COPY THE SAME, DISTILL THE DIFFERENCE: INITIALIZING LINEAR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Linear Vision Transformers (ViTs) are designed to replace the attention in Softmax ViTs with the linear-complexity attention operator for more efficient token routing, but they require from-scratch pre-training and typically underperform the original Softmax version. How to initialize linear ViTs both efficiently and effectively still remains unclear. In this work, we explicitly ask: given that most foundation ViTs are built on the mainstream Softmax attention, can linear ViTs benefit from their pre-trained weights? Recent works on Attention Transfer show that attention is the effective transferable component between Softmax ViTs, suggesting attention alone suffices for such reuse. However, we find the opposite for Softmax-to-linear transfer. The attention weights are operator-specific: copying them barely helps, and is sometimes even worse than random initialization. Instead, the attention's token routing behavior can be recovered through distillation with a proper loss design, letting linear ViTs reduce the gap and even match Softmax ones. In contrast, the MLP weights, which carry the learned representation, are operator-agnostic: they can be transferred by simple direct copying, which already carries most of the benefit of the pre-trained weights. Thus, copying MLPs can serve as an effective foundation for Softmax-to-linear transfer: paired with the distilled attention, linear ViTs eventually close the remaining gap and even surpass Softmax ones. These findings hold consistently across various linear ViT variants, different model sizes, and diverse datasets. We hope this study deepens the understanding of reusing pre-trained weights across attention operators: copy what stays the same and distill what differs, to recover the benefit across the Softmax-to-linear boundary.

1 INTRODUCTION

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) have become the dominant backbone for visual representation learning in the modern era, delivering strong performance across a wide range of vision tasks. Their Softmax attention mechanism, however, incurs a quadratic $\mathcal{O}(N^2)$ cost in the number of tokens, making token routing a computational bottleneck with high-resolution or long-context tasks. To reduce this cost, linear ViTs are designed to replace the attention in Softmax ViTs with the linear-complexity attention operator for more efficient token routing, but they require from-scratch pre-training and typically underperform the original Softmax version. How to initialize linear ViTs efficiently and effectively is therefore a question of practical significance, yet still remains unclear.

Therefore, in this work, we explicitly ask: given that most foundation ViTs are built on the mainstream Softmax attention, can linear ViTs benefit from their pre-trained weights for initialization? Analytical initializations (Trockman & Kolter, 2023; Zheng et al., 2025) have shown the potential to improve from-scratch pre-training, but they still remain well below vanilla Softmax ViTs. More directly, recent works on Attention Transfer (Li et al., 2024; Qin et al., 2026b) have demonstrated that, *between Softmax ViTs*, transferring only the attention, by copying or distilling its attention maps, can already recover the original model's performance, suggesting Softmax attention alone suffices for such reuse.

However, we find the ***opposite*** for Softmax-to-linear transfer. We test the most direct reuse strategy for initialization: copying the pre-trained Softmax attention projection weights into a linear ViT, while leaving all other components randomly initialized. As shown in Fig. 2, such simple copy-only transfer from Softmax to linear ViTs barely helps, and is sometimes even worse than pure random initialization. Such failure is consistent across the linear ViT variants we study, revealing that the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

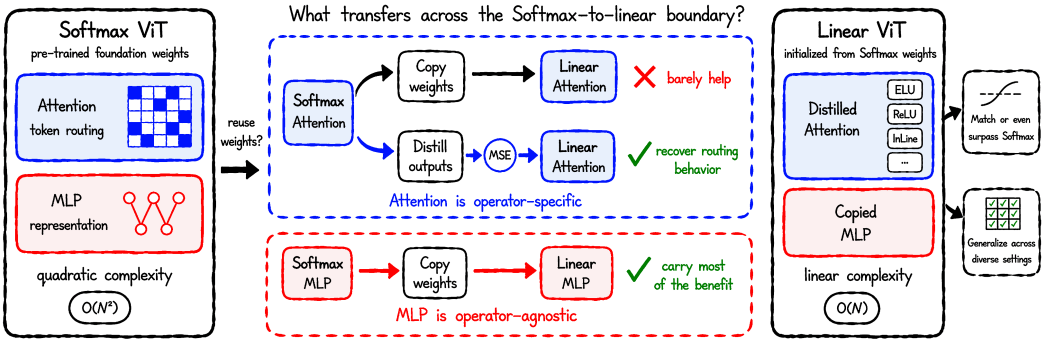

Figure 1: **What survives the Softmax-to-linear boundary?** We ask whether linear ViTs can benefit from pre-trained Softmax weights for initialization. We find that the attention weights are *operator-specific* that copying them barely helps. Their token routing behavior can instead be recovered by distilling the attention outputs, while the *operator-agnostic* MLP weights, which carry the learned representation, can be simply transferred by direct copying. Paired together, linear ViTs can match and even surpass the Softmax ones, consistently across diverse settings.

pre-trained attention weights are *operator-specific*: copied on their own, their benefit does not survive the change of the attention operator.

Does this failure mean the Softmax attention is entirely useless for linear ViTs? Crucially, no. While its weights do not transfer by copying, we identify that the token routing behavior of Softmax attention can be recovered through distillation with a proper loss design: unlike the map matching in Li et al. (2024), the loss should match the attention outputs rather than the operator-specific attention maps. This output-matching distillation lets linear ViTs largely reduce the gap to Softmax ones and even match them. The practical message is therefore not to copy the attention, but to distill it instead.

Furthermore, if the attention should be distilled rather than copied, what remains worth copying? To answer this, we decompose the transfer at the component level, selectively initializing the attention and the MLP blocks of linear ViTs from the pre-trained Softmax weights. We uncover that the MLP blocks are *operator-agnostic*: simply copying them already carries most of the benefit of the pre-trained weights. What transfers effectively across the Softmax-to-linear boundary by copying is the representation in the MLP blocks, consistent with the findings in Geva et al. (2021). Moreover, we find that copying MLPs can serve as an effective foundation for Softmax-to-linear transfer: when paired with distilled attention, linear ViTs eventually close the remaining gap and can even surpass Softmax ones; whereas copied or randomly initialized attention still leaves them below Softmax level.

We systematically validate these findings across various linear ViT variants, different model sizes, and diverse datasets. Across these settings, a simple principle emerges (see Fig. 1): copy what stays the same and distill what differs, to recover the benefit across the Softmax-to-linear boundary. We hope this principle can deepen the understanding of reusing pre-trained weights across attention operators.

In summary, our primary contributions are as follows:

1. **Systematic Study of Softmax-to-Linear Transfer.** We present the first systematic component-level study of initializing linear ViTs from pre-trained Softmax weights across linear attention operators, and reveal the attention weights are *operator-specific*: copying them barely helps and can fall below random initialization.
2. **Distillation as the Fix.** We show that a properly designed distillation loss, matching attention outputs rather than attention maps, recovers the token routing behavior and lets linear ViTs close most of the gap or even match the corresponding Softmax ones.
3. **A Simple, General Recipe.** We find that copying the *operator-agnostic* MLP weights carries most of the transfer benefit, and combining them with distilled attention can let linear ViTs surpass Softmax, yielding the principle: copy what stays the same and distill what differs.

2 Related Work

Efficient Linear ViTs. Linear ViTs replace the attention in Softmax ViTs with the linear-complexity attention operator for more efficient token routing, stemming from the kernelized linear attention

introduced in Katharopoulos et al. (2020). Recent vision designs further diversify this family: In-Line (Han et al., 2024) restores injectivity through subtraction normalization, MHLA (Zhang et al., 2026) restores expressivity through token-level multi-head computation over spatial blocks, and TTT (Han et al., 2026b) replaces explicit attention with learned test-time states. This direction also extends broadly across modalities: focused linear attention and circulant attention for vision (Han et al., 2023; 2026a), gated linear attention and delta-rule parallelization for language modeling (Yang et al., 2024a;b), and state-space models (SSMs) as an alternative linear-complexity route (Gu & Dao, 2024; Liu et al., 2024). Recent analysis further shows that a broad class of TTT architectures can be reduced to an implicit linear-attention operator (Liu et al., 2026), supporting a unified view of these models. However, these designs typically rely on dedicated pre-training from scratch, and whether they can instead be initialized from pre-trained Softmax ViTs has not been systematically studied.

Initialization for ViTs. Beyond default random initialization, analytical methods initialize weights from structural priors: Mimetic (Trockman & Kolter, 2023) shapes the query-key product toward identity-like attention, with extensions to MLPs (Trockman & Kolter, 2026) and SSMs (Trockman et al., 2024), while structured initialization (Zheng et al., 2025) imposes convolution-like attention patterns. Another line of work aims to initialize smaller models directly from the weights of larger pre-trained ones (Xu et al., 2024). These methods, however, either construct weights without any pre-trained source or reuse weights within the same attention operator; we instead study reusing pre-trained weights across the attention-operator boundary.

Attention Transfer for ViTs. Within the broader framework of knowledge distillation (Hinton et al., 2015), the idea of Attention Transfer has been extended to ViTs (Wang et al., 2022). Recently, Li et al. (2024) showed that attention patterns alone can suffice to recover the benefit of pre-trained weights between Softmax ViTs, and Qin et al. (2026b) further identified the validity boundary of such transfer under teacher-student architectural mismatch, still within the Softmax family. However, whether such attention-only transfer remains effective across the attention-operator boundary has not been studied. Our study answers this question by splitting such transfer at the component level: attention weights are operator-specific, whereas MLP weights are operator-agnostic.

Linearizing Transformers. A parallel line of work studies how to convert pre-trained Transformers into linearized ones. In language modeling, Hedgehog (Zhang et al., 2024) learns kernels to mimic the original attention maps, whereas LoLCATs (Zhang et al., 2025) and RADLADS (Goldstein et al., 2025) instead match the attention outputs when scaling such conversion to large pre-trained models. Related efforts have also appeared in vision: ViT-AdaLA (Li et al., 2026a) adapts pre-trained Softmax weights to linear ViTs, Wei & Chellappa (2025); Wang et al. (2025b) distill pre-trained ViTs into SSMs, T^5 (Li et al., 2026b) converts a pre-trained ViT into the TTT architecture, DiD (Qin et al., 2026a) extends label-free conversion to object detection by aligning detector-facing interfaces, and similar conversions have also been applied to linear diffusion transformers for more efficient image generation (Liu et al., 2025; Wang et al., 2025a). These methods show that Softmax-to-linear transfer is feasible, but each relies on dedicated adaptation before downstream use. Our study instead offers a simple, operator-general recipe: copy the operator-agnostic MLPs and distill the attention outputs.

3 Preliminaries

3.1 From Softmax to Linear Attention

Softmax Attention. Given the queries Q, keys K, and values V at block ℓ, a Softmax ViT (Dosovitskiy et al., 2021) routes tokens through the attention function[1]:

$$f^{(\ell)}_{\text{softmax}} = \text{softmax}\left(Q^{(\ell)}K^{(\ell)\top}\right)V^{(\ell)}, \tag{1}$$

which computes an *attention map* at a quadratic $\mathcal{O}(N^2)$ cost in the number of tokens N.

Linear Attention. Linear ViTs replace this Softmax operator with a linear-complexity alternative, in its standard form via a kernel function $\phi(\cdot)$ that reorders the computation as:

$$f^{(\ell)}_{\text{linear}} = \frac{\phi\big(Q^{(\ell)}\big)\left(\phi\big(K^{(\ell)}\big)^\top V^{(\ell)}\right)}{\phi\big(Q^{(\ell)}\big)\left(\phi\big(K^{(\ell)}\big)^\top \mathbf{1}_N\right)}. \tag{2}$$

[1]We omit the scaling factor $1/\sqrt{d_k}$ for simplicity.

The computational cost is therefore reduced to $\mathcal{O}(N)$ by the multiplication reordering.

3.2 Experimental Setup

Model Zoo. We study five representative linear ViT variants: ELU and ReLU (Katharopoulos et al., 2020), which instantiate ϕ as $\mathrm{ELU}(\cdot)+1$ and $\mathrm{ReLU}(\cdot)$, respectively; InLine (Han et al., 2024), which uses an identity kernel with subtraction-based normalization; MHLA (Zhang et al., 2026), which restores the expressivity of linear attention through token-level multi-head computation with auxiliary convolutions; and TTT (Han et al., 2026b), which replaces the attention computation entirely with a compact linear-complexity inner model constructed from KV pairs. The first three variants are drop-in linear operators whose weight shapes exactly match the Softmax attention, while MHLA and TTT introduce additional components. Full descriptions of different linear ViTs and the corresponding attention operators are given in Sec. B.1.

Transfer Protocol. To conduct a systematic analysis of the Softmax-to-linear attention transfer, we instantiate the five linear ViT variants across Tiny, Small, and Base sizes, all with the DeiT (Touvron et al., 2021) pre-trained weights of the corresponding model size as the transfer source. Unless specifically mentioned, all linear ViTs are with their default implementations and hyper-parameters. For linear ViTs with additional components beyond the Softmax ones, the unmatched parameters are initialized following their default settings. More details are in Sec. B.2.

Evaluation Protocol. We evaluate Tiny and Small linear ViTs on six transfer datasets: CIFAR-10, CIFAR-100 (Krizhevsky, 2009), STL-10 (Coates et al., 2011), Food (Bossard et al., 2014), Flowers (Nilsback & Zisserman, 2008), and Pets (Parkhi et al., 2012). We additionally evaluate all three sizes on ImageNet-1K (Deng et al., 2009) for large-scale validation. We conduct all the experiments with the default settings following the recipe of Xu et al. (2024), training for 300 epochs or 600 epochs depending on the dataset scale. All reported results are averaged over 3 random seeds to ensure statistical significance. More implementation details are provided in Sec. B.3.

Reference Baselines. For reference purposes, we set two baselines across all three model sizes for comparison. *Random* baseline defines the from-scratch reference of the Softmax-to-linear transfer: one transfer is considered effective only if it improves over training from scratch. *Softmax* baseline defines the recovery target: the performance of the pre-trained Softmax ViT of the same model size on the same task, which the transfer aims to recover.

4 Softmax Attention Weights Are Operator-Specific

We begin by asking whether the token routing behavior encoded in pre-trained Softmax attention weights can be simply transferred as weights by copying across the Softmax-to-linear boundary.

4.1 Attention-Only Transfer

We first evaluate the strategy motivated by Attention Transfer between Softmax ViTs (Li et al., 2024; Qin et al., 2026b): transferring only the attention. Beyond the reference baselines defined in Sec. 3.2, we further include additional weight-initialization settings as follows. *Copy* baseline loads the pre-trained Softmax attention projection weights, including queries Q, keys K, values V, into every block of the linear ViT, while leaving all other components randomly initialized. *Mimetic* (Trockman & Kolter, 2023) and *Impulse* (Zheng et al., 2025) are two analytical initializations that construct attention weights from structural priors without using any pre-trained source, serving as stronger baselines than *Random*.[2] We compare all these baselines under the same training recipe. The assumption is that if the pre-trained Softmax attention alone suffices to preserve model performance across the Softmax-to-linear boundary, *Copy* should provide a clear advantage over all three alternatives.

4.2 Main Results

Fig. 2 reports the comparison across all five linear ViTs, seven datasets, and different model sizes. The results reveal a clear failure of the expected advantage of *Copy*. Across all the experiments, simply copying the Softmax weights into the linear operator, which has been shown effective in Li et al. (2024); Qin et al. (2026b) between Softmax ViTs, *never* stands out as the strongest initialization.

[2]Mimetic and Impulse assume the standard $QK^\top$ attention parameterization and are therefore incompatible with TTT (Han et al., 2026b).

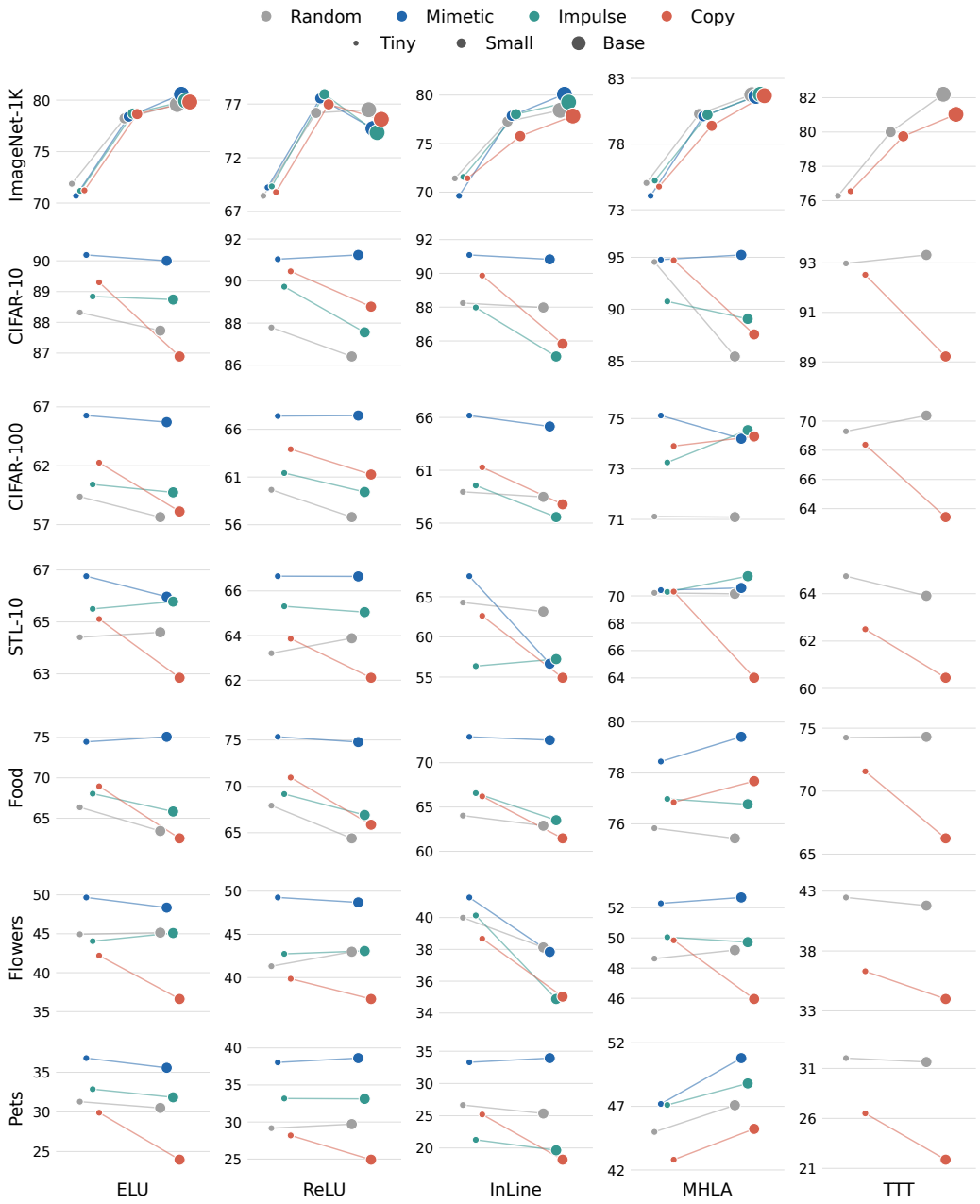

Figure 2: **Attention-only Softmax-to-linear transfer barely helps.** We begin by evaluating the attention-only initializations across five linear ViTs (columns) and seven datasets (rows). We report the Top-1 Accuracy (%); marker size denotes the model size (Tiny/Small/Base), and lines denote the same initialization across sizes. Across all operators, datasets, and sizes, *Copy* never stands out as the strongest attention-only initialization and remains at or near the *Random*-initialization tier.

This observed failure appears mainly in the following three ways: First, adopting Softmax weights loses to pre-training-free initialization: across datasets, *Mimetic*, built without ever using any pre-trained source, systematically outperforms *Copy*, which loads the pre-trained Softmax attention weights. Second, copying barely helps: even against *Random*, *Copy* yields no consistent gain, in sharp contrast to the recovery attention-only transfer achieves between Softmax ViTs. Third, copying can sometimes hurt: as shown, *Copy* degrades performance by large margins compared to *Random*; this drop is consistent for TTT and also appears for other linear ViTs on datasets such as Pets and Flowers. Taken together, these results show that direct attention-only copying provides no reliable initialization advantage across the Softmax-to-linear boundary, the opposite of the transfer behavior observed between Softmax ViTs. Further interpretations of this failure are given in Sec. D.

Table 1: **Pre-trained weights robustness.** We repeat the attention-only transfer on ViT-Base with different pre-trained Softmax weights, under the same ImageNet-1K evaluation. Gray row shows the shared *Random* baseline, and colored deltas indicate gains/drops relative to it. The failure remains consistent across all pre-trained sources: none provides a reliable advantage over random initialization.

	ELU	ReLU	MHLA	TTT
Random	79.6	76.5	81.8	82.2
DeiT	79.8 +0.2	75.6 -0.9	81.7 -0.1	81.0 -1.2
DINO	80.2 +0.6	75.2 -1.3	81.6 -0.2	80.6 -1.6
MoCov3	79.9 +0.3	75.7 -0.8	81.7 -0.1	81.1 -1.1
iBOT	80.3 +0.7	75.3 -1.2	81.8 +0.0	80.7 -1.5
MAE	80.4 +0.8	76.0 -0.5	82.0 +0.2	81.4 -0.8

Table 2: **Out-of-distribution robustness.** We evaluate the transferred ViT-Base models on four distribution-shift benchmarks. Colored deltas indicate gains/drops relative to the *Random* baseline, and *Softmax* denotes the Softmax target reference. Copying the Softmax attention gains no additional robustness and stays equally far from the Softmax reference, consistent with the in-distribution evaluation.

	ELU		TTT		Softmax
	Random	Copy	Random	Copy	
IN-A	18.1	19.1 +1.0	24.1	23.7 -0.4	29.2
IN-R	38.9	39.6 +0.7	42.3	42.2 -0.1	45.8
IN-S	27.2	27.8 +0.6	30.0	29.9 -0.1	33.4
IN-V2	66.8	67.2 +0.4	69.3	69.1 -0.2	72.4

4.3 Robustness of the Failure

One natural counter-hypothesis is that this observed failure might be an artifact of the evaluated setting: with larger model capacity, with Softmax weights from a different pre-training algorithm, or beyond the in-distribution evaluation setting, copying might still be able to deliver its expected advantage. To rule this out, we rigorously test the robustness of this failure along the three axes below.

Model Size Robustness. We first compare the attention-only initializations across the evaluated sizes on all transfer datasets. As shown in Fig. 2, the qualitative failure pattern holds consistently at all the tested sizes: within each plot, the attention-only *Copy* remains at or near the *Random*-initialization tier. In contrast, the analytical initializations hold their advantage as the model size grows, further confirming that the failure is robust to model size and is not an artifact of limited model capacity.

Pre-trained Weights Robustness. We then test whether the failure is merely a deficiency of the supervised DeiT weights. Table 1 reports the attention-only transfer with the Softmax weights from other well-known pre-training algorithms, including DINO (Caron et al., 2021), MoCov3 (Chen et al., 2021), iBOT (Zhou et al., 2022), and MAE (He et al., 2022), under the same ImageNet-1K evaluation. The same trend appears for all weights: copying can shift the performance only marginally, but can never turn into a reliable advantage over *Random*. This confirms that the failure is robust to the choice of pre-trained weights, while changing this Softmax source cannot reverse it.

Out-of-Distribution Robustness. The observed failure also persists under the out-of-distribution evaluation. Table 2 further tests the transferred models on four ImageNet-1K distribution-shift benchmarks (Hendrycks et al., 2021b;a; Wang et al., 2019; Recht et al., 2019), where *Copy* merely preserves its marginal in-distribution difference from *Random* and gains no additional robustness, while both initializations remain equally far from the Softmax reference. This further confirms that the failure is robust to substantial distribution shift.

Overall, the comprehensive evaluations above show that attention-only Softmax-to-linear copy consistently stays at or near the random-initialization tier. Thus, the pre-trained Softmax attention weights are *operator-specific*: copied on their own, their benefit does not survive the change of the attention operator. This raises the questions: does this failure mean that the Softmax attention is entirely useless for linear ViTs, and if its weights should not be copied, what remains worth copying?

5 Distill the Routing, Copy the Representation

Having established the failure of attention-only copying, we now ask what exactly the pre-trained Softmax weights can offer to linear ViTs. We answer the two questions raised in Sec. 4 in turn: the attention's token routing behavior can be effectively transferred through distillation rather than copying, while the representation carried in the MLP blocks can be transferred through direct copying; together, the two findings form a simple recipe for Softmax-to-linear transfer. Unless specifically mentioned, all results below are reported with ViT-Small on the transfer datasets.

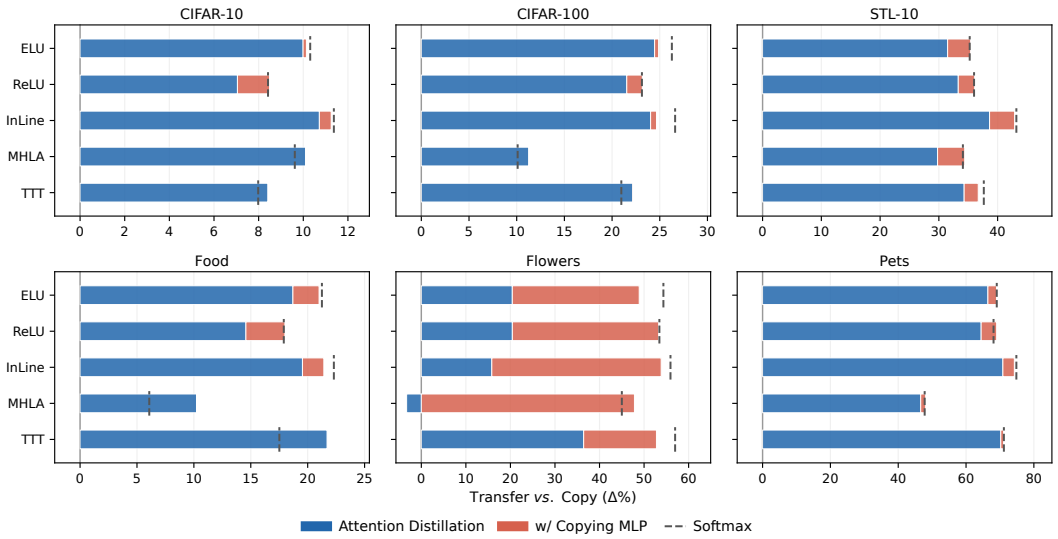

Figure 3: **Distillation rescues attention-only copying.** For each linear operator with ViT-Small, we report the accuracy gain over *Copy*: blue bars show the gain from distilling the attention on top of *Copy*, and red bars are the additional gain from further copying the MLP weights (*FullCopy*), leading the full setting to correspond to *FullCopy w/ Distill*. The dashed line marks the corresponding Softmax recovery target. Distillation effectively recovers most of the gap across linear operators on most datasets; further combining the copied MLPs can even reach the Softmax target.

5.1 Token Routing Can Be Recovered through Distillation

Given the failure of direct copying, we hypothesize that the token routing behavior encoded in the Softmax attention can be transferred as *behavior* instead of as weights. To test this, we distill from a frozen pre-trained Softmax teacher during downstream training: an additional distillation loss aligns the attention behavior between the student and the teacher at every block, with λ as the weight of this loss in the overall training objective. For transfer between Softmax ViTs, this distillation loss is computed between the attention maps as in Li et al. (2024); Qin et al. (2026b). However, for Softmax-to-linear transfer, given the differences among linear operators and the resulting operator-specific attention maps, with some operators never materializing a map at all, we deliberately place the loss between the attention *outputs*. This output-matching choice parallels the practice in language-model linearization (Zhang et al., 2025), in contrast to map-based mimicry (Zhang et al., 2024).

Fig. 3 shows that this distillation effectively rescues the attention-only copying across linear operators on various datasets.[3] Compared to *Copy*, attention distillation closes most of the gap to the Softmax recovery target for every linear operator, and can even reach this target for MHLA and TTT on some datasets. One exception is the fine-grained Flowers, where attention distillation alone leaves a larger gap, and for MHLA it even falls below *Copy*: with the small number of training samples, the task signal appears too weak to relearn the representation that attention-only supervision cannot supply.

Together with the failure of copying established in Sec. 4, these results confirm that the token routing does transfer across the Softmax-to-linear boundary: as behavior through distillation, not as weights through copying. The practical message for the attention is thus not to copy it, but to distill it instead. This directly motivates the second question: if the attention should be distilled rather than copied, what of the pre-trained weights remains worth copying?

5.2 MLP Weights Are Operator-Agnostic

To answer this, we decompose the transfer at the component level by selectively initializing the attention and the MLP blocks of linear ViTs. We additionally define two more weight-initialization settings: *MLPCopy*, which directly copies the MLP blocks from the pre-trained Softmax weights, while leaving all attention-related components randomly initialized; and *FullCopy*, which copies

[3] All reported distillation results in Fig. 3 use the output-matching loss with a fixed calibrated $\lambda = 50$, shared across all linear operators and datasets. A detailed analysis of the loss design is given in Sec. 5.4.

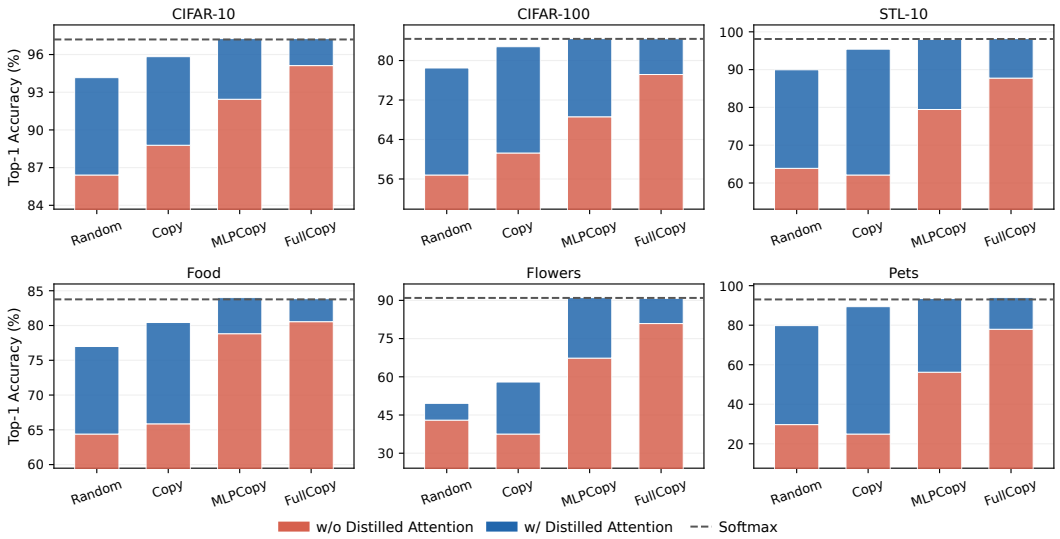

Figure 4: **MLP weights survive Softmax-to-linear transfer.** We selectively initialize the ReLU-based linear ViT-Small by copying from the pre-trained Softmax weights at the component level, yielding four weight-initialization settings: *Random*, *Copy*, *MLPCopy*, and *FullCopy*. For each, we report its plain accuracy and the accuracy with the distilled attention (*w/ Distill*). Red bars denote the accuracy of each initialization alone, and blue bars show the additional gain from the distilled attention. The dashed line marks the corresponding Softmax recovery target. Copying MLPs can already effectively carry most of the benefit of the pre-trained weights, while distilling the attention can lift every one of them and bring *MLPCopy* and *FullCopy* to match the Softmax level.

all pre-trained Softmax weights. Fig. 4 shows a direct comparison of the four weight-initialization settings with the ReLU-based linear ViTs. As shown, *Random* and *Copy* achieve comparable accuracy across datasets. Compared to both, *MLPCopy* rises far above on all datasets; while *FullCopy* yields further consistent gains over *MLPCopy* by additionally copying the attention.[4]

Therefore, the MLP blocks are *operator-agnostic*: simply copying them already carries most of the benefit of the pre-trained weights, consistent with the view of MLPs as key-value memories that store learned knowledge (Geva et al., 2021). Compared with the attention findings above for Softmax-to-linear transfer, this indicates that the learned representation carried in MLP weights can be effectively transferred generally across various transfer settings, whereas the copied attention weights bring no reliable benefit on their own once the operator changes, similar to the architectural-mismatch findings in Qin et al. (2026b). We conclude that what transfers effectively across the Softmax-to-linear boundary, by direct copying, is primarily the representation in the MLP blocks.

5.3 The Recipe

Having identified what is worth copying and what is worth distilling, we next ask whether the two transfer methods can be paired to reach the Softmax recovery target more effectively. We answer by revisiting the distillation results in Fig. 3 and the copying results in Fig. 4, in both cases replacing the randomly initialized components with their effectively transferred counterparts.

Specifically, although distilled attention brings significant gains in Fig. 3, recovery to the Softmax target is still not consistent across datasets and linear operators. Combining the copied MLPs can provide consistent further gains wherever the target is not yet reached, largely closing the remaining gap across operators and datasets. In addition, although all four weight-initialization settings shown in Fig. 4 yield accuracy below the Softmax target, distilling the attention lifts every one of them, bringing *MLPCopy* and *FullCopy* to match the Softmax level. Interestingly, *MLPCopy w/ Distill* lands close to *FullCopy w/ Distill*, showing that once the token routing behavior is distilled, the copied attention weights become nearly redundant, with no consistent advantage compared to copying-only settings. Moreover, distillation also reverses the standing of the copied attention itself: while *Copy*

[4]We focus only on the effect of loading the attention and MLP weights, as the ablation in Sec. C.4 shows that copying the remaining components yields no significant gain.

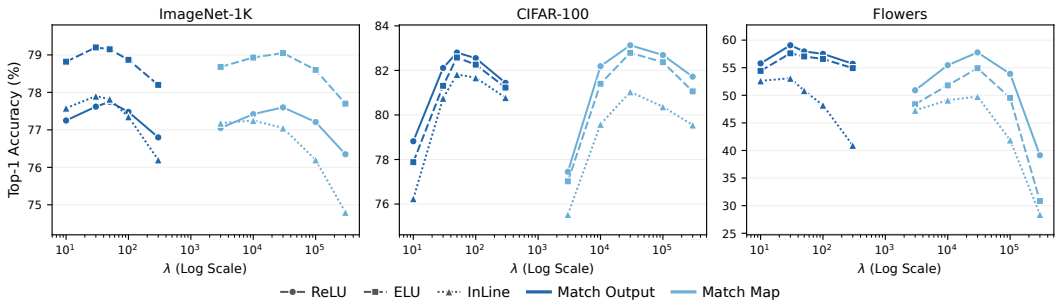

Figure 5: **Loss ablation on matching attention outputs *vs*. attention maps.** For *Copy*-initialized ViT-Small students, we compare MSE matching on attention *outputs* and attention *maps* with varying λ separately across datasets of different scales. Both matching targets show consistent accuracy trends under the λ-sweep. For ReLU and ELU, whose maps are proper distributions, the calibrated peaks are comparable. In contrast, for InLine, whose map is signed, map matching falls behind, suggesting that attention output matching is a more general and proper design.

alone stays near the *Random*-initialization tier, *Copy w/ Distill* consistently exceeds *Random w/ Distill*; we attribute this to the copied attention weights offering a useful warm start for recovering the routing behavior, although this advantage is subsumed once the MLP weights are copied.

Together, copying the operator-agnostic MLPs with the distilled attention can close the remaining gap and eventually match the Softmax recovery target, in some cases even surpassing it. Therefore, we suggest one simple principle for reusing the pre-trained weights across the Softmax-to-linear boundary: copy what stays the same and distill what differs. We provide further discussions on why copying the MLPs and distilling the attention outputs are complementary in Sec. D.

5.4 Discussion and Analysis

Ablation on Distillation Loss Design. We further examine the loss-placement choice in Sec. 5.1 by sweeping the distillation loss weight λ separately for output matching and map matching, each around the λ with the peak accuracy, across datasets of different scales. As shown in Fig. 5, for ReLU and ELU, whose attention maps are proper distributions, map matching and output matching yield comparable peak recovery, confirming that the recovery is robust to the specific loss form. However, for InLine, whose map is a signed quasi-distribution due to its subtraction normalization, map matching falls consistently behind the output matching. Therefore, a more general design for Softmax-to-linear attention distillation should match the routing behavior at the operator-agnostic attention output rather than the operator-specific attention maps.

Ablation on Distillation Loss Weight λ. Beyond the loss choice, the sweep in Fig. 5 also shows that the output-matching peak is stable in λ: accuracy varies within a broad plateau around the peak, which roughly aligns across operators and datasets; we thus adopt $\lambda = 50$ as the default distillation loss weight in our experiments.

Ablation on More Linear ViTs, Sizes, and Tasks. While all results in Sec. 5 are with ReLU-based ViT-Small on classification, we verify the generalizability of these findings with comprehensive evaluations on various linear ViTs, more model sizes, and dense prediction tasks under the same experimental protocols. As shown in Secs. C.1 to C.3, patterns consistent with those in Fig. 4 can also be observed, confirming the robustness of the transfer recipe across model types, sizes, and tasks.

6 Conclusion

In this work, we study whether linear ViTs can be initialized by reusing the pre-trained Softmax weights, and find that the most direct strategy fails: the attention weights are *operator-specific*, and copying them can even fall below random initialization. Instead, the attention's token routing behavior can be recovered through distillation with a proper loss design that matches the attention outputs. The MLP weights, in contrast, are *operator-agnostic*: simply copying them already carries most of the benefit of the pre-trained weights. Paired together, the two close the remaining gap to the Softmax target and can even surpass it. This yields a simple but general recipe: copy what stays the same and distill what differs. We hope these findings sharpen the understanding of reusing pre-trained weights across attention operators and encourage further study of efficient transfer beyond the Softmax setting.

AI USE STATEMENT

We certify the use of AI to aid in the writing, polishing, and figure plotting of this manuscript, and the use of AI to aid code implementations of involved experiments in this work. All AI-assisted content shown in this manuscript has been reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of AI.

ETHICS STATEMENT

This work is primarily a methodological study of whether linear ViTs can benefit from pre-trained Softmax weights for initialization. Its direct positive impact is to encourage more efficient and effective linear ViT initialization, and to help the community sharpen the understanding of reusing pre-trained weights across attention operators. A potential negative impact is misinterpretation: our findings should not be read as "Attention Transfer does not work across attention operators", but rather as a recipe for how it should be conducted: copy what stays the same and distill what differs. Finally, this paper relies on existing public pre-trained Softmax weights, some of which are trained on large-scale curated or web-scale datasets and may inherit dataset biases. The proposed recipe for Softmax-to-linear transfer can make this transfer more effective, but it does not remove biases or safety concerns inherited from the original teacher weights.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, we commit to making our source code publicly available upon publication. The code will include our implementations and scripts to replicate the empirical results presented in this manuscript. Comprehensive experimental setups of our analysis, including model zoo, transfer protocol, evaluation protocol, and selection of reference baselines, are provided in Sec. 3.2 and further detailed in Sec. B. Additionally, all experiments are repeated over 3 random seeds to ensure statistical significance. Table F provides the statistical significance analysis of our obtained results.

REFERENCES

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pp. 446–461, 2014. 4, 17

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pp. 9650–9660, 2021. 6

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pp. 9640–9649, 2021. 6

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *ICLR*, 2023. 20, 21

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pp. 215–223, 2011. 4, 17

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *CVPR*, pp. 248–255, 2009. 4, 17

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 1, 3

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *EMNLP*, pp. 5484–5495, 2021. 2, 8

Daniel Goldstein, Eric Alcaide, Janna Lu, and Eugene Cheah. RADLADS: Rapid attention distillation to linear attention decoders at scale. In *COLM*, 2025. URL https://openreview.net/forum?id=38GehGepDd. 3

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *COLM*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2. 3

Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *ICCV*, pp. 5938–5948, 2023. 3

Dongchen Han, Yifan Pu, Zhuofan Xia, Yizeng Han, Xuran Pan, Xiu Li, Jiwen Lu, Shiji Song, and Gao Huang. Bridging the divide: Reconsidering softmax and linear attention. In *NeurIPS*, pp. 79221–79245, 2024. 3, 4, 16, 22

Dongchen Han, Tianyu Li, Ziyi Wang, and Gao Huang. Vision transformers are circulant attention learners. In *AAAI*, pp. 21549–21557, 2026a. 3

Dongchen Han, Yining Li, Tianyu Li, Zixuan Cao, Ziming Wang, Jun Song, Yu Cheng, Bo Zheng, and Gao Huang. ViT^3: Unlocking test-time training in vision. In *CVPR*, pp. 51–61, 2026b. 3, 4, 15, 16

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pp. 2961–2969, 2017. 19

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pp. 16000–16009, 2022. 6

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pp. 8340–8349, 2021a. 6

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pp. 15262–15271, 2021b. 6

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, pp. 5156–5165, 2020. 3, 4, 16, 22

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 2009. 4, 17

Alexander C Li, Yuandong Tian, Beidi Chen, Deepak Pathak, and Xinlei Chen. On the surprising effectiveness of attention transfer for vision transformers. In *NeurIPS*, pp. 113963–113990, 2024. 1, 2, 3, 4, 7, 15, 18

Yifan Li, Seunghyun Yoon, Viet Dac Lai, Franck Dernoncourt, Jason Kuen, Yu Kong, and Trung Bui. Vit-adala: Adapting vision transformers with linear attention. *arXiv preprint arXiv:2603.16063*, 2026a. 3, 15

Yining Li, Dongchen Han, Zeyu Liu, Hanyi Wang, Yulin Wang, and Gao Huang. Linearizing vision transformer with test-time training. In *ICML*, 2026b. 3, 15

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755, 2014. 19

Junchen Liu, Sven Elflein, Or Litany, Zan Gojcic, and Ruilong Li. Test-time training with KV binding is secretly linear attention. In *ICML*, 2026. 3

Songhua Liu, Zhenxiong Tan, and Xinchao Wang. Clear: Conv-like linearization revs pre-trained diffusion transformers up. In *NeurIPS*, 2025. 3

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*, pp. 103031–103063, 2024. 3

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pp. 722–729. IEEE, 2008. 4, 17

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, pp. 3498–3505, 2012. 4, 17

Huaiyuan Qin, Gabriel James Goenawan, Zihang Lin, Muli Yang, and Hongyuan Zhu. Did it in 87 minutes: A label-free softmax-to-linear adaptation of vision transformers for object detection. *arXiv preprint arXiv:2608.22368*, 2026a. 3, 15

Huaiyuan Qin, Muli Yang, Gabriel James Goenawan, Peng Hu, Chen Gong, Xi Peng, and Hongyuan Zhu. Attention transfer is not universally effective for vision transformers. *arXiv preprint arXiv:2605.07191*, 2026b. 1, 3, 4, 7, 8, 15, 18

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, pp. 5389–5400, 2019. 6

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357, 2021. 4, 16

Asher Trockman and J. Zico Kolter. Mimetic initialization of self-attention layers. In *ICML*, pp. 34456–34468, 2023. 1, 3, 4

Asher Trockman and J. Zico Kolter. Mimetic initialization of mlps. *arXiv preprint arXiv:2602.07156*, 2026. 3

Asher Trockman, Hrayr Harutyunyan, J Zico Kolter, Sanjiv Kumar, and Srinadh Bhojanapalli. Mimetic initialization helps state space models learn to recall. *arXiv preprint arXiv:2410.11135*, 2024. 3

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. 6

Jiahao Wang, Ning Kang, Lewei Yao, Mengzhao Chen, Chengyue Wu, Songyang Zhang, Shuchen Xue, Yong Liu, Taiqiang Wu, Xihui Liu, et al. Lit: Delving into a simple linear diffusion transformer for image generation. In *ICCV*, pp. 16068–16078, 2025a. 3, 15

Kai Wang, Fei Yang, and Joost van de Weijer. Attention distillation: Self-supervised vision transformer students need more guidance. In *BMVC*, 2022. 3

Penghao Wang, Yuhao Zhou, Mengxuan Wu, Panpan Zhang, Zhangyang Wang, and Kai Wang. Data efficient any transformer-to-mamba distillation via attention bridge. *arXiv preprint arXiv:2510.19266*, 2025b. 3

Guoyizhe Wei and Rama Chellappa. Vit-linearizer: Distilling quadratic knowledge into linear-time vision models. In *ICCV*, pp. 20737–20747, 2025. 3

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pp. 432–448, 2018. 20

Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. In *ICLR*, 2024. 3, 4, 17

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *ICML*, pp. 56501–56523, 2024a. 3

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *NeurIPS*, pp. 115491–115522, 2024b. 3

Kewei Zhang, Ye Huang, Yufan Deng, Jincheng Yu, Junsong Chen, Huan Ling, Enze Xie, and Daquan Zhou. MHLA: Restoring expressivity of linear attention via token-level multi-head. In *ICLR*, pp. 12225–12245, 2026. 3, 4, 16

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *ICLR*, pp. 53551–53580, 2024. 3, 7, 22

Michael Zhang, Simran Arora, Rahul Chalamala, Benjamin Frederick Spector, Alan Wu, Krithik Ramesh, Aaryan Singhal, and Christopher Ré. Lolcats: On low-rank linearizing of large language models. In *ICLR*, pp. 46200–46253, 2025. 3, 7, 15

Jianqiao Zheng, Xueqian Li, Hemanth Saratchandran, and Simon Lucey. Structured initialization for vision transformers. In *NeurIPS*, pp. 132486–132511, 2025. 1, 3, 4

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pp. 5122–5130, 2017. 20

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 6

Contents

1 Introduction **1**

2 Related Work **2**

3 Preliminaries **3**
3.1 From Softmax to Linear Attention 3
3.2 Experimental Setup 4

4 Softmax Attention Weights Are Operator-Specific **4**
4.1 Attention-Only Transfer 4
4.2 Main Results 4
4.3 Robustness of the Failure 6

5 Distill the Routing, Copy the Representation **6**
5.1 Token Routing Can Be Recovered through Distillation 7
5.2 MLP Weights Are Operator-Agnostic 7
5.3 The Recipe 8
5.4 Discussion and Analysis 9

6 Conclusion **9**

A Scope and Relation to Similar Methods **15**

B Additional Implementation Details **16**
B.1 Details of Model Zoo 16
B.2 Details of Transfer Protocol 16
B.3 Details of Evaluation Protocol 17
B.4 Details of Attention Distillation 18

C Additional Results and Analysis **18**
C.1 Additional Results on More Linear Operators 18
C.2 Additional Results on More Model Sizes 18
C.3 Additional Results on ImageNet and Dense Prediction 19
C.4 Additional Results on Copying the Remaining Components 20
C.5 Additional Results on Statistical Significance 22

D Additional Interpretation of Softmax-to-Linear Transfer **22**
D.1 Preservation under Weight Copying 22
D.2 Why Copying MLPs and Distilling Attention Are Complementary 23

E Limitations **23**

A Scope and Relation to Similar Methods

We provide further discussion on the scope of this study and clarify its relation to existing methods for Softmax-to-linear transfer below.

Relation to linearization methods. Generally, our work is a controlled empirical study of *initialization* of linear ViTs: we explore the reuse of pre-trained Softmax weights by factorizing them at the component level, test whether each component can be transferred as weights or as behavior, and validate our findings under fixed training recipes on various downstream tasks across five linear attention operators. We do not claim to propose a new linearization method, and we do not claim to improve upon the methods below on their own benchmarks:

- **LoLCATs** (Zhang et al., 2025) linearizes large language models by training linear attention to match Softmax attention outputs under an MSE loss, followed by low-rank adaptation. The attention-output-matching objective used in Sec. 5.1 follows this established practice. Our contribution is not the loss itself, but the finding that this behavioral target is well-defined for ViTs and effective across all five linear operators, including those that never materialize an attention map.
- **ViT-AdaLA** (Li et al., 2026a) adapts pre-trained Softmax ViTs to vanilla linear attention through a dedicated pipeline of block-level attention alignment, final-layer feature alignment, and supervised fine-tuning, establishing a vision-specific use of attention alignment for one operator. Our findings instead offer a simple and operator-general recipe that requires no dedicated adaptation stage.
- **LiT** (Wang et al., 2025a) converts pre-trained diffusion transformers into linear ones, where its practical guideline is to load all parameters *except* those of the linear attention and combine the selective inheritance with hybrid distillation on the predicted noise and variance. Our finding that MLP weights are operator-agnostic while attention weights are operator-specific is consistent with this guideline, and extends the observation from one diffusion setting to a component-level comparison across five linear operators, with the distillation signal placed on attention outputs rather than task predictions.
- **T^5** (Li et al., 2026b) converts pre-trained ViTs into the TTT architecture by inheriting projection, MLP, and normalization weights while randomly initializing the TTT inner operator, which corresponds to the *FullCopy* setting in our work. With detailed attention-only and MLP-only ablations, our decomposition analysis instead identifies which component carries the benefit, and shows that the attention is better transferred by distillation than by copying. Our TTT student is the standard ViT^3 block of Han et al. (2026b) without T^5's architectural modifications, which concerns the standard form of this architecture family.
- **DiD** (Qin et al., 2026a) converts the Softmax-attention backbone of a trained detector into a linear-attention one without labels, keeping the downstream detector fixed and distilling the detector-facing interface tensors so that the converted backbone reproduces the features the detector expects. It addresses a different setting from ours: a post-hoc, label-free Softmax-to-linear conversion under a frozen detector, whereas we study how to initialize a linear ViT that is subsequently trained with supervision, and our dense-prediction results in Sec. C.3 fine-tune the whole detector with a linear backbone transferred from Softmax pre-trained weights. Our findings are complementary: DiD shows that behavior-level alignment suffices to swap the attention operator with a fixed detector, while our decomposition identifies which and how pre-trained components should be copied or distilled at the initialization stage, when downstream heads are trainable.

Relation to Attention Transfer. Attention Transfer (Li et al., 2024; Qin et al., 2026b) in vision studies whether transferring only the attention from a pre-trained Softmax ViT to a Softmax student with the same architecture can recover the original model's performance. These studies show that Attention Copy, which keeps the teacher's attention fixed throughout student training by injecting the teacher's attention maps in every forward pass, is one effective way for such transfer. Our *Copy* setting is inspired by this, but provides a different operation: the copied pre-trained attention projection weights serve only as an initialization, and every parameter, including the copied ones, is trained jointly on downstream tasks. This difference is deliberate due to the different focus on research

questions: Attention Transfer asks whether the attention alone suffices to transfer a Softmax ViT, while we ask whether the Softmax attention *weights* can initialize a linear attention operator. The failure of *Copy* in Sec. 4 is therefore a finding on weight-level initialization across the operator boundary, rather than a contradiction of Attention Transfer within the Softmax family.

B Additional Implementation Details

We provide further implementation details to support reproducibility of our experimental setup below.

B.1 Details of Model Zoo

In the Softmax-to-linear transfer experiments, all Softmax models follow the default implementation of the DeiT architecture (Touvron et al., 2021) across the Tiny, Small, and Base sizes. Linear models differ from the Softmax ones only in the attention operator inside each transformer block, while all other components remain unchanged. We study five representative linear ViT variants in this work, with detailed descriptions below:

- **ELU and ReLU** (Katharopoulos et al., 2020) instantiate Equation (2) with $\phi(x) = \mathrm{ELU}(x) + 1$ and $\phi(x) = \mathrm{ReLU}(x)$, respectively. They are computed in the reordered linear-complexity form, with the projection weight shapes identical to those in Softmax.
- **InLine** (Han et al., 2024) uses the identity kernel with subtraction normalization to restore injectivity. With d the per-head dimension, $s = d^{-1/2}$, and $\bar{k}$ the mean key, the map is $A_{ij} = \frac{s}{N}\, q_i^\top k_j + \frac{1}{N}\big(1 - s\, q_i^\top \bar{k}\big)$, whose rows sum to one while individual entries may be negative. It is computed with linear complexity as $\frac{s}{N}\, q_i^\top (K^\top V) + (1 - s\, q_i^\top \bar{k})\, \bar{v}$, where $\bar{v}$ is the mean value. We use the attention-only form without the local residual branch of the original block in their official implementation, which aggregates the 3×3 neighborhood of values with input-dependent weights, thus the projection weight shapes are identical to those in Softmax attention.
- **MHLA** (Zhang et al., 2026) partitions the token sequence into non-overlapping spatial blocks, treated as token-level heads, and applies linear attention within each block. Following the official implementation, we use the ReLU kernel $\phi(x) = \mathrm{ReLU}(x)$, with four 7×7 blocks on the 14×14 token grid. The per-block key-value summaries and normalizers are then mixed across blocks by a learned 1×1 convolution mixing matrix, and, following the official implementation, a 5×5 depthwise convolution on the values provides a positional term that is added to the output before the output projection. All projection weights have the same shapes as in Softmax attention, with the mixing matrix and the depthwise convolution as additional components.
- **TTT** follows the ViT^3 block (Han et al., 2026b), which replaces the attention in ViTs with two inner models fitted on the fly for each image: a SwiGLU inner module with per-head weights $W_1, W_2 \in \mathbb{R}^{d \times d}$ and a 3×3 depthwise-convolution inner module W_3 as one additional head. Each inner model is updated by one closed-form gradient step on the key-value pairs of the current image and then applied to the queries. The two branch outputs are concatenated and mapped back to the model dimension C by a $(C + d) \to C$ output projection. The QKV projection therefore has the $3C$ rows of standard queries, keys, and values for the SwiGLU branch plus $3d$ rows for the convolutional branch; the inner weights, the extra rows, and the wider output projection are additional components.

B.2 Details of Transfer Protocol

Unless specifically mentioned, all reported results use the transfer source of the official ImageNet-1K DeiT pre-trained weights (Touvron et al., 2021) with the same size as the student. Downstream heads for different tasks are randomly initialized. We summarize the main weight-initialization settings used in our evaluations below, and also provide more details in Table A:

- **Copy** loads the pre-trained Softmax attention projection weights, including queries Q, keys K, values V, into every block of the linear ViT, while leaving all other components randomly

Table A: **Implementation details of transfer protocol.** We list the weight-initialization details of each component in the Softmax-to-linear transfer experiments. Note that we follow the default implementations in MHLA and TTT to use sinusoidal positional embeddings and average pooling with no class token. All other settings keep the class token and learned positional embeddings. Mimetic and Impulse assume the standard $QK^\top$ attention parameterization and are therefore incompatible with TTT.

Setting	Copied components	Class token	Pos. embedding
Random	–	random	random
Mimetic	–	random	random
Impulse	–	random	random
Copy	Attention	random	random
MLPCopy	MLPs	random	random
FullCopy	all Softmax weights	copied	copied
Random w/ Distill	–	random	random
Copy w/ Distill	Attention	random	random
MLPCopy w/ Distill	MLPs	random	random
FullCopy w/ Distill	all Softmax weights	copied	copied
Softmax	all Softmax weights	copied	copied

Table B: **Details of datasets and training schedules.**

Dataset	Classes	Train	Test / Val	Epochs	Warm-up
ImageNet-1K (Deng et al., 2009)	1,000	1,281,167	50,000	300	50
CIFAR-10 (Krizhevsky, 2009)	10	50,000	10,000	300	50
CIFAR-100 (Krizhevsky, 2009)	100	50,000	10,000	300	50
STL-10 (Coates et al., 2011)	10	5,000	8,000	300	50
Food (Bossard et al., 2014)	101	75,750	25,250	300	50
Flowers (Nilsback & Zisserman, 2008)	102	1,020	6,149	600	100
Pets (Parkhi et al., 2012)	37	3,680	3,669	600	100

initialized. For TTT, whose output projection has a different shape, only the rows of the standard QKV are loaded.

- **MLPCopy** directly copies the MLP blocks from the pre-trained Softmax weights, while leaving all other components randomly initialized.
- **FullCopy**[5] copies all pre-trained Softmax weights. Unmatched components which are not in the default Softmax implementation, the two convolutions of MHLA and the inner weights, extra projection rows, and wider output projection of TTT, adopt the default initializations in their corresponding settings.

After initialization, every parameter of each setting is trained jointly on downstream tasks, and none of the student's weights are frozen.

B.3 Details of Evaluation Protocol

Table B summarizes all the datasets used in our Softmax-to-linear transfer experiments. We use the standard train/test splits and report the Top-1 Accuracy of each dataset. All reported results are averaged over 3 random seeds to ensure statistical significance, with the corresponding analysis provided in Sec. C.5. Table C summarizes the training recipes used in our experiments, adapted from Xu et al. (2024), with training schedules of 300 or 600 epochs depending on the dataset scale. All experiments are conducted on NVIDIA A100 GPUs.

[5]Compared with *Copy* or *MLPCopy*, *FullCopy* additionally loads the weights of the remaining components in ViTs, mainly the patch embedding, positional embedding, and class token. We find that weight copying for these components does not yield significant performance gains, as shown in a small component-level ablation in Sec. C.4. We therefore focus on the effect of initializing the attention and MLP weights.

Table C: **Training recipes for transfer datasets and ImageNet-1K.**

Config	Transfer datasets	ImageNet-1K
Optimizer	AdamW	AdamW
Base Learning Rate	2e-3	2e-3
Weight Decay	0.05	0.05
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$
Layer-wise LR Decay	–	–
Batch Size	512	4096
LR Schedule	cosine (min. 1e-6)	cosine (min. 1e-6)
Gradient Clipping	3.0	3.0
Augmentation	RRC, flip, color jitter 0.4	RRC, flip, color jitter 0.4
Label Smoothing	0.1	0.1
Drop Path	0	0 (Tiny, Small); 0.1 (Base)
Input / Eval. Crop	224^2 / center crop 0.9	224^2 / center crop 0.9
Distillation Loss	attention-output MSE	attention-output MSE
Distillation Loss Weight λ	50	50
Random Seeds	3	3

B.4 Details of Attention Distillation

The *w/ Distill* settings follow the Attention Distillation used in Li et al. (2024); Qin et al. (2026b), with a frozen pre-trained Softmax teacher and a per-block distillation loss, but place the loss on the attention outputs instead of the attention maps, as discussed in Sec. 5.1. The training aims to match the student's attention outputs to the teacher's through an auxiliary MSE objective at every block during the downstream fine-tuning, leading to the final training objective as $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{distill}}$. Specifically, for MHLA and TTT, which have no class token, only the patch tokens are matched when computing the loss. As discussed in Sec. 5.4, we use $\lambda = 50$ as the default distillation loss weight across all operators, datasets, and initialization settings.

For the loss ablation in Fig. 5, the map-matching loss is computed with the recorded inputs of the attention module in each block. The teacher map is the Softmax map, while the student map is the row-normalized kernel map for ReLU and ELU, and the signed subtraction-normalized map for InLine. The implementations of MHLA and TTT never materialize an attention map, thus are excluded from this ablation. Inspired by Qin et al. (2026b), since the two losses differ in magnitude, each is swept over its own range: $\lambda \in \{10, 30, 50, 100, 300\}$ for output matching and $\lambda \in \{3 \times 10^3, 10^4, 3 \times 10^4, 10^5, 3 \times 10^5\}$ for map matching, on the three reported datasets.

C Additional Results and Analysis

We provide additional experimental results and analysis to further support the generalizability and robustness of our findings below.

C.1 Additional Results on More Linear Operators

Fig. 4 reports the comparison of the four weight-initialization settings with the ReLU-based linear ViTs. We provide comprehensive results for all five linear operators in Fig. A under the same protocol. As shown, the patterns revealed in Sec. 5 hold consistently for every operator: *Copy* remains near *Random*, while *MLPCopy* alone lifts every operator far above *Random* on every dataset and *FullCopy* yields further consistent gains. Adding the distilled attention additionally lifts the accuracy in nearly every setting: *Copy w/ Distill* exceeds *Random w/ Distill* for almost every operator and dataset, confirming the warm-start effect of the copied attention weights across operators, as discussed in Sec. 5.3. *MLPCopy w/ Distill* lands close to *FullCopy w/ Distill* on every operator and dataset, with the final recipe eventually matching or even surpassing the Softmax recovery target on most datasets.

C.2 Additional Results on More Model Sizes

To show the generalizability of our findings across model sizes, we repeat the comparison in Fig. A for all five linear operators at Tiny size under the same protocol in Fig. B. We further extend the

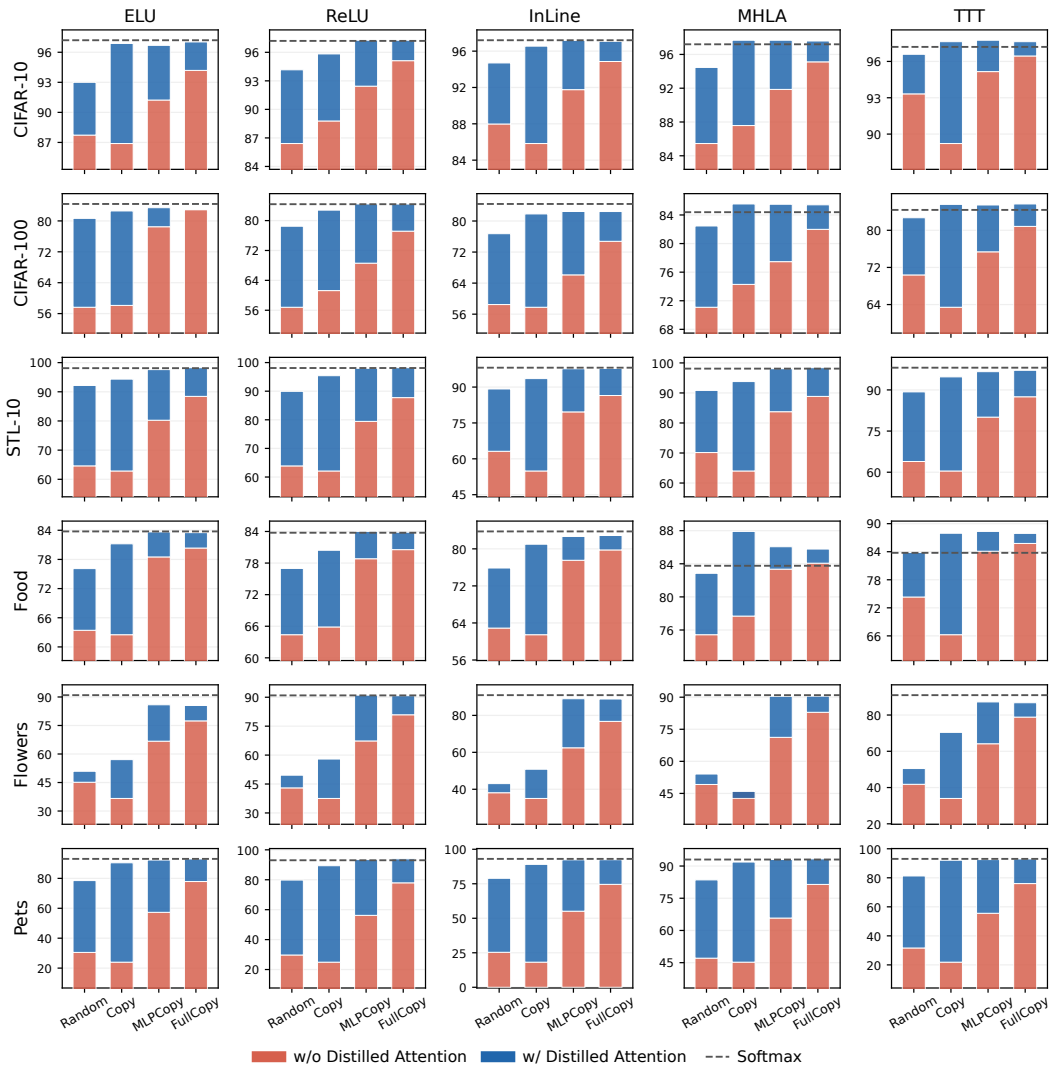

Figure A: **Extended comparison results across linear operators at Small size.** Each linear ViT is initialized under four weight-initialization settings: *Random*, *Copy*, *MLPCopy*, and *FullCopy*; each w/ and w/o attention distillation. Results are reported with the Top-1 Accuracy (%) for all five linear ViTs (columns) on the six transfer datasets (rows) at Small size. In each panel, the red bar is the accuracy of the copy setting alone, the blue block is the additional gain from attention distillation, and the dashed line is the fine-tuned Softmax target.

evaluation to Base size on ImageNet-1K and dense prediction tasks in Sec. C.3. As shown, the patterns revealed in Sec. 5 hold consistently for every operator at the smaller size as well, with smaller margins: *FullCopy* improves over *MLPCopy* on every dataset, distillation lifts nearly every setting, and *MLPCopy w/ Distill* lands close to *FullCopy w/ Distill*, which matches the Softmax target.

C.3 Additional Results on ImageNet and Dense Prediction

As mentioned, we extend the component-level comparison of four weight-initialization settings to ImageNet-1K at Base size and further test the resulting transferred weights as backbones for dense prediction tasks including object detection, instance segmentation, and semantic segmentation. We report results with ReLU and TTT, by training Base models on ImageNet-1K from each of the four weight-initialization settings, w/ and w/o attention distillation from the frozen DeiT-B teacher, resulting in eight transferred linear model weights per operator. Each linear model is then fine-tuned with Mask R-CNN (He et al., 2017) on COCO (Lin et al., 2014) under the 1× schedule, or with

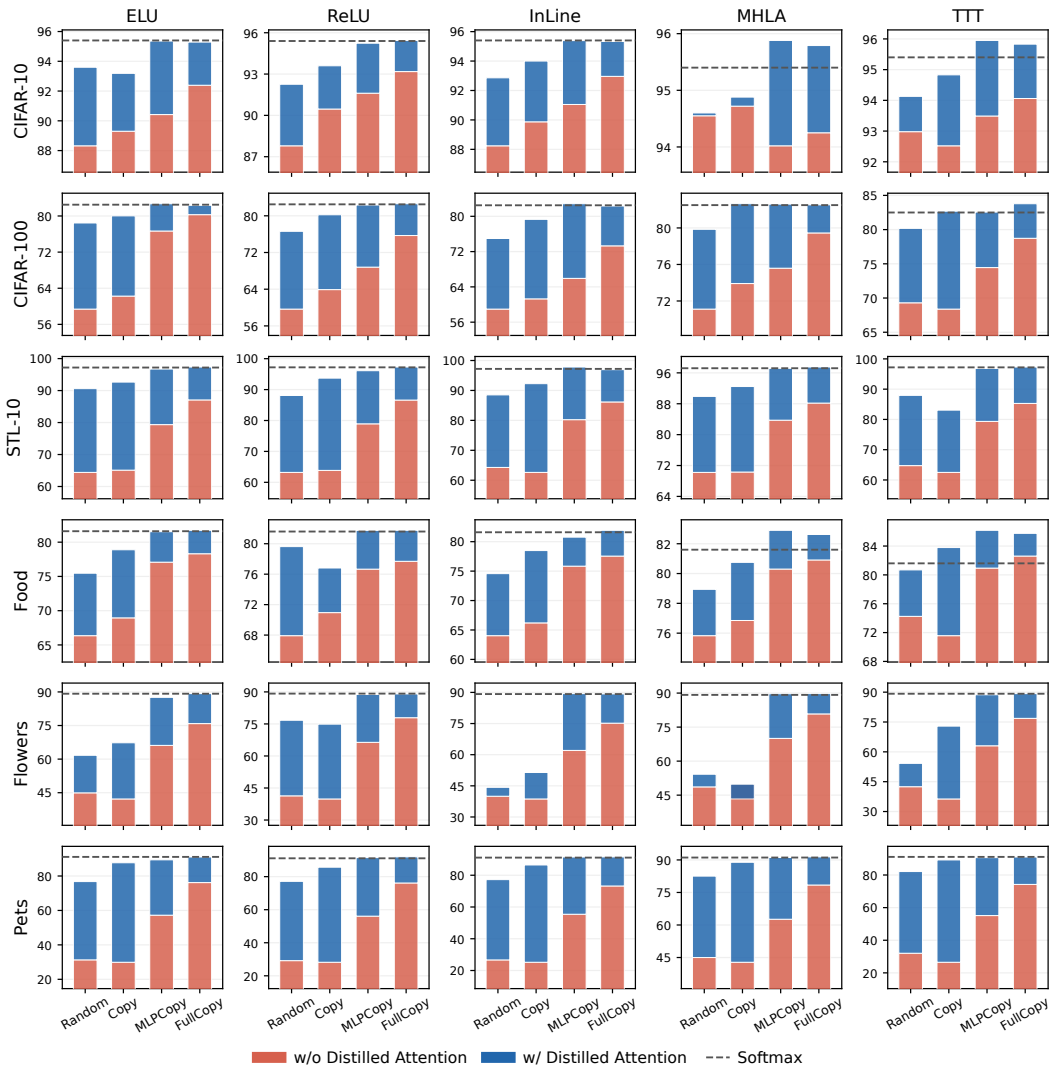

Figure B: **Extended comparison results across linear operators at Tiny size.** Each linear ViT is initialized under four weight-initialization settings: *Random*, *Copy*, *MLPCopy*, and *FullCopy*; each w/ and w/o attention distillation. Results are reported with the Top-1 Accuracy (%) for all five linear ViTs (columns) on the six transfer datasets (rows) at Tiny size. In each panel, the red bar is the accuracy of the copy setting alone, the blue block is the additional gain from attention distillation, and the dashed line is the fine-tuned Softmax target.

UperNet (Xiao et al., 2018) on ADE20K (Zhou et al., 2017) for 160k iterations, following the plain-ViT baseline implementations in Chen et al. (2023).

Tables D and E report the results for ReLU and TTT, respectively. As shown, evaluations on ImageNet-1K and on the dense tasks align consistently with the findings on the six classification datasets in our discussion: *Copy* brings no reliable advantage over *Random*, while *MLPCopy* carries most of the benefit of the pre-trained weights and *FullCopy* further adds a small additional gain, confirming that the attention weights are operator-specific and the MLP weights are operator-agnostic beyond classification. Attention distillation lifts every initialization effectively, with *MLPCopy w/ Distill* generally landing close to *FullCopy w/ Distill*, which closes most of the gap to the Softmax target and matches or even exceeds it.

C.4 Additional Results on Copying the Remaining Components

Compared with *Copy* and *MLPCopy*, *FullCopy* additionally copies the weights of the remaining components beyond the attention and MLP blocks in ViTs. These mainly include the patch embedding,

Table D: **ImageNet-1K and dense prediction results with ReLU at Base size.** All settings adopt transferred weights from DeiT and are then fine-tuned with Mask R-CNN (1×) on COCO and UperNet (160k iters) on ADE20K. We report the Top-1 Accuracy (%) on ImageNet-1K, box and mask AP on COCO, and mIoU with single and multi-scale (+MS) on ADE20K, respectively. w/ Distill with ✓ indicates applying attention distillation during the Softmax-to-linear transfer, and colored deltas indicate gains relative to the same setting w/o distillation. Softmax row refers to the Softmax reference and the plain-ViT baselines of Chen et al. (2023).

	w/	IN-1K	COCO						ADE20K	
	Distill	Acc	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	mIoU	+MS
Softmax		81.8	42.9	65.7	46.8	39.4	62.6	42.0	46.1	47.1
Random		76.5	34.0	56.5	37.6	32.1	55.5	34.4	34.9	35.7
	✓	79.4 +2.9	36.2 +2.2	59.0 +2.5	40.3 +2.7	35.8 +3.7	59.2 +3.7	37.9 +3.5	40.7 +5.8	41.8 +6.1
Copy		75.6	33.3	55.9	37.1	31.6	55.0	33.7	35.2	35.8
	✓	79.9 +4.3	36.9 +3.6	59.3 +3.4	40.7 +3.6	36.0 +4.4	59.6 +4.6	38.2 +4.5	41.5 +6.3	42.1 +6.3
MLPCopy		80.2	41.4	63.9	44.9	36.5	60.1	38.7	43.5	44.4
	✓	81.4 +1.2	42.2 +0.8	64.6 +0.7	**45.9 +1.0**	38.2 +1.7	62.0 +1.9	40.7 +2.0	45.2 +1.7	45.9 +1.5
FullCopy		80.8	41.5	64.1	45.2	36.7	60.4	39.2	43.9	44.6
	✓	**82.0 +1.2**	**42.4 +0.9**	**64.9 +0.8**	**45.9 +0.7**	**39.0 +2.3**	**62.4 +2.0**	**41.3 +2.1**	**45.3 +1.4**	**46.4 +1.8**

Table E: **ImageNet-1K and dense prediction results with TTT at Base size.** All settings adopt transferred weights from DeiT and are then fine-tuned with Mask R-CNN (1×) on COCO and UperNet (160k iters) on ADE20K. We report the Top-1 Accuracy (%) on ImageNet-1K, box and mask AP on COCO, and mIoU with single and multi-scale (+MS) on ADE20K, respectively. w/ Distill with ✓ indicates applying attention distillation during the Softmax-to-linear transfer, and colored deltas indicate gains relative to the same setting w/o distillation. Softmax row refers to the Softmax reference and the plain-ViT baselines of Chen et al. (2023).

	w/	IN-1K	COCO						ADE20K	
	Distill	Acc	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	mIoU	+MS
Softmax		81.8	42.9	65.7	46.8	39.4	62.6	42.0	46.1	47.1
Random		82.2	36.4	58.9	40.3	34.3	57.7	36.2	35.1	35.7
	✓	82.3 +0.1	40.5 +4.1	63.0 +4.1	44.2 +3.9	37.4 +3.1	60.9 +3.2	39.5 +3.3	44.5 +9.4	45.5 +9.8
Copy		81.0	35.4	58.1	39.2	33.5	57.0	35.9	35.1	36.2
	✓	82.2 +1.2	41.0 +5.6	63.6 +5.5	44.7 +5.5	38.1 +4.6	61.9 +4.9	40.3 +4.4	45.2 +10.1	46.2 +10.0
MLPCopy		82.9	42.9	65.2	46.5	38.0	61.5	40.1	44.3	45.3
	✓	83.0 +0.1	43.2 +0.3	65.7 +0.5	46.8 +0.3	39.1 +1.1	62.9 +1.4	**41.5 +1.4**	46.7 +2.4	47.5 +2.2
FullCopy		83.1	43.1	65.9	47.0	38.7	62.3	41.0	45.2	46.0
	✓	**83.2 +0.1**	**43.3 +0.2**	**66.1 +0.2**	**47.6 +0.6**	**39.4 +0.7**	**63.1 +0.8**	**41.5 +0.5**	**47.1 +1.9**	**47.9 +1.9**

Table F: **Statistical significance analysis on ImageNet-1K results.** We report the standard deviation of the Top-1 Accuracy (%) on ImageNet-1K results over 3 random seeds.

	ReLU				TTT			
	Random	Copy	MLPCopy	FullCopy	Random	Copy	MLPCopy	FullCopy
Acc. (w/o Distill)	76.5	75.6	80.2	80.8	82.2	81.0	82.9	83.1
± std	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1
Acc. (w/ Distill)	79.4	79.9	81.4	82.0	82.3	82.2	83.0	83.2
± std	0.2	0.2	0.2	0.2	0.1	0.2	0.1	0.1

positional embedding, and class token. One natural hypothesis is that the performance gain of *FullCopy* also stems significantly from loading these components. We therefore conduct a component-level ablation in Table G to isolate the contribution of loading these three components. Specifically, we gradually remove the weight loading of these components from *FullCopy*: first the class token (Variant 1), then the positional embedding (Variant 2), and finally the patch embedding (Variant

Table G: **Ablation on copying the remaining components.** Starting from *FullCopy*, we gradually remove the weight loading of the class token, positional embedding, and patch embedding, while all attention and MLP weights remain copied. We report the Top-1 Accuracy (%) on ImageNet-1K with ReLU at Base size, with colored deltas relative to the *Random* baseline.

	Weight Copying					Accuracy	
	Attn.	MLP	Patch emb.	Pos. emb.	Cls. token	w/o Distill	w/ Distill
FullCopy	✓	✓	✓	✓	✓	80.8 +4.3	82.0 +2.6
Variant 1	✓	✓	✓	✓		80.7 +4.2	81.8 +2.4
Variant 2	✓	✓	✓			80.6 +4.1	81.9 +2.5
Variant 3	✓	✓				80.7 +4.2	81.9 +2.5
MLPCopy		✓				80.2 +3.7	81.4 +2.0
Copy	✓					75.6 -0.9	79.9 +0.5
Random						76.5	79.4

3), while keeping all attention and MLP weights copied. The removed components are randomly initialized. As shown, removing the weight copying for these components does not yield significant accuracy drops, compared with the substantial benefits from loading the attention and MLP weights. Thus, we focus only on the effect of initializing the attention and MLP weights in our discussion.

C.5 Additional Results on Statistical Significance

We further provide the statistical significance analysis of the ImageNet-1K results reported in Tables D and E. We report the standard deviation of the Top-1 Accuracy under both w/ and w/o distillation across 3 random seeds in Table F.

D Additional Interpretation of Softmax-to-Linear Transfer

We provide a functional interpretation of our component-level findings in Secs. 4 and 5: why the Softmax attention weights are operator-specific, why the MLP weights are operator-agnostic, and why copying the latter and distilling the former are complementary. In the discussion below, we omit the block index ℓ, the scaling factor $1/\sqrt{d_k}$, and the residual connection in MLPs in the formulation for simplicity.

D.1 Preservation under Weight Copying

Pre-trained Softmax attention weights are optimized jointly with the Softmax computation operator during the pre-training stage. For such fixed projection weights Q, K, and V, replacing the Softmax operator in Equation (1) with a linear-complexity alternative via a kernel operator as in Equation (2) changes the output: $f_{\text{linear}} \neq f_{\text{softmax}}$ in general, as widely discussed in previous studies on linear attention (Katharopoulos et al., 2020; Zhang et al., 2024; Han et al., 2024), because the Softmax map $\mathrm{softmax}(QK^\top)$ and the normalized kernel map

$$A_\phi = \mathrm{diag}\big(\phi(Q)\phi(K)^\top \mathbf{1}_N\big)^{-1}\phi(Q)\phi(K)^\top, \tag{A}$$

are computed with different formulations although using the same Q and K. Simply copying the projection weights therefore does not preserve the pre-trained attention function, which is the functional meaning of the Softmax attention weights being *operator-specific*. In addition, when the remaining components are randomly initialized, as in *Copy*, the inputs to the copied attention projections also differ from those seen during pre-training, leading to a further mismatch in the input features on top of the replaced operator.

MLPs have the opposite property: the architecture remains unchanged across the Softmax-to-linear transfer, and the token-wise computation is fully determined by the parameters. Copying the MLP weights therefore preserves this computation exactly for any input, which is the functional meaning of the MLP weights being *operator-agnostic*. What copying does not guarantee is that the student MLP receives the same input as the teacher MLP, but this input is affected by the output from the attention module in the same transformer block.

D.2 Why Copying MLPs and Distilling Attention Are Complementary

Let z_s and z_t denote the inputs to the corresponding MLPs of the linear student and the Softmax teacher, and let M_s and M_t denote the two corresponding MLP computations, respectively. Their output difference then can be decomposed as:

$$M_s(z_s) - M_t(z_t) = \underbrace{M_s(z_s) - M_t(z_s)}_{\text{computation mismatch}} + \underbrace{M_t(z_s) - M_t(z_t)}_{\text{input mismatch}}. \tag{B}$$

At initialization, *MLPCopy* removes the first term with the identical M_s and M_t. The second term remains, which is driven mainly by the attention operator replaced from Softmax to linear. Attention distillation targets this term directly: by matching the outputs of the corresponding attention modules, it supervises this input mismatch and thus closes the gap through the auxiliary training objective. The two transfers thus address the two terms of Equation (B): copying preserves the computation that stays the same, and distillation recovers the computation that the attention operator changes.

E Limitations

Our study is empirical and its scope is bounded in the following three ways. First, all main Softmax-to-linear transfer experiments use ImageNet-1K DeiT pre-trained weights as the source, while other Softmax weights are only evaluated in the robustness check of Sec. 4.3. Thus, the effectiveness of the final recipe is validated for one teacher family, although we expect it to hold across different pre-trained weights. Second, the proposed recipe for Softmax-to-linear transfer is not a training-free adaptation: it provides an initialization strategy for better downstream performance, running with one auxiliary training objective along the fine-tuning. Third, the interpretation in Sec. D is based on and consistent with the empirical observation, rather than a formal derivation. A complete formal theory of Softmax-to-linear Transfer or Attention Transfer remains an important direction for future work.