.2	79.9
Layerwise Activation Snapshots	73.794	78.566
Memory Tokens	73.96	78.9
Layerwise Activation Snapshots + new `[CLS]` in second round	73.58	77.94
Layerwise Activation Snapshots + Memory Tokens	73.71	79.04
KV-Caching [45]	73.872	75.674
Final-Layer Token Recycling (*ThinkingViT*)	73.13	**80.05**

Table 5. Impact of different fusion strategies for integrating first-round tokens (z_1) into second-round embeddings ($\mathcal{E}_2$) during progressive inference on *ThinkingViT* with 3H → 6H.

Fusion Method	Dim Alignment	Note	Acc. [%] 1st Round	Acc. [%] 2nd Round
$\alpha \cdot z_1 + \mathcal{E}_2$	Pad z_1 with zeros	Pad the first half, $init(\alpha) = 0$	73.32%	81.37%
$\alpha \cdot z_1 + \mathcal{E}_2$	Pad z_1 with zeros	Pad the second half, $init(\alpha) = 0$	**74.12%**	80.32%
$\alpha \cdot z_1 + \mathcal{E}_2$	Repeat z_1	$init(\alpha) = 1$	72.99%	80.87%
$\alpha \cdot z_1 + \mathcal{E}_2$	Repeat z_1	$init(\alpha) = 0$	73.14%	81.41%
$z_1 + \alpha \cdot \mathcal{E}_2$	Repeat z_1	$init(\alpha) = 0$	72.89%	80.51%
$\alpha \cdot z_1 + \mathcal{E}_2$	Linear (*ThinkingViT*)	$init(\alpha) = 0$	73.58%	**81.44%**

in the second. This phenomenon, often attributed to *overthinking* [26], is also observed in other architectures such as ResNet and Swin Transformers [9], where smaller models sometimes outperform larger ones on few samples by avoiding unnecessary complexity. The majority of samples, around 70%, were correctly classified in both rounds. These generally correspond to visually simple or unambiguous cases where one round of inference is sufficient. Roughly 10% of samples were initially misclassified but corrected in the second round, demonstrating the benefit of additional reasoning for harder examples. Finally, around 16% remained misclassified across both rounds, indicating that these inputs are intrinsically ambiguous or fall outside the model's capacity, even with increased computation.

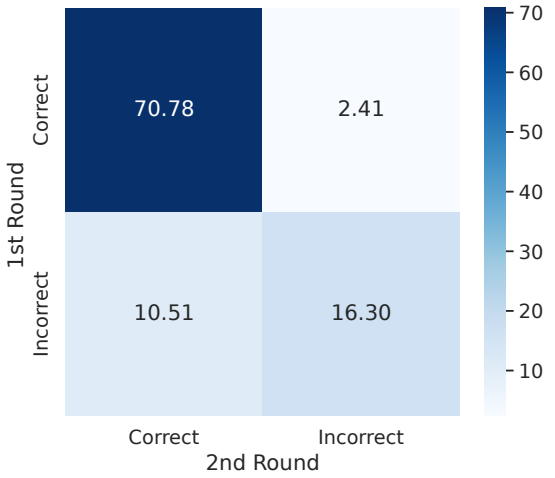

Figure 13. Prediction dynamics of *ThinkingViT* 3H → 6H across two inference rounds on ImageNet-1K.

G. Comparing and combining *ThinkingViT* with other adaptive baselines

ThinkingViT builds upon a nested backbone to enable input adaptivity, so our evaluation in Section 4 focuses on nested Transformer baselines. For completeness, we also evaluate against several representative token-level dynamic models, including MoD [35], AMoD [12], ToMe [3], A-ViT [53], AdaViT [30], and DynamicViT [34]. As shown in Figure 14, *ThinkingViT* achieves greater scalability and consistently better GMACs-Accuracy tradeoffs.

We note that *ThinkingViT* performs routing at the *image level*, which is orthogonal to token-level pruning [15, 47, 48], and thus can be combined with such approaches to provide additional flexibility. To demonstrate this, we incorporate the token pruning mechanism of DynamicViT [34] into the second round of *ThinkingViT* 3H → 6H, applying a pruning ratio of 0.8. As shown in Fig. 14, integrating DynamicViT further improves both efficiency and accuracy, indicating that *ThinkingViT* 3H → 6H is compatible with and complementary to existing token pruning approaches.

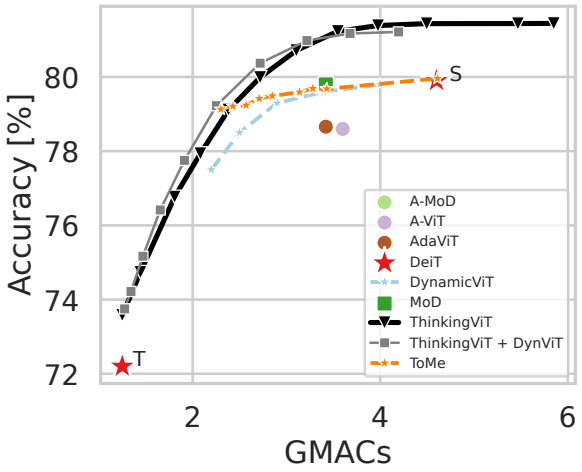

Figure 14. Comparing and combining *ThinkingViT* with token pruning baselines.

H. Comparison of accuracy vs. GMACs for baselines based on DeiT-Small

In Figure 15, we compare accuracy versus GMACs on ImageNet-1K using DeiT-Small as the common backbone for all baselines. *ThinkingViT* achieves higher accuracy at similar or lower compute budgets, showing a consistently more favorable tradeoff. This demonstrates that the benefits of *ThinkingViT* hold across backbone scales.

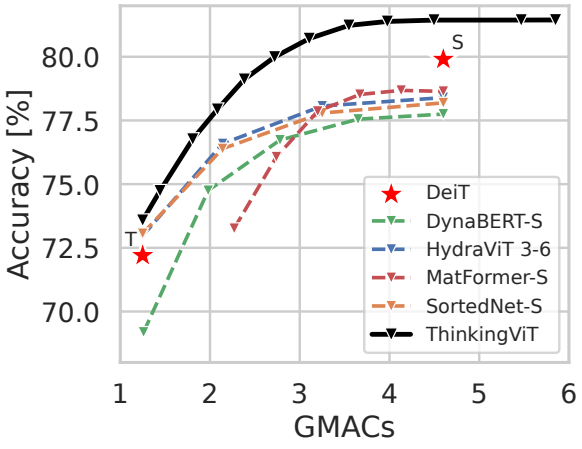

Figure 15. Accuracy versus GMACs on the ImageNet-1K validation set. All baseline models are based on DeiT-Small. *ThinkingViT* consistently outperforms these baselines by achieving higher accuracy at comparable or lower computational cost.

I. ImageNet-21K results

To evaluate *ThinkingViT* on a larger-scale classification task, we train *ThinkingViT*, MatFormer, HydraViT, DeiT-Tiny, and DeiT-Small on ImageNet-21K [38] as follows:

We use the shuffled dataset `imagenet-w21-wds` [1] with the first 12,741,248 images as the training split and the remaining 411,008 images as the validation split. All models are trained from scratch for 150 epochs with a global batch size of 2048.

Figure 16 reports accuracy versus GMACs on the validation split. In the low-compute regime, i.e., below 3.5 GMACs, all nested methods perform comparably. However, as the compute budget increases, *ThinkingViT* pulls ahead: at roughly 4.6 GMACs, *ThinkingViT* reaches 41.9% accuracy, outperforming both MatFormer at 39.8% and HydraViT at 39.4% by more than 2 percentage points. Moreover, *ThinkingViT* offers a higher accuracy ceiling of 42.4% by leveraging its second inference round for difficult samples, matching standalone DeiT-Small at 42.3% while providing a continuous range of cheaper operating points through entropy-based early exit.

These results show that the benefits of *ThinkingViT* generalize beyond ImageNet-1K to large-scale settings with an order of magnitude more classes.

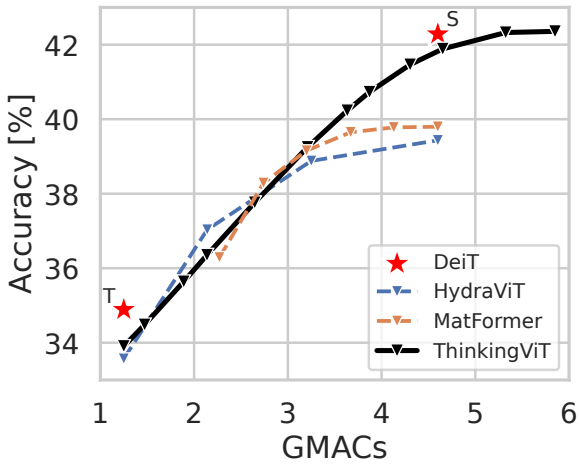

Figure 16. Accuracy versus GMACs on ImageNet-21K. All models are trained from scratch for 150 epochs. While nested methods perform similarly below 3.5 GMACS, *ThinkingViT* achieves a higher accuracy ceiling. Stars denote standalone DeiT-Tiny (T) and DeiT-Small (S).

J. Limitations

Training Overhead: Compared to training multiple standalone models, nested models like *ThinkingViT* incur higher training costs to reach similar performance levels across all stages. This is a known limitation of nested architectures, which share parameters and require joint optimiza-

[1] https://huggingface.co/datasets/timm/imagenet-w21-wds

tion to maintain accuracy at varying compute levels. In our case, *ThinkingViT* 3→6 trains DeiT-T and DeiT-S with a two-stage forward and single backward pass, taking 20 min/epoch, comparable to training DeiT-T (7 min) and DeiT-S (13 min) separately. Therefore training *ThinkingViT* takes as long as training DeiT-T and DeiT-S for 300 epochs each.

Jumping Too Far Reduces Effectiveness: When the model transitions from a very small subset of attention heads (e.g., 3H) directly to a full configuration (e.g., 12H) in the second stage, performance suffers compared to using an intermediate stage (e.g., 9H) instead. This suggests that overly aggressive compute expansion can disrupt representational continuity, emphasizing the importance of smooth progression in staged inference.

K. Batch inference

After each "thinking" round, we filter out the samples whose entropy falls below the stopping threshold (i.e., confident predictions). The remaining uncertain samples are then rebatched and forwarded to the next round. Since all rounds share the same Transformer backbone and Token Recycling only adjusts the input embeddings, no additional batch scheduling is needed. As a result, *ThinkingViT* integrates smoothly into standard batched inference engines, where the model proceeds in synchronized rounds and the batch size gradually decreases as samples exit early; see Algorithm 1.

Algorithm 1 Batched Inference with *ThinkingViT*

$\mathcal{B} \leftarrow$ input batch
for $r = 1$ to R **do** ▷ Progressive "thinking" rounds
  $\hat{y} \leftarrow \text{Forward}(\mathcal{B}, r)$
  $\mathcal{C} \leftarrow \{i \in \mathcal{B} \mid \text{Entropy}(\hat{y}_i) \leq \tau\}$ ▷ Confident samples
  $\mathcal{U} \leftarrow \mathcal{B} \setminus \mathcal{C}$ ▷ Uncertain samples
  output predictions for samples in $\mathcal{C}$
  if $\mathcal{U}$ is empty **then break**
  end if
  $\mathcal{B} \leftarrow \text{Rebatch}(\mathcal{U})$ ▷ Shrink batch
end for

L. Effect of attention-head expansion and loss weighting on multi-round thinking

Table 6 analyzes the effect of progressively increasing the attention-head capacity (e.g., 3H→6H, 3H→9H, 3H→12H) and varying loss-weighting schemes between the first and second rounds of thinking. Starting with 3 heads and expanding to 6 (3H → 6H) yields the best first-round accuracy, while 3H → 9H achieves the highest second-round accuracy on ImageNet-1K. Notably, by using loss weighting we can tune the model to put more focus on the first round or second round based on the desired trade-off between computation and accuracy. We also explore 3-stage variants in Table 7, demonstrating that *ThinkingViT* naturally scales to deeper thinking hierarchies and yields higher final accuracy. See Figure 17 for a high-resolution plot of *ThinkingViT* variants.

Table 6. Impact of attention expansion and loss weighting on *ThinkingViT* with two rounds of progressive thinking on ImageNet-1K.

Model Variant	Loss Weight	Acc. [%] 1st Round	Acc. [%] 2nd Round
ThinkingViT 2H → 3H	[0.5, 0.5]	65.35	74.13
ThinkingViT 3H → 6H	[0.5, 0.5]	73.58	81.44
ThinkingViT 3H → 6H	[0.4, 0.6]	73.22	81.43
ThinkingViT 3H → 6H	[0.6, 0.4]	73.93	81.28
ThinkingViT 3H → 9H	[0.5, 0.5]	72.51	82.02
ThinkingViT 3H → 9H	[0.4, 0.6]	71.71	82.15
ThinkingViT 3H → 9H	[0.6, 0.4]	73.02	81.92
ThinkingViT 3H → 12H	[0.5, 0.5]	72.03	81.70
ThinkingViT 3H → 12H	[0.4, 0.6]	70.78	81.80
ThinkingViT 3H → 12H	[0.6, 0.4]	72.23	81.51

Table 7. *ThinkingViT* performance with three rounds of thinking on ImageNet-1K, demonstrating that *ThinkingViT* naturally scales to deeper thinking hierarchies and yields higher final accuracy.

Model	Acc. [%] 1st Round	Acc. [%] 2nd Round	Acc. [%] 3rd Round
ThinkingViT 2H (33%) → 3H (50%) → 6H (100%)	64.62	73.56	81.43
ThinkingViT 3H (25%) → 6H (50%) → 12H (100%)	70.77	80.00	82.35

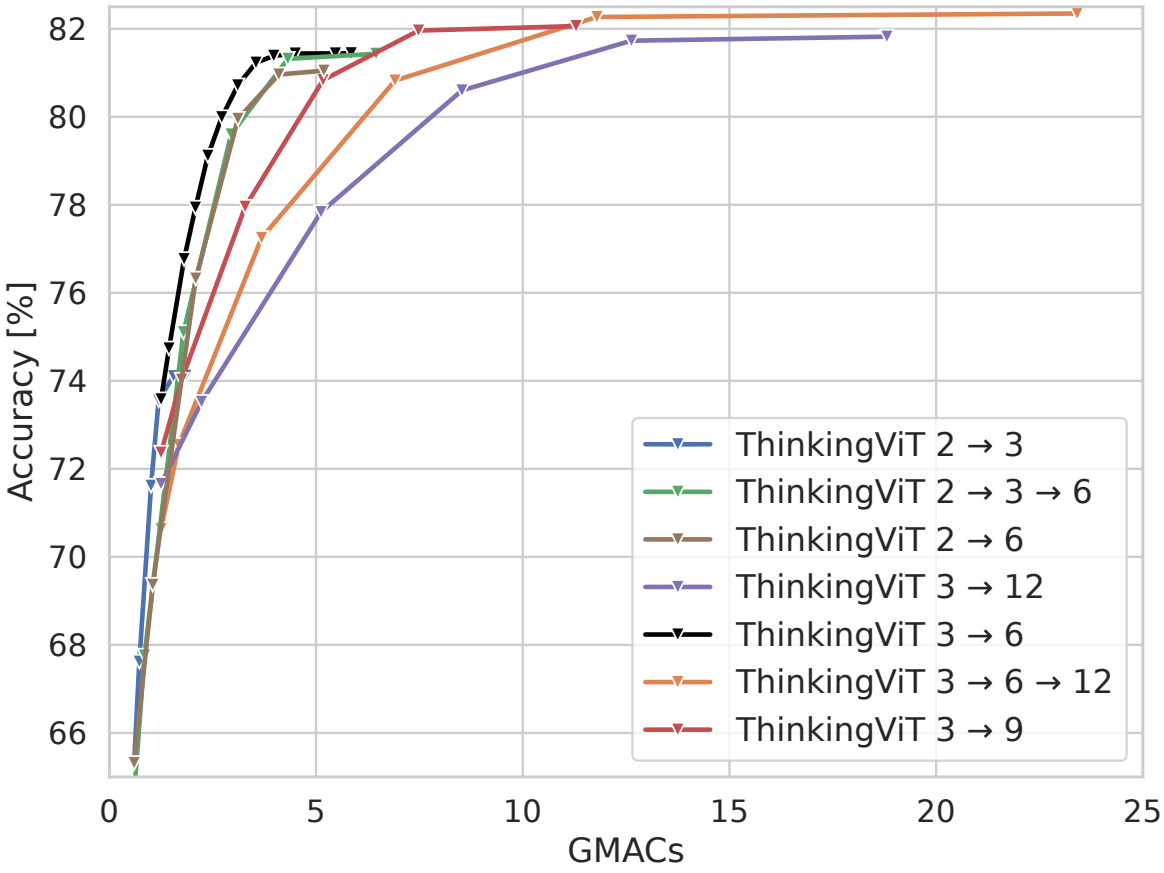

Figure 17. A higher-resolution version of Figure 3.