SPD: SKELETON-PRESERVING DISTILLATION FROM QUADRATIC TO LINEAR-TIME VISION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The accuracy of Vision Transformers relies on large-scale pretraining with a quadratic-cost attention mixer. Subquadratic vision backbones avoid this cost at inference but are usually pretrained from scratch. We propose *Skeleton-Preserving Distillation* (SPD), which converts a pretrained ViT into a linear-time model by reusing its pretrained weights: the student inherits every component of the teacher except the quadratic mixer. We apply SPD to *MambaFormer*, our ViT-based backbone with a bidirectional Mamba2 mixer in place of attention, distilled from ViT teachers whose classifier path it can inherit. Using only 10% of ImageNet-1K, MambaFormer-L recovers 98% of its ViT-L teacher's accuracy at $18.6\times$ less compute than re-training the teacher, and 99.7% with full data. With a CLIP ViT-B/16 teacher, MambaFormer-B reaches 84.25 top-1 vs. 84.30 for ViT-Linearizer at $5.5\times$ less compute, and reaches 82.26 from only 10% of the data at $20.6\times$ less. Trained from scratch on labels with $3.0\times$ more compute (teacher pretraining excluded), the same model reaches 67.75. SPD also applies to semantic segmentation, where the student stays within 0.5 mIoU of the teacher on ADE20K and Cityscapes (single-scale evaluation), beats a no-teacher baseline by 2.1–2.4 mIoU, and needs $1.4\times$ fewer inference FLOPs on Cityscapes. With fused scan kernels, MambaFormer-B matches ViT-B's throughput at 224^2 (batch 32) and is $1.55\times$ faster at 1200^2 on an H100.

1 INTRODUCTION

Vision Transformers (Dosovitskiy et al., 2021) are widely used for image classification and dense prediction, but their public pretrained checkpoints (hundreds to thousands of EFLOPs of ImageNet-21K, CLIP, and DINOv2 pretraining) were trained with an architecture whose self-attention mixer (Vaswani et al., 2017) scales as $\mathcal{O}(L^2)$ in sequence length. At classification resolutions this overhead is moderate; at semantic-segmentation resolutions, the attention term becomes a large share of the forward pass. Subquadratic mixers (structured state-space models (Gu et al., 2022; Gu & Dao, 2024; Dao & Gu, 2024), linear attention (Katharopoulos et al., 2020)) avoid the quadratic cost, but vision instantiations have so far required either ImageNet-21K-scale corpora (Ridnik et al., 2021) or substantial from-scratch training to match a ViT on ImageNet-1K (Zhu et al., 2024; Liu et al., 2024), which repeats the pretraining cost. We seek a data-efficient distillation recipe that turns a pretrained ViT into a linear-time student while closely recovering the teacher's accuracy.

We propose *Skeleton-Preserving Distillation* (SPD): a distillation method in which the student is co-designed so that every non-mixer component of the teacher (patch and positional embeddings, LayerNorms, MLPs, CLS tokens, and the classifier head) is inherited from the teacher; the inherited skeleton (patch and positional embeddings, LayerNorms, MLPs) is then kept frozen throughout the distillation pipeline, while the CLS tokens and classifier head are fine-tuned end-to-end only in the final stage. Because the inherited skeleton is already aligned with the teacher, training concentrates on the mixer, which we expect underlies the recipe's data efficiency; freezing saves compute and is optional (Section 6). We apply SPD to *MambaFormer*, a ViT-skeleton backbone whose self-attention is replaced by a bottlenecked bidirectional Mamba2 mixer, with the whole model parameter-matched to the teacher within $\sim 10\%$, with a double-CLS initialisation that starts from the teacher's CLS embedding (Section 3.1).

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.

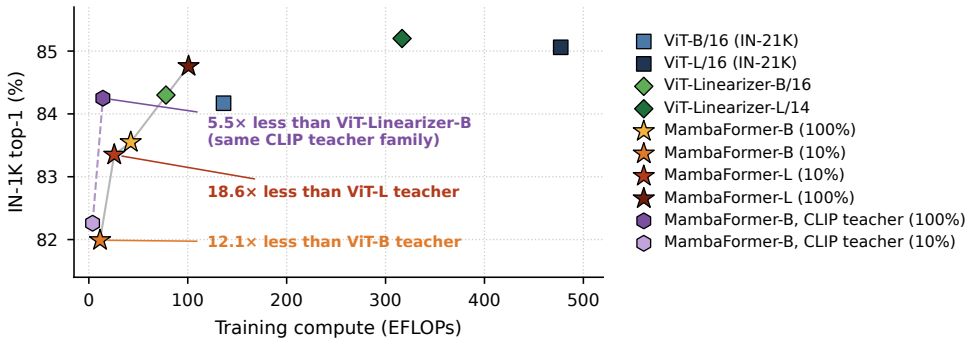


Figure 1: **Training compute vs. IN-1K top-1.** Distilled MambaFormer students (stars: IN-21K teachers, 384^2; hexagons: CLIP ViT-B/16 teacher, 224^2) sit to the left of their ViT teachers and of ViT-Linearizer (Wei & Chellappa, 2025) (diamonds) at near-teacher top-1 (Section 4).

We implement this method as a three-stage curriculum, with a stage structure inspired by MOHAWK (Bick et al., 2024) for causal language models, that progressively unfreezes only what has to be aligned: **Stage 1** aligns the mixer's implicit attention matrix with the teacher's per-block attention maps, **Stage 2** matches the per-block attention outputs, and **Stage 3** trains the mixer, CLS tokens, fusion projection, and classifier head end-to-end under soft-KL to the teacher's logits. On classification, MambaFormer-L distilled on 10% of ImageNet-1K, without ImageNet-21K images, recovers 98.0% of its ViT-L teacher's top-1 (83.35 vs. 85.06) at $18.6\times$ less compute than re-training the teacher (Figure 1), and 99.7% (84.76) with full data. With a CLIP ViT-B/16 teacher from the same pretraining as that of ViT-Linearizer (Wei & Chellappa, 2025) but a stronger IN-1K fine-tune, MambaFormer-B reaches 84.25 vs. 84.30 top-1 at $5.5\times$ less distillation compute. On segmentation, the 10%-distilled MambaFormer-L replaces the backbone of a SETR-PUP teacher, inherits its head, and is aligned by a two-stage curriculum; under single-scale sliding-window evaluation it is within 0.10 mIoU of the teacher on ADE20K and 0.49 mIoU on Cityscapes (after a label-only fine-tune), and beats a no-teacher baseline with the same initialisation by 2.1 mIoU on ADE20K. On Cityscapes it needs $1.44\times$ fewer FLOPs under the 1024 sliding window ($1.96\times$ for one full-image forward pass), SETR-PUP head included (Figure 3a and Section 5).

2 Related Work

Recurrent and state-space vision backbones. Recurrent-style vision architectures replace the self-attention blocks of Vision Transformers with recurrent or state-space modules while largely preserving the ViT macro-architecture (Alkin et al., 2025; Duan et al., 2025; Li et al., 2024; Ren et al., 2025; Yang et al., 2024; Zhu et al., 2024), often with bidirectional scanning or token reordering to supply the global context that a causal scan lacks (Zhu et al., 2024). Hierarchical and hybrid designs extend them to multi-scale modelling (Hatamizadeh & Kautz, 2025; Liu et al., 2024). Several of these models are competitive with ViTs when trained from scratch on ImageNet-1K (Xiao et al., 2025; Shi et al., 2025; Hatamizadeh & Kautz, 2025), but most are hierarchical or convolution-augmented designs that cannot inherit a pretrained ViT's weights or its task heads, and each is pretrained from scratch. We do not propose a new recurrent variant; we transfer the representations of a pretrained ViT into one.

Cross-architecture distillation. Knowledge distillation (Hinton et al., 2015) transfers a teacher's function to a student without repeating its pretraining, including across heterogeneous architectures (Yang et al., 2025); DeiT (Touvron et al., 2021) showed that Transformers can inherit knowledge from CNN teachers. Transformer-to-recurrent and Transformer-to-state-space distillation has since been studied mainly for language and multimodal models (Kasai et al., 2021; Mercat et al., 2024; Wang et al., 2024; Liao et al., 2025; Zhang et al., 2025; 2024), where recurrent students inherit a compatible left-to-right inductive bias; ViT-Linearizer (Wei & Chellappa, 2025) is the closest prior work that distils a pretrained quadratic ViT into a linear-time vision student and evaluates it at ImageNet scale, and we compare against it with a teacher from the same CLIP pretraining (Section 4). Wang et al. (2025b) distil DeiT-Tiny into Vim-Tiny: at 10% of ImageNet their attention bridge reaches 49.2%

top-1 and a MOHAWK-style baseline 45.1%. Vim blocks have no MLP, so the student cannot inherit the teacher's non-mixer weights, on which SPD relies; in image generation, initialising the non-Mamba weights from the DiT teacher is the largest single step in the DiT-to-Mamba ablation of Yao et al. (2025).

The closest methodological precedent is MOHAWK (Bick et al., 2024), which distils a causal Transformer language model into a Mamba2 student through matrix orientation, hidden-state alignment, and weight transfer with knowledge distillation. MOHAWK already transfers the non-mixer weights and reports that freezing everything but the mixer is a viable, slightly lossy option, while its default final stage trains the full student. SPD inherits the staged per-block structure and differs in what it adds and measures: the frozen skeleton is the default at every stage and its cost is quantified in vision (Section 6); a double-CLS initialisation starts both scan directions from the teacher's CLS embedding; Stage 2 keeps the attention-map term as a regulariser against a failure mode we hypothesise (Section 3.2); and the recipe is transferred a second time to dense prediction with an inherited head (Section 5). The bidirectional mixer, the augmentation stack, and the cosine temperature schedule are vision adaptations; Appendix N attributes every component with its evidence.

3 METHOD

This section details the *Skeleton-Preserving Distillation* method of Section 1. Section 3.1 specifies the *MambaFormer* architecture and the initialisation scheme that copies every non-mixer weight of the teacher. Section 3.2 gives the three-stage classification curriculum, adapted to vision from MOHAWK (Bick et al., 2024). Section 3.3 applies the same method a second time to obtain a segmentation student without retraining a backbone or training a head from scratch. Section 3.4 collects implementation details.

3.1 MAMBAFORMER ARCHITECTURE

MambaFormer is a minimal modification of a ViT: it keeps the entire ViT skeleton (patch and positional embeddings, the residual LayerNorm→Mixer→LayerNorm→MLP block, and the classifier head) and replaces only the self-attention mixer with a bidirectional Mamba2 module (BIMAMBA2; Figure 2). SSMs are inherently causal: a single left-to-right scan lets each position attend only to earlier tokens, which is unsuitable for vision where global context is required. BIMAMBA2 resolves this by running two independent Mamba2 instances, one forward and one backward, over the token sequence and combining their per-token outputs (Equations (9) and (10) in Appendix J). The Base and Large tiers use a bottlenecked inner width $d_m = D/2$ with expansion factor $E = 1$; the only architectural choice we tune, it parameter-matches the ViT teacher within $\sim 10\%$ (Appendix J).

To carry the teacher's classification path over as well, we append a second CLS token at the end of the patch sequence (the "double-CLS" configuration). Under a single CLS at position 0, the forward SSM output at the CLS position depends only on the CLS itself before any patches are read, so only the backward scan can drive the prediction. Placing one CLS at each end of the sequence lets each scan direction read all patches before reaching its own CLS, and the two CLS outputs are then fused by a $\mathbb{R}^{2D} \to \mathbb{R}^{D}$ projection. Both CLS tokens are initialised at half the teacher's single CLS value and this fusion projection is initialised as a sum, so the two tokens together start from the teacher's CLS embedding. Combined with verbatim transfer of the patch and positional embeddings, LayerNorms, MLPs, and classifier head, this lets Stages 1 and 2 of Section 3.2 align only the replaced mixer. The same construction applies to a CLIP ViT teacher, whose stem carries one extra pre-transformer LayerNorm; the student inherits it as well and nothing else changes.

3.2 THREE-STAGE CLASSIFICATION CURRICULUM

The classification recipe is a three-stage curriculum (Figure 4) adapted to vision from MOHAWK (Bick et al., 2024), originally proposed for causal language-model distillation. The staged structure allows the soft-KL signal in Stage 3 to act on a student whose mixer is already aligned with the teacher's per-block computation; the inherited skeleton (patch and positional embeddings, LayerNorms, MLPs) stays frozen across all three stages, which saves training compute at a small cost in top-1 (Section 6). Stages 1 and 2 are computed per block with teacher-provided inputs (the teacher's embeddings and the previous teacher block's output) and their losses stop at the mixer

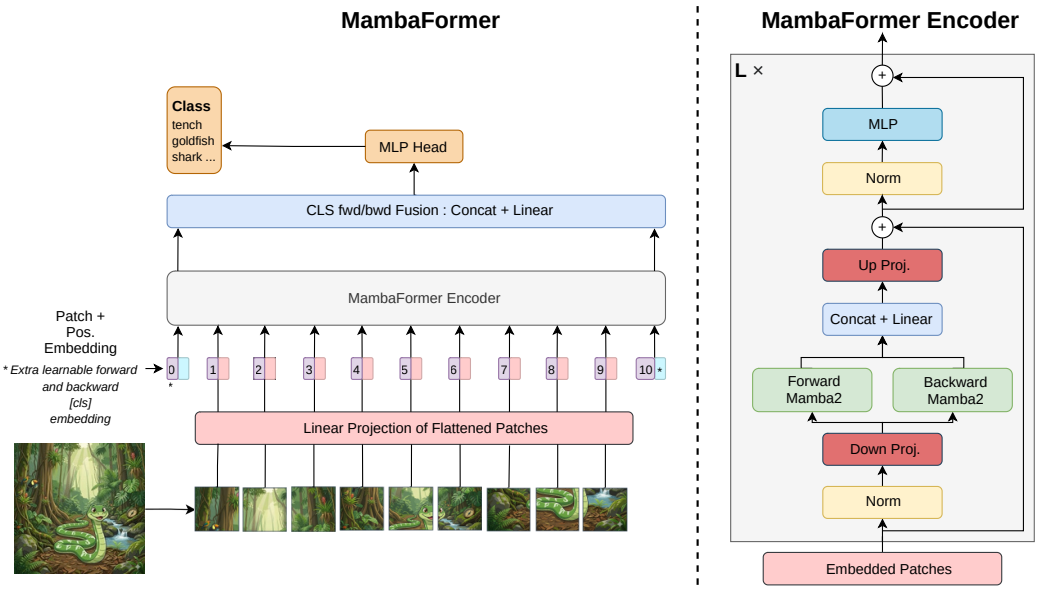


Figure 2: MambaFormer block. The ViT residual skeleton (LN→Mixer→LN→MLP) is preserved verbatim and inherited from the teacher; only the self-attention mixer is replaced, by a bidirectional Mamba2 module (BIMAMBA2) that runs two independent forward/backward Mamba2 scans and combines their per-token outputs. The BiMamba2 mixer uses a bottlenecked inner width ($d_m = D/2$, $E = 1$) that keeps the whole model within $\sim 10\%$ of the teacher's parameter count; the mixer itself is trained, not weight-transferred.

output, so every block can be aligned in parallel and, apart from the pre-mixer LayerNorms, the inherited skeleton is outside their gradient path; the freeze pattern binds only in the end-to-end Stage 3. Total epochs are split across stages on the same dataset; augmentation differs per stage (Table 6). Qualitative mixer-attention maps for the teacher and for the student at the end of each stage are shown in Appendix C.

Stage 1: Attention Map Alignment. *Trainable:* the bottleneck `down_proj` and the Mamba2 `in_proj`, plus the SSM decay parameters in the 10% runs (Table 6); the convolution stays at its identity initialisation. Stage 1 is a short (5–20 epoch) warm-up. The Mamba2 State Space Duality (SSD) (Dao & Gu, 2024) shows that the SSM computation can be expressed as a masked attention matrix $\mathbf{M} = \mathbf{\Lambda} \odot (CB^\top)$, where B, C are the input-dependent SSM input and output matrices and $\mathbf{\Lambda}$ is a lower-triangular mask derived from the SSM decay factors. This recovered attention analogue enables a direct map-to-map distillation against the teacher's softmax attention maps.

Let $\mathbf{A}^t_\ell, \mathbf{A}^s_\ell \in \mathbb{R}^{n\times n}$ denote the teacher's softmax attention and the student's non-negative SSD attention analogue (built from B, C and the decay; Appendix B) at layer ℓ, and let $\mathcal{I}_{\text{cls}} = \{0, n-1\}$ and $\mathcal{I}_{\text{patch}} = \{1, \dots, n-2\}$ index the two CLS positions and the patch positions of the double-CLS sequence, respectively. We define three per-layer alignment terms:

$$\mathcal{L}^{\text{cls}}_\ell = \tfrac{1}{2}\Big(\big\|\mathbf{A}^s_\ell[:, \mathcal{I}_{\text{cls}}] - \mathbf{A}^t_\ell[:, \mathcal{I}_{\text{cls}}]\big\|_F + \big\|\mathbf{A}^s_\ell[\mathcal{I}_{\text{cls}}, :] - \mathbf{A}^t_\ell[\mathcal{I}_{\text{cls}}, :]\big\|_F \Big), \tag{1}$$

$$\mathcal{L}^{\text{patch}}_\ell = \big\|\mathbf{A}^s_\ell[\mathcal{I}_{\text{patch}}, \mathcal{I}_{\text{patch}}] - \mathbf{A}^t_\ell[\mathcal{I}_{\text{patch}}, \mathcal{I}_{\text{patch}}]\big\|_F, \tag{2}$$

$$\mathcal{L}^{\text{prob}}_\ell = \text{KL}\Big(\mathbf{A}^t_\ell \,\big\|\, \widetilde{\mathbf{A}}^s_\ell\Big), \tag{3}$$

where $\widetilde{\mathbf{A}}^s_\ell$ is $\mathbf{A}^s_\ell$ row-normalised and the KL is computed row-wise and averaged; all terms are per head (map construction and initialisation in Appendix B). The total Stage 1 loss sums these over all N_{blk} blocks:

$$\mathcal{L}_1 = \sum_{\ell=1}^{N_{\text{blk}}} \Big[\lambda_{\text{cls}}\,\mathcal{L}^{\text{cls}}_\ell + \lambda_{\text{patch}}\,\mathcal{L}^{\text{patch}}_\ell + \lambda_{\text{prob}}\,\mathcal{L}^{\text{prob}}_\ell\Big], \quad (\lambda_{\text{cls}}, \lambda_{\text{patch}}, \lambda_{\text{prob}}) = (0.4,\, 0.4,\, 0.2). \tag{4}$$

Stage 1 initialises Stage 2; dropping it leaves final top-1 unchanged (Section 6).

Stage 2: Attention Output Alignment. *Trainable:* the mixer and bottleneck projections, and in the 10% runs the full BiMamba2 mixer; the skeleton stays frozen. The mixer is trained to align the student's per-block output with the teacher's multi-head attention (MHA) output. Let $\mathbf{m}_\ell^s, \mathbf{m}_\ell^t \in \mathbb{R}^{n\times D}$ be the student BiMamba2 and teacher MHA outputs at layer ℓ (both pre-residual; the teacher's CLS row is halved and duplicated as for the maps). The Stage 2 loss is:

$$\mathcal{L}_2 = \sum_{\ell=1}^{N_{\text{blk}}} \Big[\gamma\,\mathrm{MSE}(\mathbf{m}_\ell^s,\,\mathbf{m}_\ell^t) \;+\; (1{-}\gamma)\,\big\|\mathbf{A}_\ell^s - \mathbf{A}_\ell^t\big\|_F\Big]. \tag{5}$$

with $\gamma = 0.7$. The second term re-applies the Stage 1 attention-map alignment as an explicit regulariser. Without it, an MSE-only loss would let the student match the mixer output through any internal mechanism: the optimiser can absorb a mismatched attention pattern into the `out_proj` layers, producing the correct $\mathbf{m}_\ell^s$ while bypassing the SSD attention structure learned in Stage 1. Anchoring the attention map is meant to prevent this shortcut and keeps the Stage 1 alignment intact while the MSE term drives the new degrees of freedom toward matching the MHA output. At a matched epoch budget, removing this stage costs 12.6 pp top-1, the largest drop (Section 6).

Stage 3: Logit Distillation. *Trainable:* the full BiMamba2 mixer and bottleneck, the double-CLS tokens, the fusion projection, and the classifier head; the patch and positional embeddings, LayerNorms, and MLPs remain frozen. The trainable subset is fine-tuned end-to-end against the teacher's softened output distribution under pure soft-KL with no cross-entropy term and no labels:

$$\mathcal{L}_3 = T(t)^2 \cdot \mathrm{KL}\big(\mathrm{softmax}\left(z^t/T(t)\right) \,\big\|\, \mathrm{softmax}\left(z^s/T(t)\right)\big)\,, \tag{6}$$

where z^s, z^t are the student and teacher logits and $T(t)$ is the distillation temperature at epoch t. Temperature follows a cosine schedule $T{:}\,5\to1.5$ ($4\to1.5$ for L 10%), held at its initial value during warm-up and at its final value over the last 10% of epochs. On the 10% subsets the full augmentation stack (RandAugment (Cubuk et al., 2020), RandomErasing (Zhong et al., 2020), MixUp (Zhang et al., 2018)/CutMix (Yun et al., 2019)) is used; full-data runs disable it. An exponential moving average (EMA) shadow (Polyak & Juditsky, 1992) of the live weights is maintained during training; we report the final live weights (Appendix H).

3.3 Segmentation: the method applied a second time

For semantic segmentation we apply SPD a second time. The teacher is SETR-PUP (progressive-upsampling decoder) (Zheng et al., 2021) with a ViT-L/16 backbone; the student replaces the ViT-L backbone with the 10%-distilled MambaFormer-L of Section 3.2 used unmodified, and *inherits* the SETR-PUP head from the segmentation teacher rather than training one from scratch. Only the mixer-induced mismatch between the two backbones has to be aligned. The recipe mirrors the classification curriculum but is collapsed into two alignment stages.

Stage 1 (HA-seg, hidden-state alignment): backbone alignment. The MambaFormer-L backbone is aligned against the ViT-L SETR-PUP teacher backbone by channel-wise distillation (Shu et al., 2021) on multi-level backbone features; the inherited SETR-PUP head is held frozen in eval mode. This is the dense-prediction analogue of Stage 2 in classification.

Stage 2 (Whole-Model Distillation). Backbone and head are fine-tuned jointly with the label cross-entropy, a logit-KL loss to the teacher's outputs ($T=2$, weight α), a channel-wise distillation feature loss ($T=4$, weight β), and a 0.4 auxiliary-head term. On Cityscapes, memory limits distillation to 768 crops; a teacher-free fine-tune at crop 1024 follows (Appendix D).

We use SETR-PUP (Zheng et al., 2021), a lightweight head (four conv+BatchNorm+ReLU+$2\times$ upsample stages and a 1×1 classifier), so mIoU differences between the ViT-L teacher and the MambaFormer student can be attributed mainly to the backbone, as the head has limited capacity to compensate for backbone differences.

3.4 Implementation details

Classification runs at the teacher's resolution: 384^2 for the IN-21K ViTs, 224^2 for CLIP ViT-B/16 (`vit_base_patch16_clip_224.openai_ft_in1k` (Wightman, 2019; Cherti et al., 2023)).

Table 1: ImageNet-1K val top-1 / top-5 (single crop). Distilled MambaFormer students vs. their ViT teachers and the ViT-Linearizer baseline (Wei & Chellappa, 2025). "Train (EFLOPs)" is total student training compute (Appendix J); rows with a CLIP teacher (Radford et al., 2021) exclude its WIT-400M pretraining. CLIP teacher top-1 is given in Table 2. CLIP students have 95.2M parameters instead of 95.5M because their positional embedding covers 224^2 instead of 384^2 inputs. Bold: students distilled from 10% of IN-1K.

Student	Teacher	Subset	Top-1	Top-5	Params (M)	Train (EFLOPs)
Teachers (IN-21K → IN-1K 384 fine-tune)						
ViT-B/16	—	1,281k	84.17	97.17	86.9	136.19
ViT-L/16	—	1,281k	85.06	97.36	304.7	477.09
ViT-Linearizer (Wei & Chellappa, 2025) (224^2, CLIP pretraining excluded)						
ViT-Linearizer-B/16	CLIP ViT-B/16	1,281k	84.30	—	99.0	77.96
ViT-Linearizer-L/14	CLIP ViT-L/14	1,281k	85.20	—	346.0	316.68
MambaFormer students (ours, IN-21K teachers, 384^2)						
MambaFormer-B (100%)	ViT-B/16	1,281k	83.55	96.94	95.5	42.35
MambaFormer-B (10%)	**ViT-B/16**	**128k**	**81.99**	**96.36**	**95.5**	**11.29**
MambaFormer-L (100%)	ViT-L/16	1,281k	84.76	97.23	333.1	100.80
MambaFormer-L (10%)	**ViT-L/16**	**128k**	**83.35**	**96.88**	**333.1**	**25.69**
MambaFormer students (ours, CLIP ViT-B/16 teacher, 224^2)						
MambaFormer-B (100%)	CLIP ViT-B/16	1,281k	84.25	97.14	95.2	14.18
MambaFormer-B (10%)	**CLIP ViT-B/16**	**128k**	**82.26**	**96.57**	**95.2**	**3.78**

The 10% ImageNet-1K subset is class-stratified (seed 42, 128,066 images); top-1 is reported on the full 50,000-image validation set. We use AdamW (Loshchilov & Hutter, 2019) for every stage; per-stage learning rates, batch sizes, weight decay, gradient clipping, EMA decay, freeze patterns, and augmentation stacks are listed in Tables 6 and 7 of the appendix. Total training compute and per-stage wall-clock for each configuration are reported in Tables 8 and 9; all jobs ran on one 4×H100 node.

4 Classification results

All ImageNet-1K val numbers in this section use the standard 50,000-image val split, single-crop centred eval at the teacher's resolution (384^2 for the IN-21K ViTs, 224^2 for CLIP). FLOPs count one multiply-add as two FLOPs (Appendix J).[1]

Main results. Table 1 reports our students, their teachers, and ViT-Linearizer (Wei & Chellappa, 2025), the closest prior baseline. Our main result is MambaFormer-L distilled on the 10% subset with no IN-21K access: 83.35% top-1, 98.0% of the ViT-L/16 teacher's 85.06% (Base: 97.4%). With full data the gap narrows (99.3% Base, 99.7% Large at 84.76%), and the Large alignment stages converge early (Appendix D). The unchanged recipe also transfers to a CLIP teacher (98.8% / 96.5% recovery at 100% / 10% data).

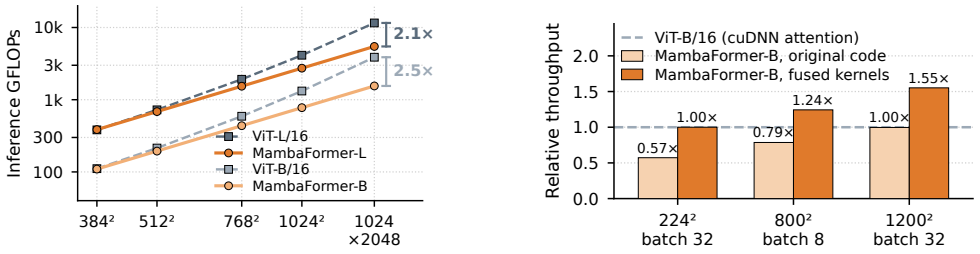
Compute–accuracy Pareto. The recipe is also compute-efficient (Figure 1): against the IN-21K + IN-1K teacher schedules (Steiner et al., 2022), B 100% / B 10% / L 100% / L 10% are 3.2 / 12.1 / 4.7 / 18.6× cheaper (per-stage breakdown in Table 8).

Comparison to ViT-Linearizer. Table 2 separates two kinds of comparison. *Controlled:* with a CLIP ViT-B/16 teacher at the data (100% IN-1K), resolution (224^2), and tier of ViT-Linearizer-B, where patch-16 makes sequence length and FLOPs comparable, MambaFormer-B reaches 84.25% vs. 84.30% at 5.5× less distillation compute, and 82.26% from the 10% subset at 20.6× less. The teachers share CLIP pretraining but are different IN-1K fine-tunes (85.26% vs. 84.70%), so neither metric isolates the method: our student is 0.05 below theirs from a 0.56-stronger teacher, while its gap to its own teacher is larger ($\Delta = -1.00$ vs. -0.40), and we do not claim Δ-parity. *Frontier points:* our IN-21K-teacher students differ from ViT-Linearizer in teacher, corpus, and resolution, so only Δ

[1] Throughout the paper, differences, ratios and recovery percentages are computed from unrounded values, so they can differ by 0.01 from the same quantity computed from rounded table entries.

Table 2: Distillation efficiency vs. ViT-Linearizer (Wei & Chellappa, 2025). *Controlled*: same teacher family, data, resolution, and tier; the teachers are different IN-1K fine-tunes, so absolute top-1 (IN-1K) is the primary metric. *Frontier points*: comparable on Δ only. "Train" is end-to-end student training compute (Appendix J); the last column is the same-tier ratio. n/r: not reported.

Method	Teacher	Student	Δ	Train EFLOPs	Less than ViT-Lin.
Controlled: CLIP ViT-B/16 teacher, 224^2					
ViT-Linearizer-B/16 (100%)	84.70	84.30	-0.40	77.96	—
MambaFormer-B (100%)	85.26	84.25	-1.00	14.18	**5.50**×
MambaFormer-B (10%)	85.26	82.26	-3.00	3.78	**20.62**×
Frontier points: different teachers and resolutions					
ViT-Linearizer-L/14 (100%)	n/r	85.20	—	316.68	—
MambaFormer-B (100%)	84.17	83.55	-0.62	42.35	**1.84**×
MambaFormer-B (10%)	84.17	81.99	-2.17	11.29	**6.91**×
MambaFormer-L (100%)	85.06	84.76	-0.30	100.80	**3.14**×
MambaFormer-L (10%)	85.06	83.35	-1.71	25.69	**12.33**×



(a) Inference GFLOPs vs. input resolution (batch 1). (b) Measured throughput relative to ViT-B/16 (H100).

Figure 3: **Inference cost.** *(a)* ViT FLOPs grow as $\mathcal{O}(L^2)$ and parameter-matched MambaFormers as $\mathcal{O}(L)$; at native Cityscapes resolution (1024×2048) the ViT teachers cost 2.1–2.5× more backbone FLOPs for a single full-image forward pass (1.44× for SETR-PUP-ViT-L, head included, under the 1024 sliding window, Appendix J.5). *(b)* Wall-clock inference on one H100 (bf16, CUDA graph), relative to ViT-B/16 with cuDNN attention: parity at 224^2 (batch 32; 0.82× against FlashAttention-2 ViT-B at batch 1, Table 23), 1.24× at 800^2 and 1.55× at 1200^2 with our fused kernels, timed with distilled weights (Appendix K).

is comparable. At full data, MambaFormer-B/L (-0.62/-0.30) are close to ViT-Linearizer-B (-0.40; no teacher accuracy is reported for L/14) at 1.84×/3.14× less compute; the 10%-data students trade larger gaps (-2.17/-1.71) for 6.91× and 12.33× less. With two data budgets per tier, the gap to the teacher shrinks with more data; we do not test whether it closes.

5 Segmentation Results

The same L 10% IN-1K-distilled MambaFormer-L of Section 4 is used unmodified as the SETR-PUP backbone for both datasets, with no ImageNet-21K images, and is aligned per dataset with the two-stage segmentation curriculum of Section 3.3. We evaluate on ADE20K (Zhou et al., 2017) and Cityscapes (Cordts et al., 2016). Evaluation uses a sliding window with stride $2/3$ of crop; "+ msflip" adds horizontal-flip and multi-scale test-time augmentation (TTA) at scales $\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$.

Comparison to ViT-Linearizer. The closest prior work that reports segmentation, ViT-Linearizer (Wei & Chellappa, 2025), distils Adventurer-B/16 from a CLIP ViT-B/16 teacher and fine-tunes it with an UperNet (Xiao et al., 2018) head (51.3 mIoU on ADE20K at 512^2, 82.0 mIoU on Cityscapes at 512×1024). The like-for-like comparison is on encoder FLOPs at deployment resolution, given in Appendix J.5; absolute mIoU is not directly comparable because decoder, student tier, teacher, and Cityscapes input all differ from our setup.

Table 3: SETR-PUP segmentation, mIoU (%). The students start from the L 10% IN-1K-distilled MambaFormer-L (Table 1) and inherit the SETR-PUP head from the teacher; they are aligned per dataset with HA-seg → Whole-Model distillation (Section 3.3). On Cityscapes, distillation runs at crop 768 (H100 80GB memory budget) and is followed by a teacher-free fine-tune at crop 1024 ("+ HR"), so that teacher and student are evaluated with the same 1024 sliding window. *No teacher*: same backbone and head initialisation and the same total iterations, trained on labels only. All rows use the final checkpoint (a dash: msflip not run). Crop: sliding-window size; mAcc/aAcc in Table 10.

Dataset	Model	Crop	slide	+ msflip	Δ (slide)
ADE20K	ViT-L/16 (teacher)	512	47.91	50.14	—
	MambaFormer-L (SPD)	512	**47.81**	**48.99**	−0.10
	MambaFormer-L (no teacher)	512	45.68	47.91	−2.23
Cityscapes	ViT-L/16 (teacher)	1024	78.99	80.86	—
	MambaFormer-L (SPD + HR)	1024	**78.50**	—	−0.49
	MambaFormer-L (SPD)	768	78.61	80.30	
	MambaFormer-L (no teacher)	768	76.21	—	

mIoU on ADE20K and Cityscapes. Table 3 reports the final mIoU on both benchmarks. On ADE20K (eval at 512, the SETR-PUP standard) the MambaFormer-L student is within 0.10 mIoU of the teacher at single-scale sliding-window evaluation; the gap widens to −1.15 mIoU with msflip TTA (+2.23 for the teacher, +1.17 for the student). On Cityscapes, with the teacher's 1024 sliding window, the student is within 0.49 mIoU (single-scale; 1.21× the teacher's training compute, Appendix G). Trained on labels only with the same initialisation and iteration budget, the student reaches 45.68 on ADE20K (−2.14 vs. SPD) and 76.21 on Cityscapes at crop 768 (−2.40), so the teacher signal contributes beyond the inherited backbone.

6 Ablations

All ablations use the B 10% pipeline, evaluated on the full IN-1K val at 384^2.

Pipeline stage ablation. Table 4 drops one stage at a time at a fixed 200-epoch total budget; epochs from a dropped stage are reassigned to Stage 3. Stage 2 (Attention Output Alignment) has the largest effect: removing it costs 12.59 pp top-1 even with 30 extra Stage-3 epochs. Stage 3 alone on top of raw weight transfer reaches 63.84%, 15 pp above random-init label-CE training and 17 pp below the full pipeline. Weight transfer alone is therefore not sufficient: the mixer needs to be re-aligned against teacher features before soft-KL is applied. Stage 1 does not change final-epoch top-1 (81.23 with it, 81.28 without); we keep it because it initialises Stage 2 on attention-map targets alone at small cost (5–20 epochs). With full data, a supervised from-scratch MambaFormer-B reaches 67.75% vs. 84.25% for the CLIP-distilled student, which uses 3.0× less compute (teacher pretraining excluded); it is a lower bound since that run needed stabilisation (Appendix M).

Freezing the skeleton. Stages 1–2 never place the skeleton in the gradient path (Section 3.2), so we ablate the freeze in Stage 3 (Table 5): embeddings, LayerNorms, and MLPs become trainable from the same Stage-2 checkpoint, all else unchanged. Unfreezing increases top-1 by 0.32 pp at +25.4% student training FLOPs, +13.4% epoch time, and 1.28× peak memory, the compute of 38 additional frozen epochs. When trainable, the skeleton weights change (12–18% relative ℓ_2 drift) without overfitting (train–val gap flat at ∼8 pp; Appendix L), so freezing does not act as a regulariser here.

Recipe ablation. One Stage-3 setting at a time (Appendix P): full augmentation gives the largest gain (+0.77 pp) even with soft-KL targets, and a cosine temperature schedule adds +0.25 pp. Neither ablation uses the final configuration; deltas are not additive onto the headline.

Table 4: Pipeline ablation at a fixed 200-epoch budget on B 10%. WT = weight transfer; S1/S2/S3 = Stages 1/2/3. All distillation rows above the last use the same trainable-parameter policy (the random-init row trains all parameters with label CE); the last row is the final recipe (Table 1).

Configuration	WT	S1	S2	S3	Top-1
Random init (scratch CE)	—	—	—	—	48.47
Stage 3 only	✓	—	—	✓	63.84
Drop S2	✓	✓	—	✓	68.63
Drop S1	✓	—	✓	✓	81.28
Full pipeline	✓	✓	✓	✓	81.23
Full pipeline, final recipe	✓	✓	✓	✓	**81.99**

Table 5: Freezing the inherited skeleton (B 10%, Stage 3, single seed, all else unchanged). Training FLOPs count the weight-gradient term exactly (Appendix L); median epoch time on one 4×H100 node.

Skeleton	Trainable	Top-1 (live / EMA)	FLOPs / img	Epoch time	Memory
Frozen (ours)	37.8M	81.99 / 82.02	260.1 G	146.6 s	8.84 GiB
Unfrozen	95.5M	82.31 / 82.34	326.3 G (+25.4%)	166.2 s (+13.4%)	11.35 GiB

7 Discussion and limitations

Gap to the teacher. The student remains below its teacher in every setting (-0.30 to -3.00 pp top-1; -0.10 / -0.49 mIoU on ADE20K / Cityscapes); our reference point is the teacher, not the state of the art.

Scope. Our claims cover ViT teachers with an inheritable CLS-to-logits path, with two demonstrated regimes: supervised IN-21K and CLIP fine-tuned on IN-1K. Recovery is modestly lower with CLIP (98.8% / 96.5% vs. 99.3% / 97.4% at 100% / 10% data, at 224^2 vs. 384^2). For headless self-supervised teachers (Oquab et al., 2024) the recipe is defined but not validated (Appendix O).

Limitations. *(i)* **Teacher and skeleton.** The recipe presupposes a public, well-trained ViT teacher of the student's size, and a student that preserves the ViT skeleton interface so the weights can be inherited verbatim; hierarchical backbones, other tokenisations, or non-LayerNorm normalisation fall outside it. SPD converts a ViT into a linear-time model of the same size (parameter-matched within $\sim$10%); it does not compress the teacher. The mixer also keeps Mamba2's first-order discretisation, which we do not ablate; Mamba-3 generalises it to a second-order rule (Lahoti et al., 2026), and in Vision Mamba a higher-order discretisation alone raises ADE20K mIoU by up to 4.4 over zero-order hold (Ibrahim et al., 2026). *(ii)* **Untested controls.** We did not run a single-stage joint optimisation of the three losses, a single-CLS student, Stage 2 without the attention-map term ($\gamma = 1$), an IN-1K-only teacher (CLIP's corpus is larger than IN-21K), data fractions below 10%, other distillation objectives, or a random-init student with the full curriculum (Table 4's random-init row also changes the loss). *(iii)* **Single seeds.** All runs use one seed; the $+0.32$, $+0.77$, $+0.25$ and 0.05 pp gaps (Sections 4 and 6) are not established above seed variance.

8 Conclusion

Skeleton-Preserving Distillation reuses the compute already spent on a pretrained ViT to obtain a linear-time student: 98.0% of a ViT-L teacher from 10% of ImageNet-1K at 18.6× less compute (99.7% with full data), 0.05 below ViT-Linearizer-B (84.25 vs. 84.30) at 5.5× less compute from a stronger CLIP teacher, and segmentation within 0.5 mIoU with 1.4× fewer inference FLOPs on Cityscapes (single-scale sliding window). Because the non-mixer weights are inherited, the ViT pretraining does not have to be repeated.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods: writing and debugging parts of the training, evaluation and analysis code, including the fused inference kernels (Appendix K). We have not used generative AI tools to develop the conceptual framework, propose or refine hypotheses, or design the method and experiments; generating synthetic data, mathematical claims and proofs, translation, dataset cleaning and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to write figure scripts, draft parts of the appendix, edit the text for readability, search the related literature and format references. We have reviewed all AI-assisted work: the kernels were validated against the reference implementation (Appendix K), every reported number was checked against the logged results, and every citation against its source. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This is standard image classification and semantic segmentation work; we are not aware of new misuse risks. Cheaper inference at native resolution may lower the cost of deploying dense-prediction models.

REPRODUCIBILITY STATEMENT

The architecture and its initialisation are specified in Section 3.1, the three-stage classification curriculum in Section 3.2, the segmentation curriculum in Section 3.3, and implementation details in Section 3.4. All hyperparameters and per-stage losses are listed in Appendix D; the training-compute accounting, including every FLOP count used for the compute ratios, is in Appendices E and J; datasets, teacher checkpoints and the evaluation protocol are in Appendix I; and the inference kernels, their correctness tests and the timing protocol are in Appendix K. All teachers and datasets are public. Code, configurations and distilled checkpoints will be released upon acceptance.

REFERENCES

Benedikt Alkin, Maximilian Beck, Korbinian Pöppel, Sepp Hochreiter, and Johannes Brandstetter. Vision-LSTM: xLSTM as generic vision backbone. In *International Conference on Learning Representations*, 2025.

Aviv Bick, Kevin Y. Li, Eric P. Xing, J. Zico Kolter, and Albert Gu. Transformers to SSMs: Distilling quadratic knowledge to subquadratic models. In *Advances in Neural Information Processing Systems*, 2024.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Yuchen Duan, Weiyun Wang, Zhe Chen, Xizhou Zhu, Lewei Lu, Tong Lu, Yu Qiao, Hongsheng Li, Jifeng Dai, and Wenhai Wang. Vision-RWKV: Efficient and scalable visual perception with RWKV-like architectures. In *International Conference on Learning Representations*, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Ali Hatamizadeh and Jan Kautz. MambaVision: A hybrid Mamba-Transformer vision backbone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25261–25270, 2025.

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

Fady Ibrahim, Guangjun Liu, and Guanghui Wang. Beyond ZOH: Advanced discretization strategies for Vision Mamba. *arXiv preprint arXiv:2604.20606*, 2026.

Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A. Smith. Finetuning pretrained Transformers into RNNs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive Transformers with linear attention. In *International Conference on Machine Learning*, 2020.

Aakash Lahoti, Kevin Y. Li, Berlin Chen, Caitlin Wang, Aviv Bick, J. Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, 2026.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. VideoMamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, 2024.

Bencheng Liao, Hongyuan Tao, Qian Zhang, Tianheng Cheng, Yingyue Li, Haoran Yin, Wenyu Liu, and Xinggang Wang. Multimodal Mamba: Decoder-only multimodal state space model via quadratic to linear distillation. *arXiv preprint arXiv:2502.13145*, 2025.

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 *Advances in Neural Information Processing Systems*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Jean Mercat, Igor Vasiljevic, Sedrick Keh, Kushal Arora, Achal Dave, Adrien Gaidon, and Thomas Kollar. Linearizing large language models. In *Conference on Language Modeling*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

Sucheng Ren, Xianhang Li, Haoqin Tu, Feng Wang, Fangxun Shu, Lei Zhang, Jieru Mei, Linjie Yang, Peng Wang, Heng Wang, Alan Yuille, and Cihang Xie. Autoregressive pretraining with Mamba in vision. In *International Conference on Learning Representations*, 2025.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K pretraining for the masses. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Yuheng Shi, Mingjia Li, Minjing Dong, and Chang Xu. VSSD: Vision Mamba with non-causal state space duality. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10819–10829, 2025.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5311–5320, 2021.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? Data, augmentation, and regularization in vision Transformers. *Transactions on Machine Learning Research*, 2022.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image Transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Feng Wang, Timing Yang, Yaodong Yu, Sucheng Ren, Guoyizhe Wei, Angtian Wang, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Adventurer: Optimizing vision Mamba architecture designs for efficiency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30157–30166, 2025a.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. The Mamba in the Llama: Distilling and accelerating hybrid models. In *Advances in Neural Information Processing Systems*, 2024.

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.

Guoyizhe Wei and Rama Chellappa. ViT-Linearizer: Distilling quadratic knowledge into linear-time vision models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20737–20747, 2025.

Ross Wightman. PyTorch image models. GitHub repository, 2019. URL `https://github.com/rwightman/pytorch-image-models`.

Chaodong Xiao, Minghan Li, Zhengqiang Zhang, Deyu Meng, and Lei Zhang. Spatial-Mamba: Effective visual state space models via structure-aware state fusion. In *International Conference on Learning Representations*, 2025.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*, pp. 418–434, 2018.

Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J. Crowley. PlainMamba: Improving non-hierarchical Mamba in visual recognition. In *British Machine Vision Conference*, 2024.

Yaoxin Yang, Peng Ye, Weihao Lin, Kangcong Li, Yan Wen, Jia Hao, and Tao Chen. Multi-level decoupled relational distillation for heterogeneous architectures. *arXiv preprint arXiv:2502.06189*, 2025.

Yuan Yao, Yicong Hong, Difan Liu, Long Mai, Feng Liu, and Jiebo Luo. Diffusion Transformer-to-Mamba distillation for high-resolution image generation. In *British Machine Vision Conference*, 2025.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The Hedgehog & the Porcupine: Expressive linear attentions with softmax mimicry. In *International Conference on Learning Representations*, 2024.

Michael Zhang, Simran Arora, Rahul Chalamala, Alan Wu, Benjamin Spector, Aaryan Singhal, Krithik Ramesh, and Christopher Ré. LoLCATs: On low-rank linearizing of large language models. In *International Conference on Learning Representations*, 2025.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6881–6890, 2021.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI Conference on Artificial Intelligence*, 2020.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning*, 2024.

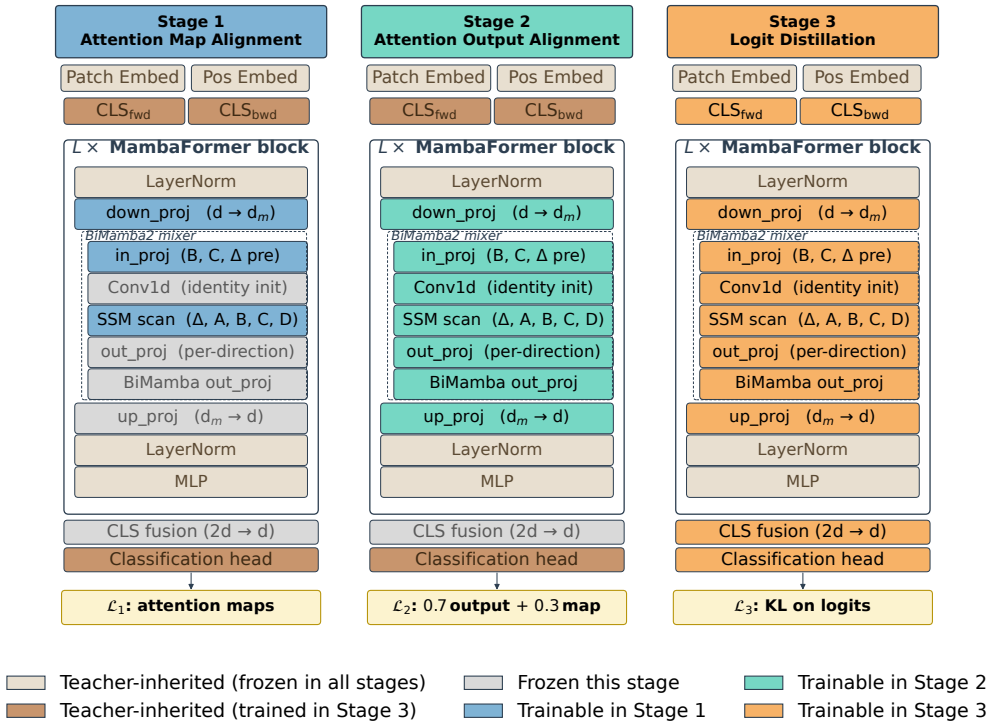


Figure 4: Three-stage classification distillation pipeline, shown with the trainable sets of the 10% runs (Table 6 lists them for every run; the 100% runs train Stage 1 without the decay parameters). Stage 1 (Attention Map Alignment) trains the bottleneck `down_proj`, `in_proj` and the SSM decay parameters against the teacher's per-block attention maps; the convolution stays at its identity initialisation. Stage 2 (Attention Output Alignment) trains the full BiMamba2 mixer, including the convolution and the bottleneck up-projection, to reproduce per-block MHA outputs. Stage 3 (Logit Distillation) additionally trains the double-CLS tokens, the fusion projection and the classifier head under soft-KL on the logits with a cosine temperature schedule. Patch and positional embeddings, LayerNorms and MLPs remain frozen throughout.

A Classification distillation pipeline

Figure 4 illustrates the three-stage classification distillation pipeline. Each stage trains a strictly larger set of student parameters and targets a progressively weaker teacher signal, so that earlier stages provide a clean initialisation for later ones.

B Student initialisation and Stage-1 attention maps

Initialisation. The student copies from the teacher the patch embedding, the positional embedding, every LayerNorm and MLP, and the classifier head. The second CLS token receives a copy of the CLS positional embedding; both CLS tokens are set to half the teacher's CLS token, and the fusion projection $\mathbb{R}^{2d}\to\mathbb{R}^{d}$ is initialised to $[\,I\;\;I\,]$ with zero bias. The BiMamba2 mixers and the bottleneck projections use the default Mamba2 initialisation, except that each depthwise convolution is set to the causal identity (kernel $[0,0,0,1]$, zero bias).

Student map. For each direction and head we form a non-negative SSD attention analogue from the captured B, C and the cumulative decay w: C is scaled by w and B divided by it, so that their unnormalised product carries the decay between positions (after the two softmaxes below this holds only approximately); Cw is normalised with a softmax over the state dimension and B/w with a softmax over the sequence, and the map is their product, restricted to its lower triangle. Each row of the untruncated product sums to one. The backward map is computed on the reversed sequence,

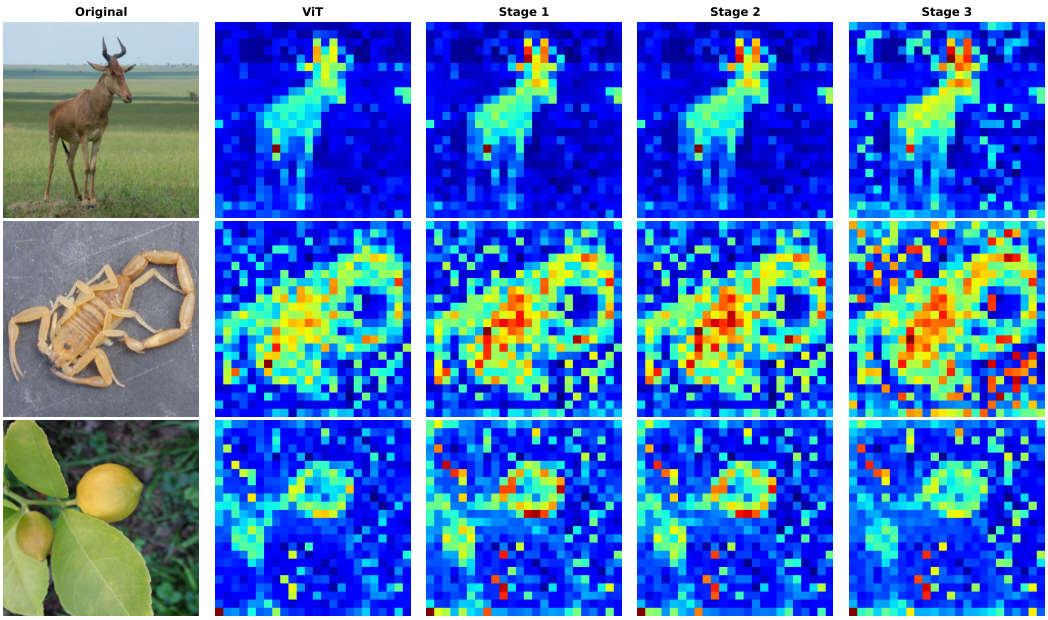


Figure 5: Diagonal-mean mixer-attention maps for the ViT-L teacher and the MambaFormer-L 10% student at the end of each distillation stage, on three ImageNet validation images. The teacher panel uses standard multi-head self-attention; the student panels use a bidirectional attention surrogate reconstructed from the Mamba2 SSD states (forward and backward scans combined). For each layer the diagonal of the (teacher or reconstructed) attention matrix is min–max normalised to $[0, 1]$ and averaged across all 24 layers, so the map highlights patches that are consistently self-salient throughout the forward pass. The student already reproduces the teacher's salient-patch structure after Stage 1 and tightens it at Stage 2; the structure persists through Stage 3 even though the loss has switched to soft-KL on the classifier logits.

flipped back, and added to the forward map; the diagonal, counted by both directions, is halved. All entries are non-negative, so the row-normalisation in Equation (3) is a proper distribution.

Teacher map and matching. The teacher's softmax attention (one CLS token, $n-1$ positions) is extended to the student's n positions by appending a copy of the CLS row and CLS column at the end of the sequence and halving both CLS columns, so that rows still sum to one. Maps are matched per head: MambaFormer uses the teacher's number of heads, and the Frobenius and KL terms are averaged over heads and images and summed over layers (Equations (4) and (5)). In our implementation the Stage-1/2 forward takes x, B, C after the identity-initialised convolution without its SiLU, which the standard Mamba2 forward used in Stage 3 and at inference applies; Stages 1–2 therefore align a slightly different function, and Stage 3 trains the deployed one end to end.

C Mixer attention maps across distillation stages

Method. For the ViT teacher we extract the standard multi-head self-attention weights at each of the 24 blocks. For the MambaFormer student we reconstruct an attention-like matrix from the Mamba2 SSD states by computing a normalised outer product of the input/output projections B and C scaled by the discrete decay w, separately for the forward and backward scans, and combining the two into a full bidirectional attention map. The diagonal-mean visualisation in Figure 5 takes the diagonal of each per-layer matrix (the patches' self-attention), min–max normalises it to $[0, 1]$ per layer, and averages across all 24 layers; this highlights patches that are consistently salient throughout the forward pass and suppresses layer-specific magnitude differences that would otherwise dominate the average.

Reading the figure, stage by stage. Stage 1 is trained purely against the teacher's per-block attention maps and the diagonal-mean panel shows that the student already reproduces the teacher's salient-patch structure across all three example images. Stage 2 swaps the primary loss for an MSE on the mixer output and keeps the attention-map term only as an explicit regulariser against the student matching that output through a mismatched attention pattern (Equation (5)); the figure shows the salient-patch structure established in Stage 1 is preserved while the mixer is re-trained. Stage 3 carries no per-block alignment term at all and is driven only by soft-KL on the classifier logits, yet the main salient-patch structure persists; the additional contrast that emerges (visible on the scorpion and lemon rows) highlights patches the teacher attends to with mid-strength signal, possibly reflecting broader teacher structure once the per-block constraint is lifted. The figure is qualitative and we make no quantitative similarity claim from it; attention-map alignment is used in Stages 1–2 as the mechanism by which the mixer is placed into the teacher's functional role, not as a separate contribution.

D HYPERPARAMETERS

Table 6 lists the per-stage hyperparameters of the classification pipeline; Table 7 does the same for the segmentation pipeline. Optimiser is AdamW (Loshchilov & Hutter, 2019) throughout. Learning-rate schedules are cosine with linear warmup; "freeze" columns indicate which student parameters are trained at that stage. The 10% subset is class-stratified (seed 42, 128,066 images) and is shared by every 10% run. Batch is the effective batch across the node; peak LR is the per-GPU base rate before world-size scaling. Pipelines prefixed "CLIP" use the CLIP ViT-B/16 teacher at 224^2; all others use the IN-21K ViT teachers at 384^2.

Table 6: Classification hyperparameters by stage. The *Trains* column uses the freeze pattern of Figure 4: "S1 subset" is the bottleneck `down_proj` and the Mamba2 `in_proj`; "+ decay" adds A and the Δ bias; "proj." is the bottleneck down/up-projections and the mixer's input and output projections; "full mixer" adds the depthwise convolution and the SSM parameters (A, Δ bias, D); "mixer + heads" is the full mixer plus the double-CLS tokens, the fusion projection, and the classifier head. Patch and positional embeddings, LayerNorms, and MLPs remain frozen throughout. Epochs in parentheses are the planned budget where a stage was stopped early at converged alignment targets.

Stage	Pipeline	Epochs	Batch	Peak LR	Aug	EMA	Trains
1 (Attn. Map Alignment)	B 100%	10	64	5×10^{-5}	off	off	S1 subset
1 (Attn. Map Alignment)	B 10%	20	128	1×10^{-4}	light	off	S1 + decay
1 (Attn. Map Alignment)	L 100%	5 (8)	64	5×10^{-5}	off	off	S1 subset
1 (Attn. Map Alignment)	L 10%	10	64	1×10^{-4}	light	off	S1 + decay
1 (Attn. Map Alignment)	CLIP B 100%	10	128	5×10^{-5}	off	off	S1 subset
1 (Attn. Map Alignment)	CLIP B 10%	20	128	1×10^{-4}	light	off	S1 + decay
2 (Attn. Output Alignment)	B 100%	15	128	5×10^{-5}	off	off	proj.
2 (Attn. Output Alignment)	B 10%	30	128	5×10^{-5}	light	off	full mixer
2 (Attn. Output Alignment)	L 100%	6 (12)	64	5×10^{-5}	off	off	proj.
2 (Attn. Output Alignment)	L 10%	20	64	5×10^{-5}	light	off	full mixer
2 (Attn. Output Alignment)	CLIP B 100%	15	128	5×10^{-5}	off	off	proj.
2 (Attn. Output Alignment)	CLIP B 10%	30	128	5×10^{-5}	light	off	full mixer
3 (Logit Distillation)	B 100%	50	128	1×10^{-5}	off	on	mixer + heads
3 (Logit Distillation)	B 10%	150	128	1×10^{-5}	on	on	mixer + heads
3 (Logit Distillation)	L 100%	40	64	5×10^{-6}	off	on	mixer + heads
3 (Logit Distillation)	L 10%	100	64	1×10^{-5}	on	on	mixer + heads
3 (Logit Distillation)	CLIP B 100%	50	128	1×10^{-5}	off	on	mixer + heads
3 (Logit Distillation)	CLIP B 10%	150	128	1×10^{-5}	on	on	mixer + heads

Common to all Stage 3 runs: weight decay 0.05, gradient clipping 1.0, warmup 2–3 epochs, EMA decay 0.9998 (0.9995 for L 10%), pure soft-KL loss (no cross-entropy term, no labels), and a cosine temperature schedule $T : 5 \to 1.5$ ($4 \to 1.5$ for L 10%). In Stage 3, augmentation is used on the 10% subsets only (Stages 1–2 on the 10% subsets use a light RandAugment, $n=2$, $m=3$ / $m=5$): RandAugment ($n=2$, $m=7$) (Cubuk et al., 2020), RandomErasing ($p=0.25$) (Zhong et al., 2020),

MixUp + CutMix ($\alpha = 0.8$ / $\alpha = 1.0$) (Zhang et al., 2018; Yun et al., 2019). Full-data runs disable it, since MixUp/CutMix perturb the soft-KL targets and full data needs no extra regularisation. CLIP runs use the CLIP normalisation statistics.

Early convergence of the L 100% alignment stages. With full data, both alignment stages of the Large pipeline were stopped once their targets plateaued: Stage 1 after 5 of 8 planned epochs, with mean per-block cosine similarity to the teacher's attention maps of 0.916 after epoch 2 and 0.922 at epoch 5; Stage 2 after 6 of 12 planned epochs, with mean per-block centred kernel alignment (CKA) to the teacher's attention outputs of 0.937 after epoch 2 and 0.946 at epoch 6. All compute figures use the realised epochs.

Table 7: Segmentation hyperparameters by stage. HA-seg and WM distillation use the same teacher in the loop; HR fine-tune is teacher-free.

Stage	Dataset	Iters	Batch	Crop	Peak LR
HA-seg	ADE20K	16,000	16	512	5×10^{-5}
WM	ADE20K	64,000	16	512	6×10^{-5}
HA-seg	Cityscapes	8,000	8	768	5×10^{-5}
WM	Cityscapes	32,000	8	768	6×10^{-5}
HR ft	Cityscapes	18,000	16	1024	1×10^{-5}

The per-stage losses (verified against our training code, to be released upon acceptance; CWD is channel-wise distillation (Shu et al., 2021)) are:

$$\begin{aligned}
\mathcal{L}_{\text{HA-seg}} &= \mathrm{CWD}(T{=}4) \text{ on tap features,}\\
\mathcal{L}_{\text{WM, ADE20K}} &= \mathrm{CE} + 0.4\,\mathrm{aux_CE} + 1.0\,\mathrm{KL}(T{=}2) + 3.0\,\mathrm{CWD}(T{=}4),\\
\mathcal{L}_{\text{WM, Cityscapes}} &= \mathrm{CE} + 0.4\,\mathrm{aux_CE} + 3.0\,\mathrm{KL}(T{=}2) + 1.5\,\mathrm{CWD}(T{=}4),\\
\mathcal{L}_{\text{HR ft}} &= \text{OHEM-CE} + 0.4\,\text{OHEM-aux_CE (no teacher).}
\end{aligned}$$

HA-seg is feature-only: just multi-tap CWD between the student and teacher backbone outputs, with no logit or auxiliary-head terms. WM adds the SETR convention of a $0.4\times$ auxiliary cross-entropy on each of four aux heads, plus the teacher's KL on logits and CWD on tap features. HR ft is teacher-free, applies OHEM (Shrivastava et al., 2016) to both the main and the auxiliary cross-entropy, and uses a layer-wise learning-rate decay (LLRD 0.9, smaller updates near the input layers). Weight decay 0.05 throughout; gradient clip 1.0.

E Training compute breakdown

Per-image distillation cost follows the Kaplan / Chinchilla convention $F_{\text{img}}^{\text{distill}} = 3F_{\text{stu}}^{\text{fwd}} + F_{\text{tea}}^{\text{fwd}}$ (student fwd+bwd $\approx 3\times$ fwd; teacher is frozen, fwd only). Per-image fwd values are full-model profiler totals at 384^2 (analytic at 224^2), including the classifier head, double-CLS fusion and final LayerNorm. Table 8 aggregates the four IN-21K-teacher classification pipelines. Totals are computed on the realised epochs and rounded to two decimals, at which precision they are insensitive to the exact 10% subset size (± 0.01 EFLOPs). The CLIP-teacher pipelines run at 224^2 and cost 14.18 EFLOPs (100%: 1.89 / 2.84 / 9.45 per stage) and 3.78 EFLOPs (10%: 0.38 / 0.57 / 2.84; stage values rounded individually, as for L 10% in Table 8); their teacher's WIT-400M pretraining is excluded, as for ViT-Linearizer (Appendix J).

Segmentation training compute uses backbone forward FLOPs at the training crop (Appendix J.4): $N_{\text{iter}} \times \text{batch} \times (3F_{\text{stu}} + F_{\text{tea}})$ for HA-seg and WM, $3F_{\text{tea}}$ per image for the teacher schedule ($160\text{k}\times16$ at 512 on ADE20K, $80\text{k}\times8$ at 768 on Cityscapes) and $3F_{\text{stu}}$ for the teacher-free HR fine-tune, with the schedules of Table 7. Per dataset: ADE20K HA-seg 0.71 EFLOPs + WM 2.84 EFLOPs $=$ 3.55 EFLOPs total (64% of the 5.56 EFLOPs SETR-PUP teacher schedule); Cityscapes HA-seg 0.42 EFLOPs + WM 1.67 EFLOPs $= 2.09$ EFLOPs total (57% of the 3.68 EFLOPs teacher schedule). The teacher-free Cityscapes HR fine-tune ($18{,}000$ iterations, batch 16, crop 1024) adds 2.36 EFLOPs, for a Cityscapes total of 4.45 EFLOPs ($1.21\times$ the teacher schedule).

Table 8: Classification training compute breakdown by pipeline and stage (EFLOPs). All numbers at 384^2.

Stage	B 100%	B 10%	L 100%	L 10%	Notes
Map Align (Stage 1)	5.65	1.13	9.88	1.98	S1 subset
Output Align (Stage 2)	8.47	1.69	11.86	3.95	mixer
Logit KD (Stage 3)	28.23	8.47	79.06	19.77	mixer + heads, EMA on
Total	**42.35**	**11.29**	**100.80**	**25.69**	
Teacher pre-training	136.19	136.19	477.09	477.09	cited reference
Ratio (ours / teacher)	0.31×	0.083×	0.21×	0.054×	3.2 / 12.1 / 4.7 / 18.6× cheaper

Wall-clock cost. All jobs ran on a single node with 4×NVIDIA H100 SXM5 GPUs. Table 9 reports approximate H100-GPU-hours per stage, rounded to the hour. The largest single line item is the L 100% Stage 3 logit-distillation run at ~ 221 H100-GPU-hours. Rows marked † sum per-epoch train times from the logs (validation excluded) because those stages ran across several jobs.

Table 9: Approximate per-stage wall-clock in H100-GPU-hours (4×H100 SXM5, one node), rounded to the hour; totals are sums of the rounded stages. †Summed per-epoch train time.

Pipeline	Stage	H100-GPU-hours
B 100%	Map Align (Stage 1)	52
	Output Align (Stage 2)	61
	Logit KD (Stage 3)	78
	Total	**191**
B 10%	Map Align (Stage 1)	12
	Output Align (Stage 2)	14
	Logit KD (Stage 3)	28
	Total	**54**
L 100%	Map Align (Stage 1)†	71
	Output Align (Stage 2)†	67
	Logit KD (Stage 3)†	221
	Total	**359**
L 10%	Map Align (Stage 1)	16
	Output Align (Stage 2)	24
	Logit KD (Stage 3)	56
	Total	**96**
CLIP B 100% (224^2)	Map Align (Stage 1)	13
	Output Align (Stage 2)	20
	Logit KD (Stage 3)†	63
	Total	**96**
CLIP B 10% (224^2)	Map Align (Stage 1)	4
	Output Align (Stage 2)	5
	Logit KD (Stage 3)	21
	Total	**30**
Freeze ablation (Appendix L)	Logit KD, skeleton unfrozen	31
From scratch (Appendix M)	300 epochs, 224^2†	161
ADE20K seg	HA-seg	4
	WM	24
	Total	**28**
Cityscapes seg	HA-seg	2
	WM	8
	Total	**10**
Cityscapes HR ft	HR	**9**

F FULL SEGMENTATION METRICS

Table 10 lists mIoU, mAcc and aAcc for every segmentation model in Table 3. The Cityscapes HR fine-tune starts from the Whole-Model distillation checkpoint, trains without the teacher at crop 1024 for 18,000 iterations (OHEM cross-entropy, learning rate 10^{-5}, effective batch 16), and is evaluated every 3,000 iterations with the 1024 sliding window. Its validation mIoU stays between 78.27 and 78.69 over the run. As for every model in this paper, we report the final checkpoint (78.50, -0.49 vs. the teacher) rather than the best periodic evaluation, since Cityscapes has no public test labels. The Cityscapes no-teacher baseline is likewise reported at its final checkpoint (40,000 iterations, 76.21); its best periodic evaluation was 76.74.

Table 10: Segmentation: all metrics (%). Sliding-window evaluation with the given crop, stride $2/3$ of the crop; "+ msflip" adds flips and scales $\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$. A dash marks msflip not run for that final checkpoint.

			slide			+ msflip		
Dataset	Model	Crop	mIoU	mAcc	aAcc	mIoU	mAcc	aAcc
ADE20K	ViT-L/16 (teacher)	512	47.91	59.45	82.51	50.14	60.88	83.56
	MambaFormer-L (SPD)	512	47.81	58.82	82.53	48.99	59.19	83.25
	MambaFormer-L (no teacher)	512	45.68	56.37	81.56	47.91	57.81	82.75
Cityscapes	ViT-L/16 (teacher)	1024	78.99	86.33	96.15	80.86	87.35	96.46
	MambaFormer-L (SPD + HR)	1024	78.50	85.43	96.26	—	—	—
	MambaFormer-L (SPD)	768	78.61	85.84	96.19	80.30	86.75	96.47
	MambaFormer-L (no teacher)	768	76.21	83.51	96.00	—	—	—

G COMPUTE–ACCURACY PARETO FOR SEGMENTATION

Figure 6 extends the classification compute–accuracy Pareto of Figure 1 to the segmentation setting. The MambaFormer-L student trained via the two-stage segmentation pipeline (HA-seg $\rightarrow$ Whole-Model Distillation; Section 3.3) is 0.10 mIoU below the SETR-PUP/ViT-L teacher on ADE20K at 64% of the teacher's training compute. On Cityscapes, distillation at crop 768 costs 57% of the teacher schedule, but the distilled student evaluated with the teacher's 1024 window is 1.17 mIoU below the teacher. The teacher-free HR fine-tune at crop 1024 reduces the gap to 0.49 mIoU and brings the total to $1.21\times$ the teacher's schedule, so on Cityscapes the student does not dominate the teacher in training compute. Neither budget includes the teacher's ImageNet-21K pretraining, which the student reuses. The teacher was itself trained on 768 crops (the SETR Cityscapes schedule) and reaches 78.99 with the 1024 window, whereas the distilled student drops from 78.61 at 768 to 77.82 at 1024 before the HR fine-tune. In this setting the student transfers less well than the ViT teacher to windows larger than its training crops, and the HR fine-tune is what closes that gap.

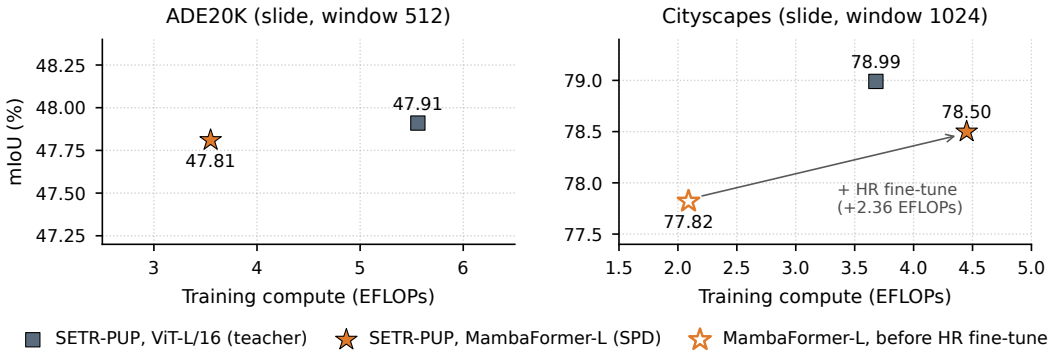


Figure 6: Segmentation training compute (backbone EFLOPs) vs. single-scale sliding-window mIoU, final checkpoints. On Cityscapes the arrow shows the teacher-free HR fine-tune that adapts the student to the teacher's 1024 window.

H EMA TRAINING-TIME BEHAVIOUR

Table 11 reproduces the early-, mid- and late-training EMA-vs-live deltas on the B 10% Stage 3 run (EMA decay 0.9998, 150-epoch cosine LR). The mid-training advantage of EMA over the live trajectory (in teacher agreement) peaks at 22% of the schedule and persists until the final $\sim 15\%$, then collapses to noise as cosine LR decay shrinks the step size.

Table 11: EMA-vs-live across the three training phases on B 10% Stage 3 (teacher-agreement on the val split).

Epoch	Live	EMA	Gap (EMA − live)
Phase 1, warm-up: EMA lags			
3	73.80	8.93	−64.87
6	78.60	62.54	−16.05
9	80.88	76.67	−4.21
Phase 2, steady state: EMA leads live			
33	84.83	85.98	+1.16 (peak)
Phase 3, convergence: gap closes			
150	88.706	88.690	−0.016

The same pattern holds across runs. The peak EMA gain in teacher agreement is $+1.16$ pp on B 10% Stage 3, $+1.08$ and $+1.16$ pp on two other B 10% Stage-3 runs of the recipe ablation (Appendix P); on L 10% it reaches $+3.49$ pp at epoch 6 and is $+1.86$ pp at epoch 30. Final-epoch deltas are below 0.06 pp in every case. In top-1, the final EMA and live weights differ by at most 0.03 pp (B 10%: 82.02 vs. 81.99; L 10%: 83.36 vs. 83.35; CLIP-B 100%: 84.26 vs. 84.25), and all reported accuracies use the live weights. At decay 0.9998 the EMA half-life is $\ln 2/\ln(1/\rho) \approx 3{,}465$ steps, a low-pass filter on the SGD trajectory with a few-thousand-step characteristic time. While LR is non-trivial and aug is heavy, that filter trims high-frequency noise on top of the slow improvement signal; once cosine decay drives LR toward $\eta_{\min}$, the live update magnitude shrinks toward zero and both trajectories converge.

I REPRODUCIBILITY

Code, configs, checkpoints. Training and evaluation code, all hyperparameter configs, the inference kernels of Appendix K, and the distilled MambaFormer-B / MambaFormer-L checkpoints will be released publicly upon acceptance.

Datasets. ImageNet-1K (Deng et al., 2009) (research-licence, accessed via the official torchvision interface); ADE20K (Zhou et al., 2017) (BSD-3 / non-commercial research); Cityscapes (Cordts et al., 2016) (custom non-commercial research licence; we comply with the redistribution clause and link to the official download).

Teachers. ViT-B/16 and ViT-L/16 IN-21K $\rightarrow$ IN-1K-384 checkpoints from the official `google-research/vision_transformer` release (Apache-2.0). CLIP ViT-B/16 fine-tuned on IN-1K: `vit_base_patch16_clip_224.openai_ft_in1k` from `timm` (Wightman, 2019; Cherti et al., 2023) (Apache-2.0), which evaluates at 85.26% top-1 at 224^2 in our pipeline. SETR-PUP-ViT-L/16 ADE20K and Cityscapes checkpoints from the official SETR release (Apache-2.0).

Eval pipeline. Classification: single-crop centred eval at the teacher's resolution (384^2, or 224^2 for CLIP) on the standard $50{,}000$-image IN-1K val split, top-1 reported as the percentage of images whose argmax matches the ground-truth label. Segmentation: sliding-window inference with stride $2/3$ of crop; "+ msflip" adds horizontal-flip and multi-scale TTA at scales $\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$. The eval scripts and the exact metric implementations will be part of the code release.

J FLOPs AND COMPUTE BUDGET

We report multiply–add FLOPs counting one multiply-add (MAC) as two FLOPs at batch size 1, and convert per-image inference cost to per-image training cost via the standard Kaplan/Chinchilla approximation $F_{\text{img}}^{\text{train}} \approx 3\, F_{\text{img}}^{\text{fwd}}$ (forward+backward $\approx 3\times$ forward). For a distillation step the frozen teacher contributes one fwd:

$$F_{\text{img}}^{\text{distill}} \;=\; 3\, F_{\text{stu}}^{\text{fwd}} \;+\; F_{\text{tea}}^{\text{fwd}}. \tag{7}$$

We use (7) for every distillation pipeline reported in this paper and, for context, for the ViT-Linearizer (Wei & Chellappa, 2025) pipelines we compare against.

J.1 PER-BLOCK FLOPs

Three building blocks cover every model in this analysis: the standard ViT block (used by ViT-B/16, ViT-L/16 and CLIP ViT-B/16, ViT-L/14); the BiMamba2 block (our MambaFormer student); and the single-direction Mamba2 block (Dao & Gu, 2024) (the Adventurer student (Wang et al., 2025a) of ViT-Linearizer). Let L be the token sequence length.

ViT block. With model dimension D and MLP width D_{mlp},

$$\begin{aligned} F_{\text{attn}}^{\text{ViT}} \;&=\; \underbrace{2L\,D\,(3D)}_{\text{QKV proj}} + \underbrace{2L^2 D}_{\text{QK}^\top} + \underbrace{2L^2 D}_{\text{AV}} + \underbrace{2L\,D^2}_{\text{out proj}} \;=\; 8L\,D^2 + 4L^2 D, \\ F_{\text{block}}^{\text{ViT}} \;&=\; F_{\text{attn}}^{\text{ViT}} + 4L\,D\,D_{\text{mlp}}. \end{aligned} \tag{8}$$

Single-direction Mamba2 block. Let E be the SSM expansion factor with $d_{\text{inner}} = Ed$, N the state dimension, K the causal-conv kernel width, G the number of B, C groups and H the number of SSM heads. Then

$$F_{\text{Mamba2}}(d) \;=\; 2L\,d\big(2d_{\text{inner}} + 2GN + H\big) + 2LK\big(d_{\text{inner}} + 2GN\big) + 6L\,d_{\text{inner}}\,N + 2L\,d_{\text{inner}}\,d. \tag{9}$$

The MambaFormer student wraps a bidirectional pair plus a fusion projection $\mathbb{R}^{2d_m} \to \mathbb{R}^{d_m}$:

$$F_{\text{BiMamba2}}(d_m) \;=\; 2\,F_{\text{Mamba2}}(d_m) \;+\; 2L\,(2d_m)\,d_m. \tag{10}$$

Adventurer uses a single direction at full width ($d = D$, $E = 2$).

Channel mixers. MambaFormer uses a 2-layer GELU MLP at hidden width $4D$, contributing $16L\,D^2$ per block. Adventurer uses a SwiGLU MLP with hidden width $D_h = 2.5D$ and three linears (gate, up, down), contributing $6L\,D\,D_h = 15L\,D^2$ per block.

Whole-model FLOPs. Let L_l be the number of encoder blocks, p the patch size, I a square input resolution (H, W for non-square segmentation inputs in Appendix J.5), n_{cls} the number of class/CLS tokens, and $L = (I/p)^2 + n_{\text{cls}}$. Then

$$\begin{aligned} F_{\text{patch}} \;&=\; 6\,D\,I^2, \\ F_{\text{model}}^{\text{ViT}} \;&\approx\; F_{\text{patch}} + L_l\,F_{\text{block}}^{\text{ViT}}, \\ F_{\text{model}}^{\text{MF}} \;&\approx\; F_{\text{patch}} + L_l\big(F_{\text{BiMamba2}}(d_m) + 4L\,D\,d_m + 16L\,D^2\big), \\ F_{\text{model}}^{\text{Adv}} \;&\approx\; F_{\text{patch}} + L_l\big(F_{\text{Mamba2}}(D)\big|_{E=2} + 15L\,D^2\big). \end{aligned} \tag{11}$$

The MambaFormer form additionally absorbs the $D \to d_m$ down/up bottleneck projections ($4L\,D\,d_m$ term). LayerNorms, GELU/SiLU, biases, the classifier head, and SSM auxiliary parameters together contribute $< 0.5\%$ and are absorbed. We will write $F_{\text{model}}^{\text{ViT}}$ as

$$F_{\text{model}}^{\text{ViT}} \;\approx\; 6\,D\,I^2 + L_l\big(8L\,D^2 + 4L^2 D + 4L\,D\,D_{\text{mlp}}\big), \quad L = (I/p)^2 + 1. \tag{12}$$

J.2 CONFIGURATIONS AND VALIDATION

Table 12: Architecture hyperparameters used in the FLOPs model.

Model	D	D_{mlp}/d_m	E	N	K	G	H	L_l / p	n_{cls}
ViT-B/16 (Dosovitskiy et al., 2021)	768	3072	—	—	—	—	—	12/16	1
ViT-L/16 (Dosovitskiy et al., 2021)	1024	4096	—	—	—	—	—	24/16	1
CLIP ViT-B/16 (Radford et al., 2021)	768	3072	—	—	—	—	—	12/16	1
CLIP ViT-L/14 (Radford et al., 2021)	1024	4096	—	—	—	—	—	24/14	1
MambaFormer-B	768	384	1	64	4	12	12	12/16	2
MambaFormer-L	1024	512	1	64	4	16	16	24/16	2
Adv-B/16 (Wang et al., 2025a)	768	1920	2	128	4	1	24	12/16	2
Adv-L/14 (Wang et al., 2025a)	1024	2560	2	128	4	1	32	24/14	2

CLIP ViT-B/16 is the standard ViT-B/16 block with one extra pre-transformer LayerNorm, an $\mathcal{O}(LD)$ term absorbed into its per-image count (caveat 6). For ViT we verified (12) against `fvcore.nn.FlopCountAnalysis` (after restoring the $QK^\top$ and AV terms that fvcore omits because the fused `nn.MultiheadAttention` kernel hides them) to within 0.1%. For Adventurer the official paper does not list the Mamba2 hyperparameters explicitly; plugging the Mamba2 (Dao & Gu, 2024) defaults ($E=2$, $N=128$, headdim$=64$, $G=1$, $K=4$) and $D_h=2.5D$ into the standard parameter accounting reproduces the counts in Table 7 of Wei & Chellappa (2025) to within 1% (99.8 M vs reported 99 M for Adv-B/16; 349.1 M vs reported 346 M for Adv-L/14).

J.3 Per-image inference FLOPs

Table 13: Per-image inference FLOPs at 224^2 and 384^2 (analytic count for ViT, Adventurer, and MambaFormer at 224^2; profiler count for MambaFormer at 384^2).

Model	$I=224$	$I=384$
ViT-B/16	35.13 G	110.97 G
MambaFormer-B (ours)	*37.49* G	**109.89 G**
Adv-B/16	**41.93 G**	*122.40* G
CLIP ViT-B/16	**35.13 G**	*110.97* G
ViT-L/16	123.11 G	382.13 G
MambaFormer-L (ours)	*132.08* G	**386.78 G**
CLIP ViT-L/14	162.02 G	—
Adv-L/14	**189.12 G**	*535.87*† G

Bold marks each model's evaluation resolution; *italic* entries are computed off-resolution and shown for reference only. MambaFormer-B is evaluated at 384^2 when distilled from ViT-B/16 and at 224^2 when distilled from CLIP ViT-B/16. †Adv-L/14 has $p=14$, which does not divide 384; we use $L=\lfloor 384/14 \rfloor^2+2=731$.

The MambaFormer student is parameter-matched to its same-size ViT teacher within $\sim 10\%$ by construction; FLOPs happen to coincide at 384^2 (Base: -1.0%; Large: $+1.2\%$) because the sequence length is short, so the teacher–student accuracy comparisons reported in this paper are at an effectively matched inference budget at this resolution. The linear-time advantage emerges as L grows (Appendix J.4). Adventurer, in contrast, is $+19\%/+17\%$ heavier than its same-class ViT at 224^2: the Mamba2 `in_proj` ($\approx 4.4D$ output) and the $6L\,d_{\text{inner}}\,N$ SSD term together exceed the linear part of self-attention at modest L. The $1.20\times/2.74\times/4.21\times$ throughput speed-ups reported in Tables 1–3 of Wei & Chellappa (2025) are wall-time benefits of removing the $\mathcal{O}(L^2)$ attention kernel, not FLOPs reductions.

J.4 Inference FLOPs across resolutions

Evaluated with (11)/(12) at $I \in \{384, 512, 768, 1024\}$. For MambaFormer, this analytic count is 0.3–0.4% below the per-model table above at 384^2 (109.43 vs 109.89 G; 385.58 vs 386.78 G), which is why the Large-tier gap is $+0.9\%$ here instead of $+1.2\%$. As I grows, ViT's $4L^2D$ attention term dominates while MambaFormer remains linear in L.

Table 14: Backbone inference GFLOPs across input resolutions (analytic count, no decoder). The 1024×2048 column is one full Cityscapes image, the setting of Figure 3a.

Base ($L_l = 12$)	$I = 384$	$I = 512$	$I = 768$	$I = 1024$	1024×2048
ViT-B/16	110.97	214.05	590.13	1319.56	3875.91
MambaFormer-B (ours)	109.43	194.25	436.60	775.89	1551.40
Δ (ViT extra)	$+1.41\%$	$+10.19\%$	$+35.16\%$	$+70.07\%$	$+149.83\%$
Large ($L_l = 24$)	$I = 384$	$I = 512$	$I = 768$	$I = 1024$	1024×2048
ViT-L/16	382.13	723.97	1918.09	4131.02	11559.97
MambaFormer-L (ours)	385.58	684.43	1538.31	2733.74	5466.14
Δ (ViT extra)	-0.89%	$+5.78\%$	$+24.69\%$	$+51.11\%$	$+111.48\%$

The $\mathcal{O}(L)$ vs. $\mathcal{O}(L^2)$ gap widens monotonically: at 1024^2, ViT-L is $1.51\times$ MambaFormer-L on classification, and the gap reaches $2.11\times$ on one full Cityscapes image ($2.50\times$ for Base), or $1.96\times$ with the SETR-PUP head (Appendix J.5). For comparison, Adv-B/16 falls below ViT-B/16 in raw FLOPs from $I \approx 544$ and reaches $\approx 0.5\times$ only at $I \gtrsim 1280$ (where $L \approx 6402$); the MambaFormer's bottlenecked BiMamba2 saves FLOPs at much shorter sequences.

J.5 Segmentation inference FLOPs (SETR-PUP)

We swap only the encoder, keeping the SETR-PUP head identical between teacher (ViT-L/16) and student (MambaFormer-L). The backbone reuses (11) with $I^2 \to HW$ and $L = (H/p)(W/p) + n_{\text{cls}}$ ($n_{\text{cls}} = 1$ teacher, 2 student). The PUP head consumes the last backbone tap, applies four 3×3 Conv+SyncBN+ReLU+ bilinear2$\times$ stages, and a 1×1 classifier with channel width $C = 256$, K output classes, $hw = HW/p^2$:

$$F_{\text{PUP}} \;=\; 18\,D\,C\,hw \;+\; 1512\,C^2\,hw \;+\; 2\,C\,K\,HW \;\approx\; \big(4.055 \times 10^5 + 512\,K\big)\,HW. \qquad (13)$$

The four auxiliary heads are training-only. The full segmentation cost is $F_{\text{seg}} = F_{\text{backbone}} + F_{\text{PUP}}$.

Table 15: SETR-PUP segmentation FLOPs per forward pass, backbone and decoder.

	ADE20K ($K = 150$)	Cityscapes ($K = 19$)		
	512×512	768×768	1024×1024	1024×2048
SETR-PUP-ViT-L (teacher)	**850.40**	**2163.00**	**4566.42**	**12430.78**
SETR-PUP-MambaFormer (ours)	**810.87**	**1783.22**	**3169.14**	**6336.94**
Δ (teacher extra)	$+4.88\%$	$+21.30\%$	$+44.09\%$	$+96.16\%$

For reference, ViT-Linearizer reports *encoder-only* segmentation throughput (decoder excluded; UperNet (Xiao et al., 2018), not SETR-PUP) for Adv-B/16. At ADE20K 512^2 the encoder is FLOPs-tied with ViT-B/16 (217.28 vs 214.05 GFLOPs); at Cityscapes 512×1024 Adv-B/16 saves 14% encoder FLOPs while the reported wall-time speedup is $4.21\times$. The crossover where Adv encoder FLOPs alone beat the ViT teacher by $\geq 2\times$ falls at substantially longer sequences than for MambaFormer.

Per-image cost under MS+flip evaluation (Cityscapes). mIoUs are produced under sliding-window inference with multi-scale + flip TTA (SETR/mmseg convention). For $S = \{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$ and flip on, the total per image is $N_{\text{tot}} = 2\sum_{s \in S} n_{\text{win}}(s)$, where $n_{\text{win}}(s) = n_h(s)\, n_w(s)$ with $n_h(s) = \max(\lceil (\max(sH - c, 0))/r \rceil + 1,\ 1)$ and likewise for n_w; each window is a full $c \times c$ forward (smaller crops are padded). Cityscapes val is 1024×2048:

Table 16: Per-image Cityscapes inference FLOPs under sliding-window evaluation.

Eval (crop c, stride r)	N_{tot}	Teacher F_{image}	Student F_{image}
Native ($c = 768$, $r = 512$)	124	268.21 T	221.12 T
1024^2 HR ($c = 1024$, $r = 683$)	74	337.92 T	234.52 T
Full image, no slide, no MS+flip	1	12.43 T	6.34 T

A single MS+flip val image costs 221–268 TFLOPs at native and 235–338 TFLOPs at 1024^2; the MambaFormer student is $1.21\times$/$1.44\times$ cheaper than the SETR-PUP-ViT-L teacher under the two settings, and $1.96\times$ cheaper under a single full-image forward.

J.6 PER-IMAGE DISTILLATION COST

Plugging the per-image fwd FLOPs of Appendix J.3 into (7):

Table 17: Per-image distillation cost, $3F_{\text{stu}}^{\text{fwd}} + F_{\text{tea}}^{\text{fwd}}$.

Pipeline	Teacher	Resolution	$F_{\text{stu}}^{\text{fwd}} / F_{\text{tea}}^{\text{fwd}}$	$F_{\text{img}}^{\text{distill}}$
MambaFormer-B (ours)	ViT-B/16	384^2	109.89 / 111.08 G	440.74 G/img
MambaFormer-L (ours)	ViT-L/16	384^2	386.78 / 382.42 G	1,542.76 G/img
MambaFormer-B (ours)	CLIP ViT-B/16	224^2	37.49 / 35.13 G	147.58 G/img
Adv-B/16 (Wei & Chellappa, 2025)	CLIP ViT-B/16	224^2	41.93 / 35.13 G	160.91 G/img
Adv-L/14 (Wei & Chellappa, 2025)	CLIP ViT-L/14	224^2	189.12 / 162.02 G	729.38 G/img

For the IN-21K teachers, the teacher forward cost in this table is the profiler count (111.08 / 382.42 G), as for the students; the analytic values of Appendix J.3 are 0.1% lower. The 10%-data totals use a subset count of 128,116 images, 0.04% above the 128,066 images of the final subset; no printed ratio changes.

The Adventurer pipeline additionally includes a teacher-free fine-tuning stage at 224^2 with $F_{\text{img}}^{\text{ft}} = 3\,F_{\text{stu}}^{\text{fwd}}$ (125.79 G/img Base, 567.36 G/img Large).

J.7 TEACHER PRETRAINING COMPUTE (REFERENCE)

The MambaFormer teachers are the `google-research/vision_transformer` ViT-B/16 and ViT-L/16 checkpoints: ImageNet-21K pretraining at 224^2 for 90 epochs of 14 M images (1.26×10^9 images seen) and ImageNet-1K fine-tuning at 384^2 for 20,000 steps at batch 512 (1.024×10^7 images, ≈ 7.99 epochs of IN-1K). Applying the $3\times$ Kaplan factor:

Table 18: Teacher pretraining compute (reference).

Stage	Images seen	ViT-B/16	ViT-L/16
IN-21K pretrain (224^2, 90 ep)	1.26×10^9	132.78 EFLOPs	465.35 EFLOPs
IN-1K fine-tune (384^2, 20k×512)	1.024×10^7	3.41 EFLOPs	11.74 EFLOPs
Teacher total		**136.19 EFLOPs**	**477.09 EFLOPs**

The CLIP ViT-B/16 and ViT-L/14 teachers used by ViT-Linearizer were pretrained on WIT-400M for ≈ 12.8 B image–text pairs at 224^2. A naïve estimate ($12.8\times 10^9 \times 3\,F_{\text{tea}}^{\text{fwd}}$) gives ≈ 1.35 ZFLOPs and ≈ 6.22 ZFLOPs respectively, one to two orders of magnitude above the distillation budgets below (17–20$\times$ ViT-Linearizer's, $\sim$95–360$\times$ our CLIP-distilled students'); ViT-Linearizer does not account for teacher pretraining, and we likewise exclude it, for ViT-Linearizer and for our CLIP-distilled students alike.

J.8 PER-PIPELINE DISTILLATION TOTALS

MambaFormer (ours). Three-stage distillation (Map Align / Output Align / Logit Distill) at 384^2 (IN-21K ViT teachers) or 224^2 (CLIP ViT-B/16 teacher) on IN-1K full or class-stratified 10% subset (1,281,167 / 128,066 images). Epochs are realised epochs.

Table 19: Distillation compute per pipeline and stage.

Pipeline	Stage	Epochs	Images	Compute
B 100%	Map Align	10	12.81 M	5.65 EFLOPs
	Output Align	15	19.22 M	8.47 EFLOPs
	Logit Distill	50	64.06 M	28.23 EFLOPs
	Total	**75**	**96.09 M**	**42.35 EFLOPs**
B 10%	Map Align	20	2.56 M	1.13 EFLOPs
	Output Align	30	3.84 M	1.69 EFLOPs
	Logit Distill	150	19.21 M	8.47 EFLOPs
	Total	**200**	**25.61 M**	**11.29 EFLOPs**
L 100%	Map Align	5	6.41 M	9.88 EFLOPs
	Output Align	6	7.69 M	11.86 EFLOPs
	Logit Distill	40	51.25 M	79.06 EFLOPs
	Total	**51**	**65.34 M**	**100.80 EFLOPs**
L 10%	Map Align	10	1.28 M	1.98 EFLOPs
	Output Align	20	2.56 M	3.95 EFLOPs
	Logit Distill	100	12.81 M	19.77 EFLOPs
	Total	**130**	**16.65 M**	**25.69 EFLOPs**
CLIP B 100%	Map Align	10	12.81 M	1.89 EFLOPs
	Output Align	15	19.22 M	2.84 EFLOPs
	Logit Distill	50	64.06 M	9.45 EFLOPs
	Total	**75**	**96.09 M**	**14.18 EFLOPs**
CLIP B 10%	Map Align	20	2.56 M	0.38 EFLOPs
	Output Align	30	3.84 M	0.57 EFLOPs
	Logit Distill	150	19.21 M	2.84 EFLOPs
	Total	**200**	**25.61 M**	**3.78 EFLOPs**

ViT-Linearizer (Adventurer student) (Wei & Chellappa, 2025). 300 epochs of distillation on IN-1K at 224^2 (batch 2048) followed by 100 (B) or 50 (L) epochs of teacher-free fine-tuning at 224^2 (batch 1024), both stages on the full IN-1K.

Table 20: ViT-Linearizer (Adventurer student) training compute.

Pipeline	Distillation	Fine-tune	Total
Adv-B/16 ← CLIP ViT-B/16	61.85 EFLOPs (300 ep)	16.12 EFLOPs (100 ep)	**77.96 EFLOPs**
Adv-L/14 ← CLIP ViT-L/14	280.34 EFLOPs (300 ep)	36.34 EFLOPs (50 ep)	**316.68 EFLOPs**

J.9 END-TO-END COMPARISON

All compute values below are in EFLOPs.

Table 21: End-to-end training compute of our students against their teachers and ViT-Linearizer (EFLOPs).

Pipeline	Distill	Same-size teacher	vs. teacher	vs. same-tier Adv
MambaFormer-B (B 100%)	42.35	136.19 (ViT-B)	3.22× less	1.84× less
MambaFormer-B (B 10%)	11.29	136.19 (ViT-B)	12.06× less	6.91× less
MambaFormer-L (L 100%)	100.80	477.09 (ViT-L)	4.73× less	3.14× less
MambaFormer-L (L 10%)	25.69	477.09 (ViT-L)	18.57× less	12.33× less
MambaFormer-B (CLIP B 100%)	14.18	— (CLIP excluded)	—	5.50× less
MambaFormer-B (CLIP B 10%)	3.78	— (CLIP excluded)	—	20.62× less
Adv-B/16 (Wei & Chellappa, 2025)	77.96	— (CLIP excluded)	—	—
Adv-L/14 (Wei & Chellappa, 2025)	316.68	— (CLIP excluded)	—	—

Read-out. MambaFormer-L (L 10%) trains for 25.69 EFLOPs, 5.4% of its own teacher's lifetime compute and 12.3× less than Adventurer-L's distillation pipeline (which itself excludes the ZFLOP-scale CLIP pretraining of its teacher); with full data (L 100%) it trains for 100.80 EFLOPs, 4.7× and 3.1× less respectively. At the Base tier, the controlled comparison is the CLIP-distilled MambaFormer-B, which shares the teacher family, data, and 224^2 resolution of Adv-B/16: 14.18 EFLOPs, 5.5× less

than Adv-B/16's 77.96 (3.78 EFLOPs and 20.6× less from the 10% subset). The IN-21K-teacher B 100% and B 10% pipelines are 1.84× and 6.9× cheaper than Adv-B/16 but differ from it in teacher and resolution. Two factors contribute: the skeleton-preserving design (Section 1) keeps the inherited patch and positional embeddings, LayerNorms, and MLPs frozen, so training compute is spent only on aligning the new mixer; and the three-stage curriculum converges in 51–200 epochs on 10–100% of IN-1K, versus 300 ep distillation + 50–100 ep teacher-free fine-tuning on full IN-1K for ViT-Linearizer.

J.10 Caveats

1. **Kaplan 3×.** Eq. (7) assumes fwd+bwd ≈ 3× fwd for the student. Exact accounting of the frozen Stage-3 MambaFormer-B gives 2.37× fwd (2.97× with the skeleton unfrozen; Appendix L), because frozen weights skip their weight gradients. The convention therefore overestimates the training cost of our frozen students, and every budget here is conservative for them. In Stages 1–2 only the per-block mixer path runs, so charging the full-model $3F_{\text{stu}}^{\text{fwd}}$ further overestimates their cost; this more than covers the $\mathcal{O}(n^2)$ cost of forming the Stage-1/2 attention maps, which is not otherwise counted. The SSD term of (9) is the recurrent-form count; chunked SSD kernels execute additional intra-chunk work (a few percent of MambaFormer-B's FLOPs at 384^2, growing linearly with L), which we do not count.
2. **Live teacher.** We assume the teacher runs every step. Caching teacher logits/features over the longer stages would zero the teacher term (saving 16–19% of total compute if cached in Stage 3 only, 24–25% if cached in every stage).
3. **Pretraining accounting.** IN-21K is reported at 14 M images; deduplication corrects this by $<2\%$, neglected. CLIP teacher pretraining is excluded for Adv and for our CLIP-distilled students alike (a naïve estimate would dominate every budget tabulated above).
4. **Activation-matching loss.** The pairwise cosine-similarity term in ViT-Linearizer's Eq. 4 contributes $\approx 4{\times}2L^2D$ FLOPs/img, i.e. $\approx 0.15\%$ of the per-image distillation cost; absorbed.
5. **Adv segmentation FLOPs.** The Adv-B/16 segmentation numbers in Appendix J.5 are encoder-only, since Wei & Chellappa (2025) reports throughput "based solely on encoding time" and uses an UperNet decoder rather than SETR-PUP.
6. **CLIP `ln_pre`.** The CLIP-distilled MambaFormer-B inherits the teacher's extra pre-transformer LayerNorm. It is an $\mathcal{O}(LD)$ term ($<0.002\%$ of the per-image forward) present in both student and teacher, and is absorbed into the 224^2 per-image counts.

K Efficient inference kernels and wall-clock throughput

FLOPs (Figure 3a) do not account for memory traffic or kernel-launch overhead. We therefore measure wall-clock inference of MambaFormer-B against ViT-B/16 on an H100. The original MambaFormer code runs each BiMamba2 mixer as two independent upstream Mamba2 modules (Dao & Gu, 2024), which costs about 1,260 kernel launches per forward pass (ViT-B: about 530) and writes a 256×256 fp32 $CB^\top$ matrix per head and chunk to GPU memory. We re-implemented the inference path without changing the model, its weights or the checkpoint format; the new path issues about 530 launches per forward pass.

Exact algebraic fusion. The input projections of the two scan directions are concatenated into one GEMM. The output side (the two Mamba2 output projections, the BiMamba2 projection that mixes the two directions, and the bottleneck up-projection) is a chain of linear maps and is composed once, in fp32, into a single GEMM. The fused weights are non-persistent buffers derived from the live parameters.

Depthwise convolution. Each Mamba2 module keeps its width-4 depthwise causal convolution, initialised to the identity and trained from Stage 2 or Stage 3 on, so it is not the identity in any distilled checkpoint. At inference, a single streaming Triton kernel applies the convolution and its SiLU to x, B and C of both directions (causal for the forward scan, anti-causal for the backward one) and writes the activated values, which the SSD kernel reads directly. Each element is read and written once.

Fused bidirectional SSD kernel. Within a chunk, the SSD output is $y_m = \sum_{k \le m} (C_m^\top B_k)\, e^{I_m - I_k}\, \Delta_k x_k + D x_m$, gated by $\mathrm{SiLU}(z_m)$, where I is the cumulative log-decay. This is a causal attention with a decay mask, so we compute it with one Triton kernel in the style of FlashAttention (Dao et al., 2022): $C^\top B$, the decay mask and the product with x are formed tile by tile on chip and never written to HBM. Both directions run in the same launch; the backward direction reads the sequence in reverse order, so no flip or copy is needed. The D skip and the z gate are applied inside the kernel. On tiles strictly below the diagonal the decay is factorised as $e^{I_m - I_e}\, e^{I_e - I_k}$, with I_e the decay at the tile boundary. Both factors are at most one, so the product cannot overflow and needs no mask. Decays are stored in base 2 so that each one costs a single `exp2`. Beyond 256 tokens, the same design is applied chunk-wise (64-token chunks) with a per-chunk state kernel and a sequential state-passing scan; these kernels use 64-bit offsets. Precision follows upstream: bf16 tensor-core products, fp32 states and decays.

Kernel configurations. Tile sizes, warps and pipeline stages are fixed for the H100 from sweeps timed under CUDA graphs; 4-warp, 64-row tiles halve the scan time at every shape from batch 8 upward. Configurations are static, so CUDA-graph capture is safe.

Correctness. The fused SSD kernels are tested against the upstream `mamba_chunk_scan_combined` for sequence lengths from 50 to 5,627, head dimensions 32/64/128, state dimensions 32/64/128, grouped heads, and bf16 and fp32 inputs. The convolution kernel is tested against `F.conv1d` (causal and anti-causal, lengths 1 to 1,026), and the SSD kernel fed by it against the SSD applied after a reference convolution. Whole-model outputs of the new path match the original within bf16 rounding. On the full ImageNet-1K validation set, the distilled checkpoints give the same top-1 on both paths (EMA weights; CLIP-B: 84.28 vs. 84.26; L 10%: 83.35 vs. 83.36), with 99.8–99.9% of predictions identical; all accuracies in this paper use the original path.

Protocol. Models: torchvision `vit_b_16` and MambaFormer-B (12 layers, width 768, BiMamba2 mixer of width 384 with 12 heads of dimension 32 and state dimension 64). MambaFormer-B is timed with the weights of the distilled CLIP-B checkpoint (trained convolution); timing does not otherwise depend on the weight values. All three models (ViT-B, original MambaFormer-B, new MambaFormer-B) use the same deployment setting: bf16 weights and the whole forward pass captured in a CUDA graph. Latency is the median over 10 to 200 iterations (about 3 s per cell) after warm-up, timed with CUDA events on an exclusive H100 80GB (PyTorch 2.11, Triton 3.6). Token counts are 198, 486, 2,502 and 5,627 at 224^2, 352^2, 800^2 and 1200^2. By default ViT-B dispatches to FlashAttention-2 through `scaled_dot_product_attention`. cuDNN attention is faster at long sequences, and Table 22 uses it as the stronger baseline for the claims in Figure 3b.

Table 22: H100, bf16 weights, CUDA graph: ViT-B/16 with cuDNN attention against MambaFormer-B. Throughput in images/s; the last column is the throughput ratio of our MambaFormer-B to ViT-B (cuDNN), computed from latencies. The ViT-B column was measured in a separate session with the same protocol.

Res., batch	ViT-B (cuDNN)	MF-B orig.	MF-B ours	vs. ViT-B
224^2, 32	5,766	3,303	5,762	1.00×
800^2, 8	374	294	465	1.24×
1200^2, 8	138	136	213	1.54×
1200^2, 32	138	138	215	1.55×

Reading the numbers. Our implementation is 1.5–1.7× faster than the original code at every resolution and batch size. Against ViT-B, MambaFormer-B is at least 0.97× as fast at 224^2 and 352^2 from batch 8 upward (0.97–1.06×), and its lead grows with resolution because the scan is linear in sequence length: at batch 32, 1.01× at 198 tokens, 1.06× at 486, 1.35× at 2,502 and 1.78× at 5,627. Batch 1 at 224^2 and 352^2 is latency-bound: there MambaFormer-B runs at 0.82–0.83× ViT-B throughput, because of the additional bottleneck GEMM and convolution pass in each block and a scan kernel with a longer dependent chain than attention. Without CUDA graphs (eager PyTorch), the original code is bound by launch overhead at small batch (21–24 ms per forward pass at every

Table 23: Full inference grid on H100, bf16 weights, CUDA graph. Latency in ms per batch. ViT-B uses its default FlashAttention-2 path. "vs. orig." is the speed-up of our implementation over the original MambaFormer code; "vs. ViT" is the throughput ratio of our MambaFormer-B to ViT-B (above 1: MambaFormer-B is faster).

Res.	Batch	ViT-B	MF orig.	MF ours	vs. orig.	vs. ViT
224^2	1	1.02	1.99	1.24	1.60	0.82
	8	2.00	3.46	2.07	1.67	0.97
	32	5.60	9.69	5.55	1.74	1.01
352^2	1	1.15	2.26	1.39	1.63	0.83
	8	3.82	6.19	3.77	1.64	1.01
	32	14.31	21.60	13.46	1.60	1.06
800^2	1	2.87	4.57	2.79	1.64	1.03
	8	23.27	27.23	17.22	1.58	1.35
	32	90.11	104.39	66.68	1.57	1.35
1200^2	1	7.55	8.35	5.43	1.54	1.39
	8	66.56	58.83	37.47	1.57	1.78
	32	265.09	231.90	148.94	1.56	1.78

resolution), and our implementation is 3.3–$3.7\times$ faster there; once the GPU is saturated, the eager speed-ups are close to the graph-mode ones.

L FREEZING THE INHERITED SKELETON: FULL ABLATION

Where the freeze binds. Stages 1 and 2 are teacher-forced per block: the first block receives the teacher's patch and positional embeddings and CLS token (detached), block ℓ receives the teacher's block-$(\ell-1)$ output, and the loss stops at the mixer output. Of the 57.7M skeleton parameters of MambaFormer-B, only the pre-mixer LayerNorm affines ($18{,}432$ parameters, directly in front of the trainable bottleneck down-projection) lie in the Stage-1/2 computation graph; patch and positional embeddings, the post-mixer LayerNorms, the MLPs, and the final LayerNorm do not. Unfreezing upstream of Stage 3 is therefore a near no-op by construction, and the ablation re-runs Stage 3 only, warm-started from the exact Stage-2 checkpoint of the frozen B 10% run.

Setup. One configuration flag changes: patch embedding, positional embedding, all LayerNorm affines, and all MLPs join the single AdamW group at the stage learning rate (no separate learning rate, no layer-wise decay, no extra warmup). Everything else is identical to the frozen control: 150 epochs, effective batch 128, peak LR 10^{-5}, weight decay 0.05, gradient clip 1.0, $T: 5 \to 1.5$, pure soft-KL, the 10% augmentation stack, EMA decay 0.9998 (the newly trainable parameters are inside the EMA average), seed, and data subset. Trainable parameters per stage are $35{,}584{,}128$ / $35{,}814{,}528$ / $37{,}765{,}480$ for the control and $95{,}507{,}560$ in the unfrozen Stage 3 (100% of the model).

Table 24: Freeze ablation, top-1 / top-5 on the full IN-1K val at 384^2 (epoch 150). Both checkpoint kinds are evaluated for both runs with the same evaluator; re-evaluating the control reproduced its original numbers exactly.

Stage-3 skeleton	Top-1 (live)	Top-5 (live)	Top-1 (EMA)	Top-5 (EMA)
Frozen (control)	81.99	96.36	82.02	96.35
Unfrozen	82.31	96.61	82.34	96.61
Δ	$+0.32$	$+0.25$	$+0.33$	$+0.26$

Cost. Unfreezing changes neither the forward pass nor the input-gradient term: the frozen control already back-propagates through every MLP to reach the trainable mixers of earlier blocks. The entire difference is the weight-gradient term of the newly trainable tensors (Table 25). Over Stage 3 (150 epochs $\times$ $128{,}066$ images) student compute grows from $4{,}997$ to $6{,}269$ PFLOPs ($+1{,}271$ PFLOPs), the cost of 38 additional epochs of the frozen recipe. Measured train time per epoch rises from 146.6 s

to 166.2 s (median, +13.4%; 6.14 h to 6.96 h over 150 epochs), less than the FLOPs ratio because the frozen teacher's forward pass and the data pipeline are identical in both runs. Peak memory rises from 8.84 to 11.35 GiB (1.28×), with no gradient accumulation needed. At pipeline level, where Stages 1–2 are unchanged, wall-clock grows by about 6%.

Table 25: Per-image student training FLOPs at 384^2 (GFLOPs), decomposed exactly. The frozen student costs 2.37× its forward pass rather than the 3× assumed by the Kaplan convention used for every budget in this paper (Appendix J), so those budgets overestimate the frozen students' training cost.

Term	Frozen	Unfrozen	Ratio
Forward	109.89	109.89	1.00
Backward, input gradients	109.21	109.21	1.00
Backward, weight gradients	41.04	107.22	2.61
Student total	**260.14**	**326.32**	**1.25**

Of the +66.18 GFLOPs per image, the MLP weight gradients account for 65.46 (98.9%), the patch embedding for 0.68, and all LayerNorm affines for 0.04.

Overfitting and skeleton drift. If freezing helped by regularising, the unfrozen model (95.5M trainable parameters on 128,066 images) should overfit. It does not: on a fixed 10,000-image slice of the training subset under the evaluation transform, the train–val gap is 7.70 pp at epoch 15, 7.92 at 45, 7.99 at 75, 8.07 at 105, 8.11 at 135, and 8.24 at 150. Meanwhile the skeleton moves substantially from its inherited values: relative drift $\|W - W_0\|/\|W_0\|$ at epoch 150 is 0.135 (patch embedding), 0.158 (positional embedding), 0.141 (pre-mixer LayerNorms), 0.122 (post-mixer LayerNorms), 0.180 (MLPs), and 0.153 (final LayerNorm), with 97% of the final MLP drift reached by epoch 105. The best checkpoints of the unfrozen run (epoch 135 live, 82.37; epoch 147 EMA, 82.37) are within 0.06 pp of the final ones.

Reading. Unfrozen, the skeleton moved by 12–18% without overfitting and gained 0.32 pp, which suggests that the inherited weights already suit the student and that freezing them costs little and does not act as a regulariser. Two caveats bound this reading: the run is a single seed, so a 0.32 pp difference is not established above seed variance, and the unfrozen arm uses the control's hyperparameters; a tuned arm (e.g., with layer-wise learning-rate decay) could gain more. We keep the frozen design and report the trade-off.

M SUPERVISED FROM-SCRATCH REFERENCE

To measure the gain from inheritance plus distillation at full data, we trained the same MambaFormer-B (double CLS, 95.2M parameters) from random initialisation with hard labels and no teacher on the full IN-1K at 224^2, following the Vim/DeiT supervised recipe (Zhu et al., 2024; Touvron et al., 2021): 300 epochs, effective batch 512, AdamW with weight decay 0.05, 5 warmup epochs, label smoothing 0.1, stochastic depth 0.1, RandAugment, RandomErasing ($p = 0.25$), MixUp/CutMix (0.8/1.0), and EMA decay 0.99996, in bf16. Two deviations from that recipe were needed. The timm RandAugment magnitude of DeiT made the model collapse at epoch 2, so a milder torchvision RandAugment was used. Training then diverged at epochs 93 and 123–124. After the first divergence gradient clipping was tightened from 1.0 to 0.5 and optimiser steps with an outlying global gradient norm were skipped; after the second, the run resumed from epoch 120 with the peak learning rate scaled from 10^{-3} to 6×10^{-4} and the skip threshold set to a gradient norm of 3.0. That guard skipped 31% of the optimiser steps of epochs 121–300, and most steps of the final epochs.

The final-epoch top-5 is 88.39 (66.95 top-1 with EMA weights); the best epoch reached 67.91 (epoch 294), which we do not report as the result because it is selected on the validation set. Because of the instabilities and the skipped steps, 67.75% is a lower bound on what this architecture reaches from scratch with a tuned recipe, and the 16.5 pp gap to distillation should not be read as a tight estimate. It does show that the distilled student is well above what the same architecture obtained from this standard supervised recipe at 3.0× its compute.

Table 26: Supervised from-scratch training vs. three-stage distillation of the same architecture. Accuracies are final-epoch top-1 on the full IN-1K val. The from-scratch compute counts every step, including skipped ones (3×37.49 GFLOPs per image, 1,281,167 images, 300 epochs).

Training	Data, resolution	Epochs	Top-1	Train (EFLOPs)	H100-GPU-h
From scratch, hard labels	100%, 224^2	300	67.75	43.22	161
SPD, CLIP ViT-B/16 teacher	100%, 224^2	75	**84.25**	**14.18**	**96**
From scratch, CE	10%, 384^2	200	48.47	—	—
SPD, ViT-B/16 teacher	10%, 384^2	200	**81.99**	**11.29**	**54**

N Relation to MOHAWK: component-level attribution

MOHAWK (Bick et al., 2024) distils a causal Transformer language model into a Mamba2 student in three stages: matrix orientation (aligning the sequence-mixing matrices), hidden-state alignment (aligning each mixer block's output, block by block), and weight transfer with knowledge distillation (the teacher's embeddings, MLPs, norms, and head are copied into the student, which is then trained end-to-end on the teacher's logits). MOHAWK reports that the final stage can keep everything but the mixer frozen at a small cost (its Table 8), but trains the full student by default. Table 27 attributes each component of SPD.

Table 27: Component-level attribution of SPD relative to MOHAWK. "Inherited" components come from MOHAWK; "adaptation" components are required by the vision setting; "new" components are not part of MOHAWK.

Component	Relation	Evidence
Per-block alignment stages with teacher-provided inputs (Stages 1–2)	Inherited; targets adapted to non-causal attention and two CLS positions	Table 4, Section 3.2
Weight transfer of the non-mixer components	Inherited	Section 3.1
Logit distillation as the final stage	Inherited (standard KD)	Table 4
Frozen skeleton as the default at every stage	Made default; cost measured	Table 5: $+0.32$ top-1 if unfrozen, at $+25.4\%$ FLOPs
Parameter-matched, bottlenecked bidirectional Mamba2 mixer	Vision adaptation	Section 3.1
Double-CLS initialisation reproducing the teacher's CLS path at $t{=}0$	New	Section 3.1
Attention-map regulariser kept in Stage 2	New	Failure mode in Section 3.2
Cosine distillation-temperature schedule	Recipe choice, not in MOHAWK	Table 28: 80.73 vs. 80.48
Augmentation stack under pure soft-KL targets	Vision adaptation	Table 28: $+0.77$
Low-data distillation budget	Shared aim (MOHAWK: $\sim$3B tokens)	Table 1: 97.4% / 98.0% recovery at 10% data
Second transfer to dense prediction with an inherited head	New	Table 3

O Extension to headless self-supervised teachers (not validated)

The validated scope of this paper is ViT teachers with an inheritable CLS-to-logits path (Section 7). For a headless self-supervised ViT such as DINOv2 (Oquab et al., 2024), the recipe is defined as follows; we have not run it and make no claim about its outcome.

- **Architecture.** Skeleton inheritance is unchanged (patch size and extra tokens follow the teacher). The double-CLS initialisation reduces to plain CLS inheritance, since there is no classifier head whose input must be reproduced; extra tokens such as registers are treated like CLS positions in the Stage-1 loss.

- **Stages 1–2.** Unchanged: attention maps and per-block mixer outputs exist in any ViT, and both stages stop at the mixer output, never touching the classification path.
- **Stage 3.** The logit KL is replaced by feature alignment to the frozen teacher backbone on the final-block CLS and patch tokens (optionally on intermediate blocks as well), with the same trainable subset minus the classifier head.
- **Evaluation.** Teacher recovery is measured under the same linear probe (optionally an attentive probe) for teacher and student, instead of logit agreement or top-1 of an inherited head.

P STAGE-3 RECIPE ABLATION

Within the soft-KL stage we change one setting at a time (Table 28). The full augmentation stack (RandAugment + RandomErasing + MixUp/CutMix) gives the largest gain ($+0.77$ pp) even with pure soft-KL targets. An EMA shadow changes final-epoch top-1 by only -0.10 pp; its effect is mid-training, where it leads the live trajectory in teacher agreement by up to $+1.2$ pp on B 10% before both converge (Appendix H). Fixed temperatures ($T = 4, 3$) do not improve top-1 and are omitted; the cosine schedule $T : 5 \to 2$ adds $+0.25$ pp and is kept in the final recipe. As it had not fully converged at 100 epochs, the final runs use 150 epochs with T_{end} lowered to 1.5 to keep the cosine sweep proportional.

Table 28: Stage-3 recipe ablation on B 10%, Stage 3, 100 epochs; each row changes one setting and starts from the same Stage-2 checkpoint.

Change vs. previous row	Aug	EMA	T	Top-1	Δ vs. baseline
baseline (no aug, no EMA, $T=5$)	—	—	5.0 fixed	79.81	—
+ full aug stack	✓	—	5.0 fixed	80.58	+0.77
+ EMA (decay 0.9998)	✓	✓	5.0 fixed	80.48	+0.67
cosine $T : 5 \to 2$	✓	✓	cos $5 \to 2$	**80.73**	**+0.92**