ConvNeXt Project

Muhammad Qais & Guy Shiff

February 11, 2026

1 Introduction

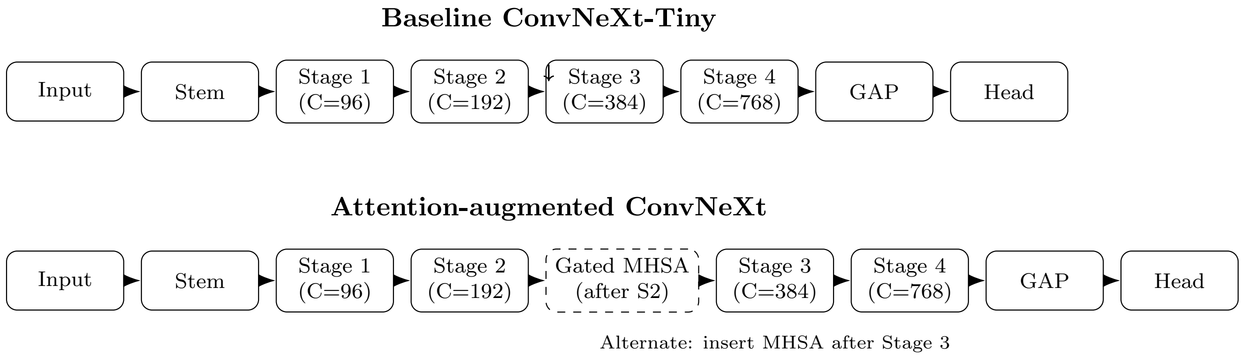
ConvNeXt is a convolutional architecture that incorporates several design and training choices that became common in recent large-scale vision models, while remaining fully convolutional. In this project, we evaluate ConvNeXt-Tiny on image-classification benchmarks such as CIFAR-100 under several training settings, including ImageNet fine-tuning and stage-wise freezing/unfreezing schedules. We study whether adding lightweight attention blocks to a strong ConvNeXt backbone yields consistent gains under comparable compute and parameter budgets.

1.1 ConvNeXt recap

ConvNeXt is a hierarchical CNN with four stages, where spatial resolution decreases while channel width increases. We use the TorchVision implementation of ConvNeXt-Tiny as our baseline. Its stage widths are $C = (96, 192, 384, 768)$ with block depths $B = (3, 3, 9, 3)$. The network starts with a strided convolutional stem, followed by stages of ConvNeXt blocks separated by downsampling layers, and ends with global average pooling and a linear classifier. The model has 28,589,128 parameters in TorchVision.

1.2 Attention-augmented ConvNeXt

We use TorchVision ConvNeXt-Tiny (without added attention) as the baseline and compare it to attention-augmented variants. Specifically, we insert a gated multi-head self-attention (MHSA) module at two insertion points—after Stage 2 or after Stage 3—and train these variants under several freezing/unfreezing strategies. The gate is initialized near zero so the attention path has minimal influence at the start of training; during optimization, the model can increase the gate if the attention module improves performance.


Baseline ConvNeXt-Tiny

Attention-augmented ConvNeXt

Alternate: insert MHSA after Stage 3

Figure 1: Compact architecture summary with channel dimensions. We compare baseline ConvNeXt-Tiny to attention-augmented variants with gated MHSA inserted after Stage 2 (main) or after Stage 3 (alternate).

2 Attention Injection Experiments

2.1 Single Attention Block (E1–E6) vs. Baseline

We fine-tune ImageNet-pretrained ConvNeXt-Tiny on CIFAR-100 for 20 epochs. The no-attention baseline reaches **77.24%** best test accuracy.

To isolate the effect of attention, we inject a single *gated* MHSA block with a learnable residual scale α initialized at 0:

$$x \leftarrow x + \alpha\,\mathrm{MHSA}(\mathrm{LN}(\mathrm{tokens}(x))).$$

We evaluate two insertion points: **P2** (after Stage 2, C=192) and **P3** (after Stage 3, C=384), and three freezing regimes: **F0** (backbone frozen throughout), **F1** (unfreeze only the last downsampling+Stage 4 after a 5-epoch warmup), and **F2** (unfreeze the full backbone after warmup with a smaller LR on backbone parameters).

All single-block variants underperform the baseline. Frozen/partially-frozen regimes degrade accuracy substantially (**63.70–69.42%**), while full unfreezing is best but still below baseline, reaching **75.33%** (P2) and **75.60%** (P3). In all runs α moves away from zero, indicating the attention branch is used, but it does not translate into higher accuracy under this budget.

Experiment	Placement	Regime	Best Test Acc.
E1_P2_F0	P2 (after S2)	F0	66.52%
E2_P2_F1	P2 (after S2)	F1	69.42%
E3_P2_F2	P2 (after S2)	F2	75.33%
E4_P3_F0	P3 (after S3)	F0	63.70%
E5_P3_F1	P3 (after S3)	F1	65.18%
E6_P3_F2	P3 (after S3)	F2	75.60%
Baseline (no attn)	—	—	**77.24%**

Table 1: Single attention block grid on CIFAR-100 (20 epochs).

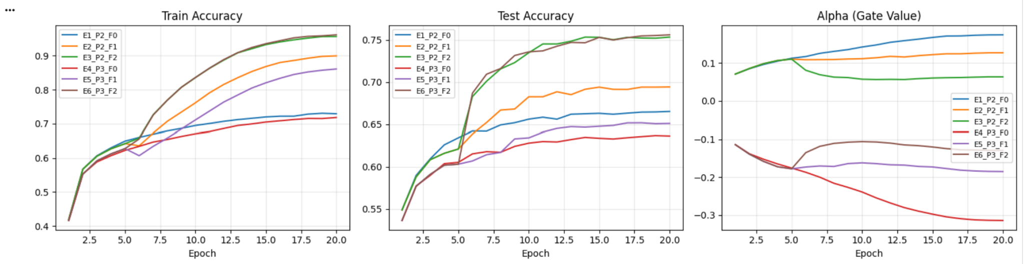


Figure 2: E1–E6 (single-block) learning curves. The dashed line marks the Phase 2 unfreezing (epoch 6) in F1/F2, after which F2 variants show the largest gains in test accuracy. The gate parameter α moves away from zero across runs, indicating that the injected attention branch is actively used, though this does not necessarily translate into higher accuracy.

2.2 Experiment A: Dual Attention (P2 + P3)

Single-block attention may be too limited, so Experiment A inserts **two** gated MHSA blocks—after Stage 2 (P2) and after Stage 3 (P3)—and trains with the same two-phase schedule as F2: a 5-epoch warm-up with the backbone frozen, followed by full unfreezing with differential learning rates. Dual attention reaches **75.93%** best observed test accuracy over 20 epochs. This slightly improves over the best single-block F2 runs and clearly outperforms the single-block frozen/partially-unfrozen settings, but it remains below the baseline (**77.24%**). After unfreezing, both gate parameters stabilize at small non-zero magnitudes, indicating that both attention branches remain active once the backbone adapts, yet the overall gain is not sufficient to close the gap to the strong convolutional baseline under this training budget.

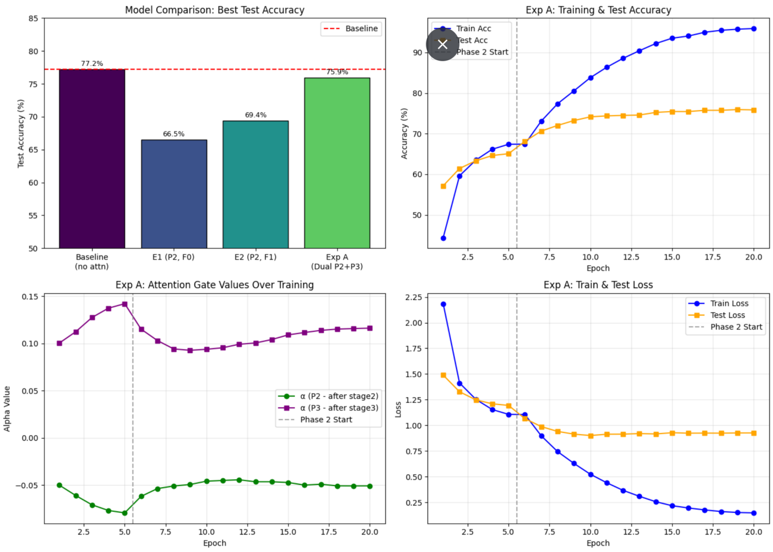


Figure 3: Experiment A summary and comparison to baseline and single-block variants.

2.3 Qualitative Attention Visualization (planned)

We visualize attention allocation from the two MHSA blocks in Experiment A using the best saved checkpoint. At inference time, we run a modified MHSA module that returns per-head attention matrices over the $L = H \times W$ spatial tokens (no CLS token). We average attention weights across heads and compute, for each token, the mean attention it *receives* across all query tokens, producing a spatial map at the feature resolution (P2: 8×8, P3: 4×4). The maps are upsampled to 32×32 and overlaid on the input image to qualitatively compare P2 vs. P3 behavior and contrast correct vs. incorrect predictions.

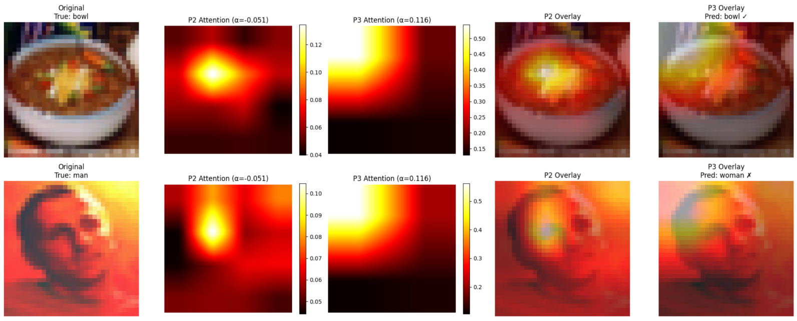


Figure 4: Experiment A: **Spatial importance maps** induced by the injected MHSA blocks (P2 vs. P3), shown as heatmaps and input overlays for representative correct () and incorrect (×) examples. Qualitatively, P2 tends to produce more *localized* saliency around the main object, whereas P3 often yields *broader, smoother* importance patterns that can extend into background regions. The misclassified example illustrates that the attention pathway may still emphasize non-discriminative cues, so these visualizations are best interpreted as *diagnostic* evidence of what the model attends to rather than a guarantee of improved accuracy.

2.4 Experiment B: Attention Head Count Ablation

We next ablated the number of attention heads in the dual-attention model (P2+P3), while keeping the architecture and training recipe fixed. Since multi-head attention partitions the same feature dimension across different head groupings (without changing the projection sizes), this experiment tests whether the head factorization itself affects performance on CIFAR-100 under the same 20-epoch budget.

Head count has a negligible effect: all configurations cluster tightly around **75.8–75.9%** best observed test accuracy and remain below the baseline ConvNeXt-Tiny (**77.24%**). The best setting is **1 head** at

75.94%, but differences across head counts are within a few tenths of a percent. Gate values are also consistent across runs: P2 converges to a small negative value ($\alpha_{P2} \approx -0.04$ to -0.05), while P3 converges to a modest positive value ($\alpha_{P3} \approx 0.11$ to 0.13). Overall, these results suggest that **head count is not a key driver** in this setup, so we focus next on other design choices.

Heads	Attn Params	Best Acc.	Final α_{P2}	Final α_{P3}
1	740,738	75.94%	-0.0394	0.1061
2	740,738	75.79%	-0.0496	0.1100
4	740,738	75.93%	-0.0510	0.1161
8	740,738	75.80%	-0.0532	0.1260
Baseline (no attn)	—	**77.24%**	—	—

Table 2: Experiment B: dual-attention (P2+P3) with different numbers of attention heads. All variants remain below the no-attention baseline.

2.5 Experiment C: Alternative Attention Mechanisms (P3 Only)

In Experiment C we fix the insertion point to **P3** (after Stage 3, C=384) and keep the same two-phase training schedule (5-epoch warmup with a frozen backbone, then full unfreezing with differential learning rates). We restrict this ablation to P3 because, in the single-block grid, the best P3 setting was slightly stronger than P2 under full unfreezing, allowing us to focus the comparison on the *attention mechanism* rather than placement.

Results. We test five P3 modules: gated MHSA (1 head), SE, ECA, CBAM, and a spatial-only block. Best observed test accuracies are tightly clustered: MHSA **75.63%**, ECA **75.58%**, CBAM **75.50%**, SE **75.44%**, and Spatial **75.40%**. The spread is only ≈ 0.23%, and none of the mechanisms closes the gap to the no-attention baseline (**77.24%**), suggesting that under this 20-epoch budget the specific attention choice has only a marginal effect on accuracy.

Added parameter cost varies substantially: MHSA adds **592,129** parameters, SE **18,433**, CBAM **18,531**, Spatial **99**, and ECA only **6**. Given essentially tied accuracy, ECA is the most attractive option when efficiency matters, even though no mechanism improves over the baseline.

Attention module (P3)	Best test acc.	Added params
Gated MHSA (1-head)	75.63%	592,129
SE	75.44%	18,433
ECA	75.58%	6
CBAM	75.50%	18,531
Spatial	75.40%	99
Baseline (no attn)	**77.24%**	—

Table 3: Experiment C (P3-only): accuracy vs. added parameter cost for alternative attention mechanisms.

2.6 CIFAR stem patch and validation-based checkpointing

We also evaluated a CIFAR-100–adapted ConvNeXt-Tiny setup that modifies the input stem for 32×32 images. The default ConvNeXt stem (kernel 4, stride 4, padding 0) down-samples aggressively, so we patch it to stride 2 with padding 1 to preserve higher early spatial resolution. We additionally use label smoothing ($\epsilon = 0.1$) and select the final checkpoint by validation accuracy using a 10% split of the CIFAR-100 training set (no augmentation on validation). Under this setup, the best checkpoint reaches 85.76% test accuracy (best validation 86.16% at epoch 18). Importantly, repeating the same pipeline without attention achieves essentially the same accuracy, indicating that the improvement is driven by the CIFAR-specific stem/training recipe rather than by the added attention blocks; consistent with this, the learned attention gates remain near zero, suggesting the attention branches are largely unused.

3 Training from Scratch (with vs. without Attention)

To test whether attention helps when representations are learned end-to-end (rather than starting from ImageNet features), we also trained ConvNeXt-Tiny from scratch on CIFAR-100 (`weights=None`). In all scratch runs we kept the CIFAR stem patch enabled, modifying the input stem from stride 4 to stride 2 and padding $0 \to 1$, since CIFAR inputs are only 32×32 and aggressive early downsampling can remove spatial detail.

We first trained a no-attention scratch baseline (planned 100 epochs; label smoothing $\epsilon = 0.1$) but stopped early after validation accuracy plateaued around $\sim$ 53–55% by epochs 40–54 (best 54.56% at epoch 53). We then trained a matched scratch + attention variant, injecting gated MHSA blocks after Stage 2 (P2, C=192) and after Stage 3 (P3, C=384) using a residual-gated form $x + \alpha\,\mathrm{MHSA}(x)$ with α initialized at zero. Under a 50-epoch budget, this model reached a best validation accuracy of 48.56% (epoch 49) and a test accuracy of 49.44%. The learned gates drifted slightly negative and stabilized near zero ($\alpha_{P2} \approx -0.020$, $\alpha_{P3} \approx -0.016$), suggesting the attention branches were only weakly utilized.

Overall, these scratch runs are far below the ImageNet fine-tuning setting, indicating that pretraining and training scale dominate performance here; under the tested budgets, adding MHSA does not close the gap.

3.1 ViT vs. ConvNeXt and MLP-Mixer (CIFAR-100, from scratch)

To reduce the confound of pretraining and focus on architectural differences, we trained two token-based models from scratch on CIFAR-100 using the same scratch recipe as our ConvNeXt runs (AdamW, cosine LR, label smoothing ϵ=0.1, 50 epochs, and the same data augmentation and evaluation protocol). We used a ViT-Tiny and an MLP-Mixer-Tiny with identical patchification: 32×32 images are split into 4×4 patches, yielding 8×8=64 tokens, with embedding dimension d=192 and depth 12. This isolates the mixing mechanism: ViT mixes tokens via multi-head self-attention, whereas MLP-Mixer replaces attention with a token-mixing MLP followed by a channel-mixing MLP. The models are comparable in scale (ViT-Tiny $\approx$ 5.38M params; MLP-Mixer-Tiny $\approx$ 3.16M params in our implementations). Under this budget, both plateaued around $\sim$ 41% test accuracy, with the mixer closely matching ViT. In contrast, scratch ConvNeXt performed substantially better, suggesting that on CIFAR-100 at this training scale, convolutional inductive bias (and/or optimization/data scale) matters more than the presence of attention by itself.

3.2 Matched-capacity ConvNeXt vs. ViT (CIFAR-100, from scratch)

The previous scratch comparison (ViT-Tiny / Mixer-Tiny) could be criticized as a capacity mismatch, since ConvNeXt-Tiny has substantially more parameters than typical "Tiny" token models. To control for this, we ran an additional **matched-parameter** experiment in which we trained **ConvNeXt-Tiny (scratch)** and a **Vision Transformer (scratch)** with *approximately the same number of parameters* under an identical training recipe (AdamW, cosine schedule with warmup, label smoothing ϵ=0.1, 30 epochs, and the same CIFAR-100 augmentation/normalization). For ConvNeXt we used `weights=None` and kept the **CIFAR stem patch enabled** (stride $4 \to 2$, padding $0 \to 1$) to avoid overly aggressive early downsampling on 32×32 inputs. For ViT, we used patch size 4 and performed a small search over embedding dimension and head count *only to match the ConvNeXt-Tiny parameter count* (depth fixed at 12, MLP ratio 4). Despite the matched scale, ConvNeXt remained clearly stronger in the scratch regime: ConvNeXt-Tiny reached a best test accuracy of **55.63%**, while the matched-parameter ViT reached **49.29%** (gap $\approx$ 6.3 points). In our implementation, ViT was also slower per epoch (roughly 2×), reinforcing the practical advantage of the convolutional inductive bias under this training budget. Overall, this experiment suggests the scratch performance gap is not explained solely by parameter count, but is consistent with ConvNeXt benefiting from a more suitable inductive bias for small natural images when trained without large-scale pretraining.

4 Fashion-MNIST Experiments

We repeated the attention-injection study on **Fashion-MNIST** (10 classes; 60,000 train / 10,000 test). Images are 28×28 grayscale; we resize to 32×32 and replicate channels to match ConvNeXt's RGB input. We fine-tuned ImageNet-pretrained ConvNeXt-Tiny for 20 epochs using the same two-phase recipe as

CIFAR-100 (5-epoch warmup with a frozen backbone, then full unfreezing with differential learning rates). The no-attention baseline achieved **94.56%** test accuracy.

Across all ablations, attention did not improve performance. Dual gated MHSA at P2+P3 reached **94.37%** (−0.19%), with gates $\alpha_{P2} \approx -0.071$ and $\alpha_{P3} \approx +0.034$. Varying head count (1/2/4/8) had negligible effect and all settings remained below baseline. Alternative P3 mechanisms were also tightly clustered (**94.29–94.43%**); the best was Spatial-only at **94.43%** (−0.13%), while lightweight channel attention (SE/ECA) matched closely (**94.42%**), with ECA adding only **6** parameters (vs. 592,129 for MHSA).

Experiment	Configuration	Best Test Acc.	Δ vs. Baseline
Baseline	No attention	**94.56%**	—
A	Dual MHSA (P2+P3)	94.37%	−0.19%
B	Heads (1/2/4/8)	94.31–94.37%	−0.25 to −0.19
C	P3 attention type	94.29–94.43%	−0.27 to −0.13

Table 4: Fashion-MNIST attention ablations (20 epochs, ImageNet-pretrained ConvNeXt-Tiny). No attention variant exceeds the baseline.

Compared to CIFAR-100 (dual-MHSA drop of −1.31%), the penalty on Fashion-MNIST is smaller (−0.19%), suggesting attention is less harmful on the simpler dataset but still not beneficial under our setup.

5 Limitations and Conclusions

Main takeaway. Across CIFAR-100 and Fashion-MNIST, adding attention to ConvNeXt-Tiny did *not* improve over the no-attention baseline under our budgets. On CIFAR-100 (ImageNet fine-tuning, 20 epochs), the baseline reached **77.24%**, while the best attention variants stayed lower (best single-block **75.60%**; dual attention **75.93%**). On Fashion-MNIST, the baseline was already very strong (**94.56%**) and all attention variants were essentially tied but slightly worse (**94.29–94.43%**). Overall, with a strong pretrained convolutional backbone, attention added complexity without measurable gains.

What mattered in practice. The fine-tuning regime dominated outcomes: attention variants were only competitive when the backbone was fully unfrozen (F2), while freezing/partial unfreezing (F0/F1) caused large drops on CIFAR-100 (**63.70–69.42%**), suggesting inserted attention paths are not plug-and-play with pretrained ConvNeXt features.

Scratch regime results. From-scratch CIFAR-100 performance was much lower and attention did not help: scratch ConvNeXt-Tiny plateaued around **53–55%**, while scratch ConvNeXt+MHSA reached **49.44%** test (best val **48.56%**). Token-based scratch baselines (ViT/MLP-Mixer) plateaued around ∼**41%**; even with matched parameters, ViT (**49.29%**) trailed scratch ConvNeXt (**55.63%**) under a 30-epoch budget, consistent with convolutional inductive bias being stronger for small images under limited compute.

Limitations. (i) Budgets are short (20–50 epochs), and some scratch runs were stopped after clear plateauing, so results reflect the explored schedules rather than asymptotic performance; (ii) we did not perform exhaustive hyperparameter tuning (LR/WD/warmup/augmentation or attention-specific design choices); (iii) token-based baselines (especially MLP-Mixer) were not evaluated at larger capacities or via a scaling sweep.

Conclusion. Under the tested conditions, performance is driven primarily by pretraining and the convolutional backbone, and inserting attention into ConvNeXt-Tiny does not provide a net benefit. Future work could test longer schedules, larger-scale data, or architectures that integrate attention more natively rather than as an added residual branch.