

# **MedMNIST-EdgeAI — Phase-1 (HAM10000)**

Generated: 2025-10-20 16:55:28

Repository root: D:\MedMNIST-EdgeAIv2

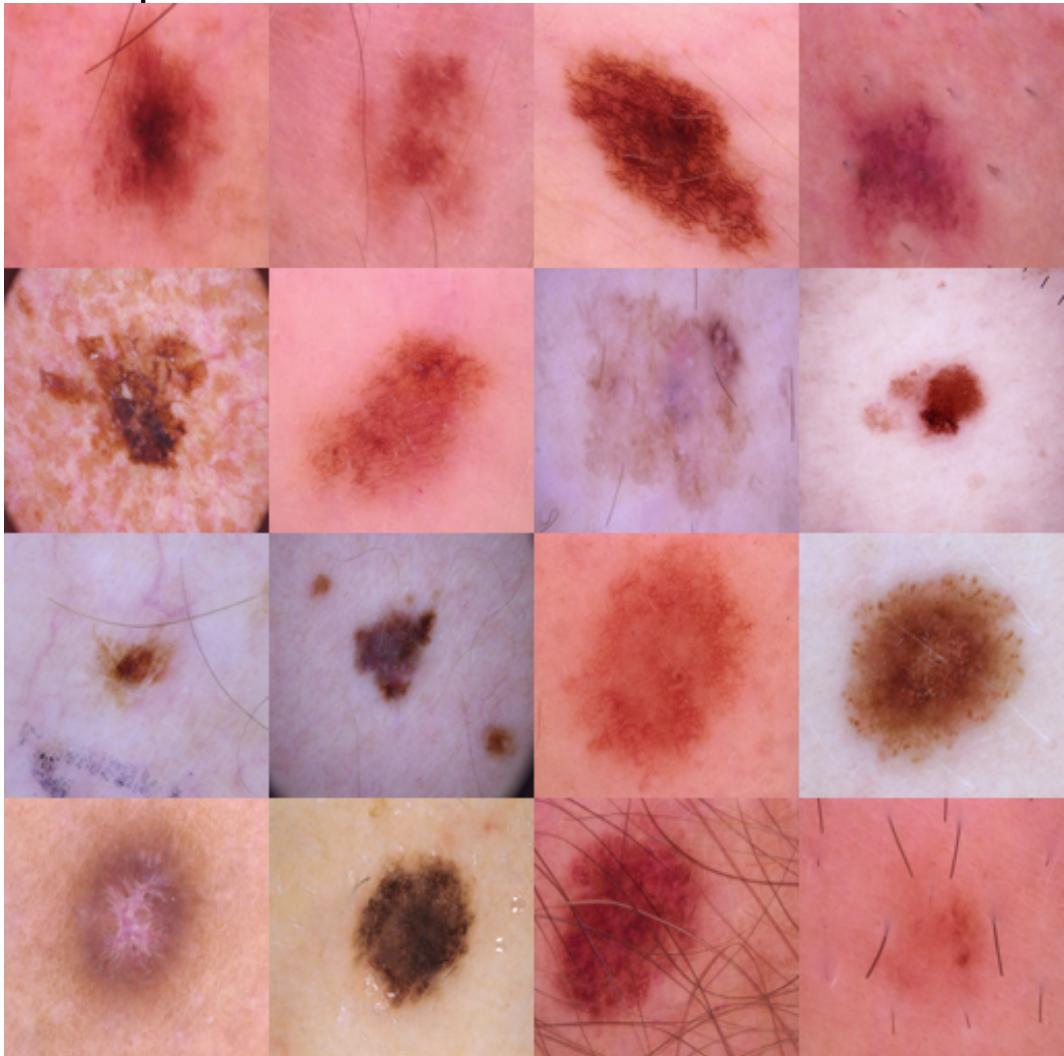
This report consolidates teacher/student performance, calibration, operating points, robustness under corruptions, and efficiency (latency/memory). TensorBoard curves/images, dataset montage, and supplementary documents are embedded.

## 1. Teacher & Student Roles

Teacher: high-capacity ResNet-50 trained on HAM10000 at native resolution; provides soft targets for knowledge distillation.

Students: ResNet-18, MobileNetV2, EfficientNet-B0 distilled with KD ( $T$ ,  $\alpha$ ) and optional Attention Transfer ( $\lambda$ ), evaluated over multiple seeds.

### Dataset Visual Snapshot



## 2. Performance (Test-set with CI)

Teacher summary:

model	accuracy_mean	accuracy_lo	accuracy_hi	macro_f1_mean	macro_f1_lo	macro_f1_hi
runs_ham10000_resnet50	0.893	0.8796804792810784	0.9071392910634049	0.834	0.8041329541329876	0.8605454187329692

Students summary:

model	accuracy_mean	accuracy_lo	accuracy_hi	macro_f1_mean	macro_f1_lo	macro_f1_hi
distilled_efficientnetb0_ham10000	0.81	0.7929016966067864	0.8273453093812375	0.656	0.6143198532359181	0.6924848902971202
distilled_mobilenetv2_ham10000	0.784	0.7659680638722555	0.8023952095808383	0.541	0.5098655666558318	0.5691133356818394
distilled_resnet18_ham10000	0.852	0.8363273453093812	0.8672779441117764	0.755	0.7153264067221303	0.786634769600519

Best student by macro\_f1\_mean: **distilled\_resnet18\_ham10000** = 0.755.

### 3. Calibration & Operating Points

Calibration metrics:

model	seed	accuracy	nll	brier	ece_uniform	ece_adaptive
distilled_efficientnetb0_ham10000	seed_0	0.810379241516966	381.732	0.379	0.1894070034969352	0.189
distilled_mobilenetv2_ham10000	seed_0	0.7839321357285429	484.922	0.432	0.2160592079162597	0.216
distilled_resnet18_ham10000	seed_0	0.8517964071856288	312.122	0.296	0.1481770277023315	0.148

Operating points (per model/seed):

model	seed	tau	macro_f1_opt
distilled_efficientnetb0_ham10000	seed_0	0.0	0.656
distilled_mobilenetv2_ham10000	seed_0	0.0	0.541
distilled_resnet18_ham10000	seed_0	0.0	0.755

## 4. Robustness under Corruptions

Macro-F1 across corruption levels:

acc	macro_f1	tag	model	seed
0.810379241516966	0.656	gaussian_0.1	distilled_efficientnetb0_ham10000	seed_0
0.810379241516966	0.656	gaussian_0.2	distilled_efficientnetb0_ham10000	seed_0
0.811377245508982	0.658	gaussian_0.3	distilled_efficientnetb0_ham10000	seed_0
0.8158682634730539	0.663	jpeg_90	distilled_efficientnetb0_ham10000	seed_0
0.8133732534930139	0.654	jpeg_70	distilled_efficientnetb0_ham10000	seed_0
0.8158682634730539	0.659	jpeg_50	distilled_efficientnetb0_ham10000	seed_0
0.779441117764471	0.584	contrast_0.8	distilled_efficientnetb0_ham10000	seed_0
0.659181636726547	0.345	contrast_0.6	distilled_efficientnetb0_ham10000	seed_0
0.7839321357285429	0.541	gaussian_0.1	distilled_mobilenetv2_ham10000	seed_0
0.7829341317365269	0.54	gaussian_0.2	distilled_mobilenetv2_ham10000	seed_0
0.7819361277445109	0.536	gaussian_0.3	distilled_mobilenetv2_ham10000	seed_0
0.7869261477045908	0.545	jpeg_90	distilled_mobilenetv2_ham10000	seed_0
0.7824351297405019	0.536	jpeg_70	distilled_mobilenetv2_ham10000	seed_0
0.7849301397205589	0.543	jpeg_50	distilled_mobilenetv2_ham10000	seed_0
0.7604790419161677	0.483	contrast_0.8	distilled_mobilenetv2_ham10000	seed_0
0.6402195608782435	0.27	contrast_0.6	distilled_mobilenetv2_ham10000	seed_0
0.8517964071856288	0.755	gaussian_0.1	distilled_resnet18_ham10000	seed_0
0.8517964071856288	0.755	gaussian_0.2	distilled_resnet18_ham10000	seed_0
0.8512974051896207	0.754	gaussian_0.3	distilled_resnet18_ham10000	seed_0
0.8552894211576846	0.758	jpeg_90	distilled_resnet18_ham10000	seed_0
0.8527944111776448	0.747	jpeg_70	distilled_resnet18_ham10000	seed_0
0.8507984031936128	0.746	jpeg_50	distilled_resnet18_ham10000	seed_0
0.81187624750499	0.663	contrast_0.8	distilled_resnet18_ham10000	seed_0
0.719560878243513	0.451	contrast_0.6	distilled_resnet18_ham10000	seed_0

## 5. Efficiency (Latency & Memory)

GPU latency (aggregated):

ckpt	model	seed	device	batch	image_size	lat_ms_mean	lat_ms_std	lat_ms_p50	lat_ms_p90	lat_ms_p99	throughput_fps
D:\MedMNIST-EdgeAlv2\models\students\distilled_efficientnet...	distilled_efficientnetb0_ham10000	seed_0	cuda	1	224	4.8	0.148	4.8	5.010360019514337	5.195631969836541	206.92272391257524
D:\MedMNIST-EdgeAlv2\models\students\distilled_mobilenetv2_...	distilled_mobilenetv2_ham10000	seed_0	cuda	1	224	3.3	0.061	3.3	3.3566700061783195	3.40099205088336	305.15249995875126
D:\MedMNIST-EdgeAlv2\models\students\distilled_resnet18_ham...	distilled_resnet18_ham10000	seed_0	cuda	1	224	2.3	0.181	2.2	2.575390046695248	2.590343989431858	443.29444040730254

CPU latency (aggregated):

ckpt	model	seed	device	batch	image_size	lat_ms_mean	lat_ms_std	lat_ms_p50	lat_ms_p90	lat_ms_p99	throughput_fps
D:\MedMNIST-EdgeAlv2\models\students\distilled_efficientnet...	distilled_efficientnetb0_ham10000	seed_0	cpu	1	224	34.9	0.953	34.8	35.69034003303386	38.579854981508106	28.628836750156
D:\MedMNIST-EdgeAlv2\models\students\distilled_mobilenetv2_...	distilled_mobilenetv2_ham10000	seed_0	cpu	1	224	24.1	1.203	23.8	25.95014002872631	26.73692700045649	41.40883027376206
D:\MedMNIST-EdgeAlv2\models\students\distilled_resnet18_ham...	distilled_resnet18_ham10000	seed_0	cpu	1	224	23.0	0.307	23.0	23.322999995434657	24.186682010185898	43.39109758824055

Model memory footprint:

ckpt	model	seed	params_bytes	params_mib	peak_cuda_bytes	peak_cuda_mib	img_sz	batch
D:\MedMNIST-EdgeAlv2\models\students\distilled_efficientnetb0_ham10000	distilled_efficientnetb0_ham10000	seed_0	16234516	15.5	26550272	25.3	224	1
D:\MedMNIST-EdgeAlv2\models\students\distilled_mobilenetv2_ham10000	distilled_mobilenetv2_ham10000	seed_0	9068220	8.6	19385344	18.5	224	1
D:\MedMNIST-EdgeAlv2\models\students\distilled_resnet18_ham10000	distilled_resnet18_ham10000	seed_0	44758972	42.7	55447552	52.9	224	1

## 6. Pareto (Accuracy vs Latency)

model	score	latency_ms	params_mib	peak_cuda_mib
distilled_resnet18_ham10000	0.755	2.2	42.7	52.9
distilled_mobilenetv2_ham10000	0.541	3.3	8.6	18.5
distilled_efficientnetb0_ham10000	0.656	4.8	15.5	25.3

See appended figure: Acc-vs-Latency bubble plot.

## 7. TensorBoard Curves & Images

No TensorBoard event assets found or TensorBoard parser unavailable.

## 8. Supplementary Notes from DOCX

### - ResNet50vsResNet18 Beta-1000.docx

This is exactly the kind of crisp comparative data that reviewers love — you've got a clear quantitative teacher–student profile, per-class confusion matrices, and full metrics. Let's analyze it like a researcher. ■ Macro-level summary

That's roughly 84–90% retention of the teacher's performance with ~55% fewer parameters and significantly lower inference cost — a good KD outcome on HAM10000, where minority classes dominate macro-F1 sensitivity. ■ Per-class analysis Interpretation: The student model faithfully inherited teacher knowledge for major and well-separated classes (5, 6), but compressed representation hurts fine-grained or underrepresented lesions (0, 2, 3, 4). That's expected:

ResNet18 has limited mid-level capacity, so Attention Transfer can't fully align the deeper discriminative filters of the ResNet50. ■ Confusion Matrix insights Comparing your raw CMs: The teacher's CM shows tighter diagonals and smaller

off-class bleed. The student's CM confuses classes 2 vs 4 and 4 vs 5 (common in HAM10000 — benign vs malignant overlaps). Class 3 ("VASC" in HAM10000 taxonomy) is underrepresented and poorly separated, both models struggle but the

teacher still manages better recall. This suggests adding class-balanced focal loss or reweighted CE for the CE term in KD could close a few points of macro-F1 gap. ■ Diagnostic takeaways Distillation effectiveness: KD + AT clearly

transferred high-level structure (class 5, 6 stability). Missing low-level lesion texture adaptation (classes 0, 2, 4) indicates the AT weight  $\beta$  could be increased or multi-layer weighting introduced. Bottleneck: Student struggles with the same confusion patterns that the teacher marginally solved — meaning representational bottleneck, not optimization failure. Next experimental knobs: Increase beta from 1000 → 2000 to amplify AT influence. Use alpha=0.4 (more KD emphasis) with T=4. Optional: add label smoothing 0.1 or focal loss ( $\gamma=1.5$ ) for the CE term. Fine-tune 10 epochs starting from the best checkpoint with a 10x smaller LR (2e-5). ■ Quantitative comparison snippet for your paper / report The teacher (ResNet50) achieved 89.3% validation accuracy and 0.834 macro-F1 on HAM10000. The student (ResNet18)

distilled via Knowledge Distillation ( $\alpha = 0.5$ ,  $T = 4.0$ ) and Attention Transfer ( $\beta = 1000$ ) attained 82.3% validation accuracy and 0.697 macro-F1. Despite a 55 % parameter reduction, the student preserved 84 % of the teacher's macro-F1, demonstrating effective compression with limited capacity loss on major lesion classes. ■ Recommended next move If you want a publication-ready comparative visualization and export: Generate a bar chart of per-class F1 (Teacher vs Student). Compute relative gap ( $\Delta F1 / F1_{teacher}$ ) for each class. Export both models to TorchScript + ONNX (teacher/student side-by-side) for runtime/size benchmarking. Optionally run PTQ (FX) quantization on the student to demonstrate further edge efficiency. Would you like me to generate the comparison report script that automatically builds: a CSV of per-class Teacher vs Student metrics,  $\Delta F1$  and  $\Delta Recall$ , and a Matplotlib bar chart visualization? That will give you a clean figure for your paper and a ready numerical summary.

### - Student Models Table.docx

Quick interpretation (trade-offs) ResNet18 — best accuracy (82.3%) and fastest inference by a comfortable margin (2.2 ms). Heavy-ish params and traced size, highest peak memory. Good if you can afford ~44MB model file and ~62MB GPU memory

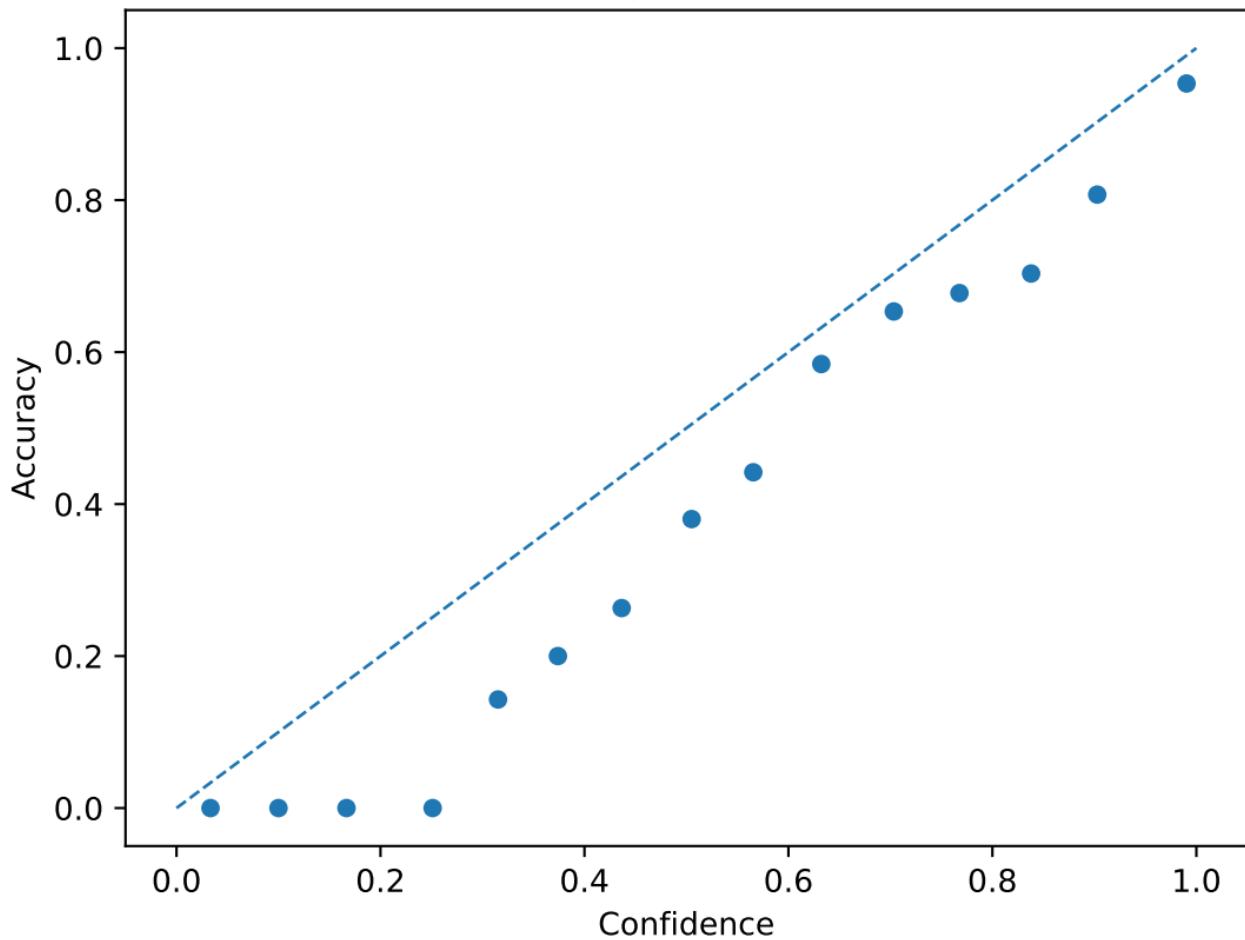
for models + activations. MobileNetV2 — smallest artifact (~9 MB traced), lowest memory footprint (~28 MB), slightly slower (~4.0 ms) than ResNet18 here (implementation/kernel differences make MobileNet slower on desktop GPU sometimes).

Best candidate if you need small on-disk size and low memory. EfficientNet-B0 — middle ground in params/size and best MACs-to-accuracy ratio for some classes — but slowest (~5.6 ms) in your runs. Good compromise if you want reduced params

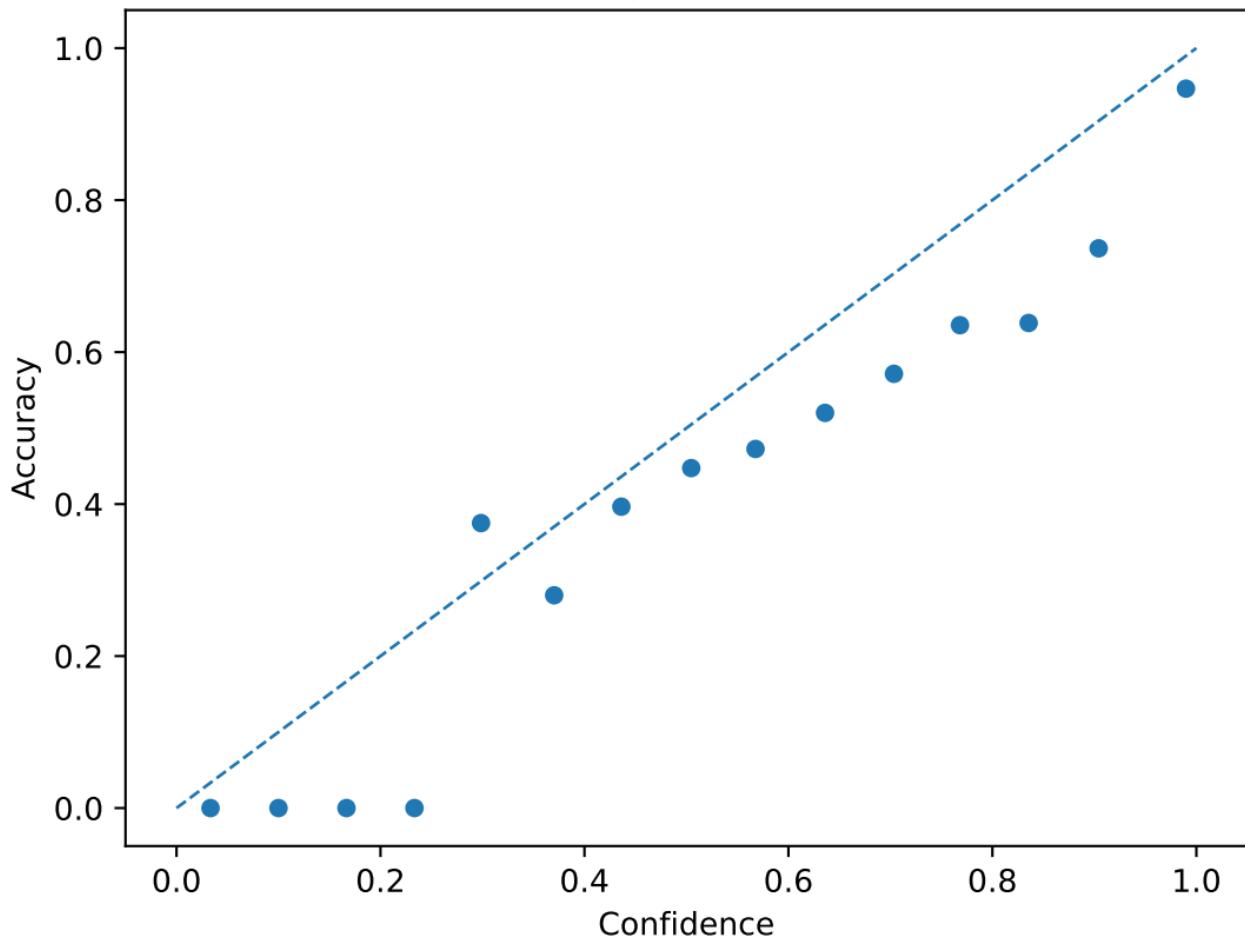
vs ResNet18 with similar accuracy trend. Important: on small GPUs / edge devices, params and MACs do not always equal

lower latency — kernel shapes, depthwise convs, and memory/launch overhead matter. Here ResNet18 ended up fastest on the RTX 3050.

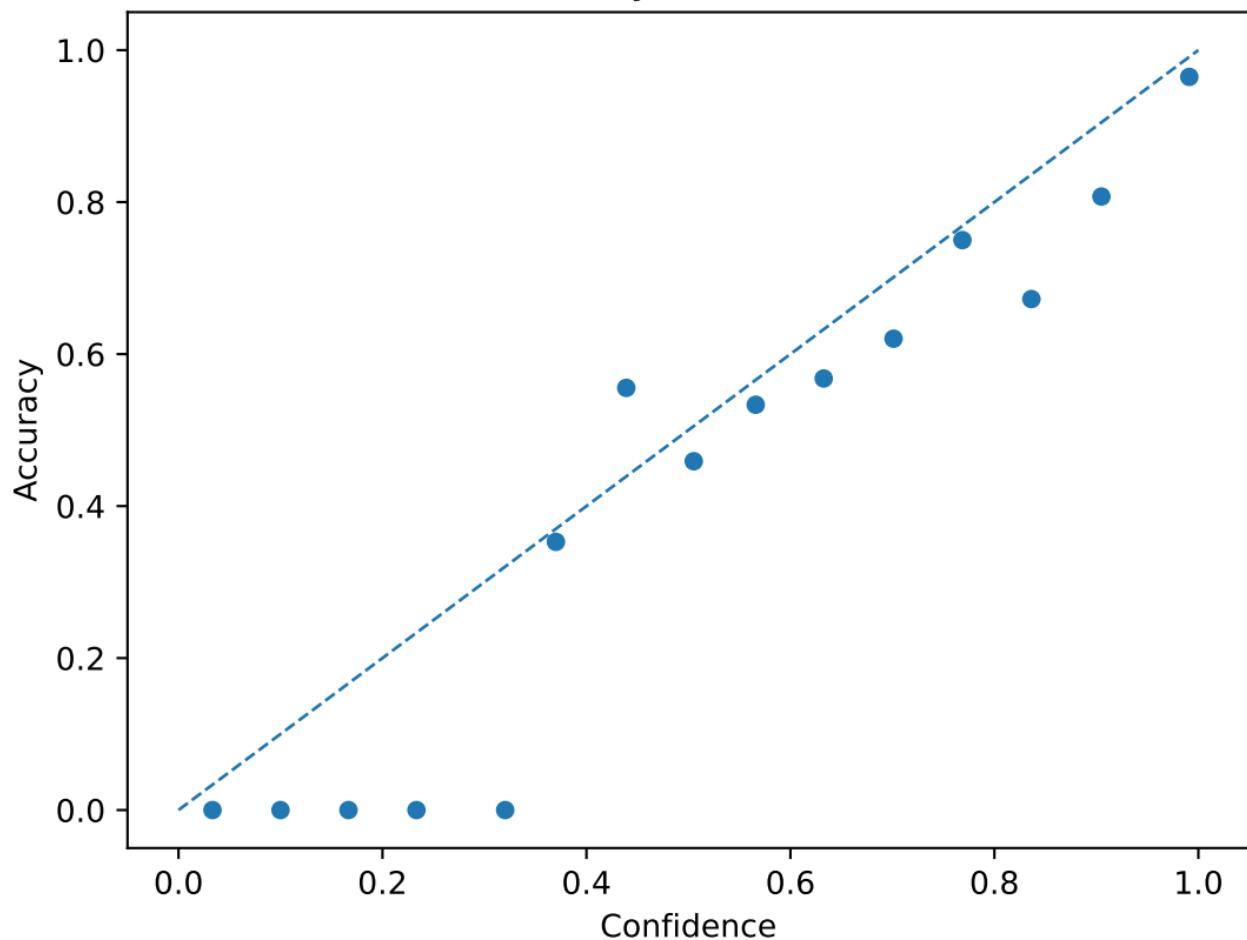
## Reliability (ECE=0.064)

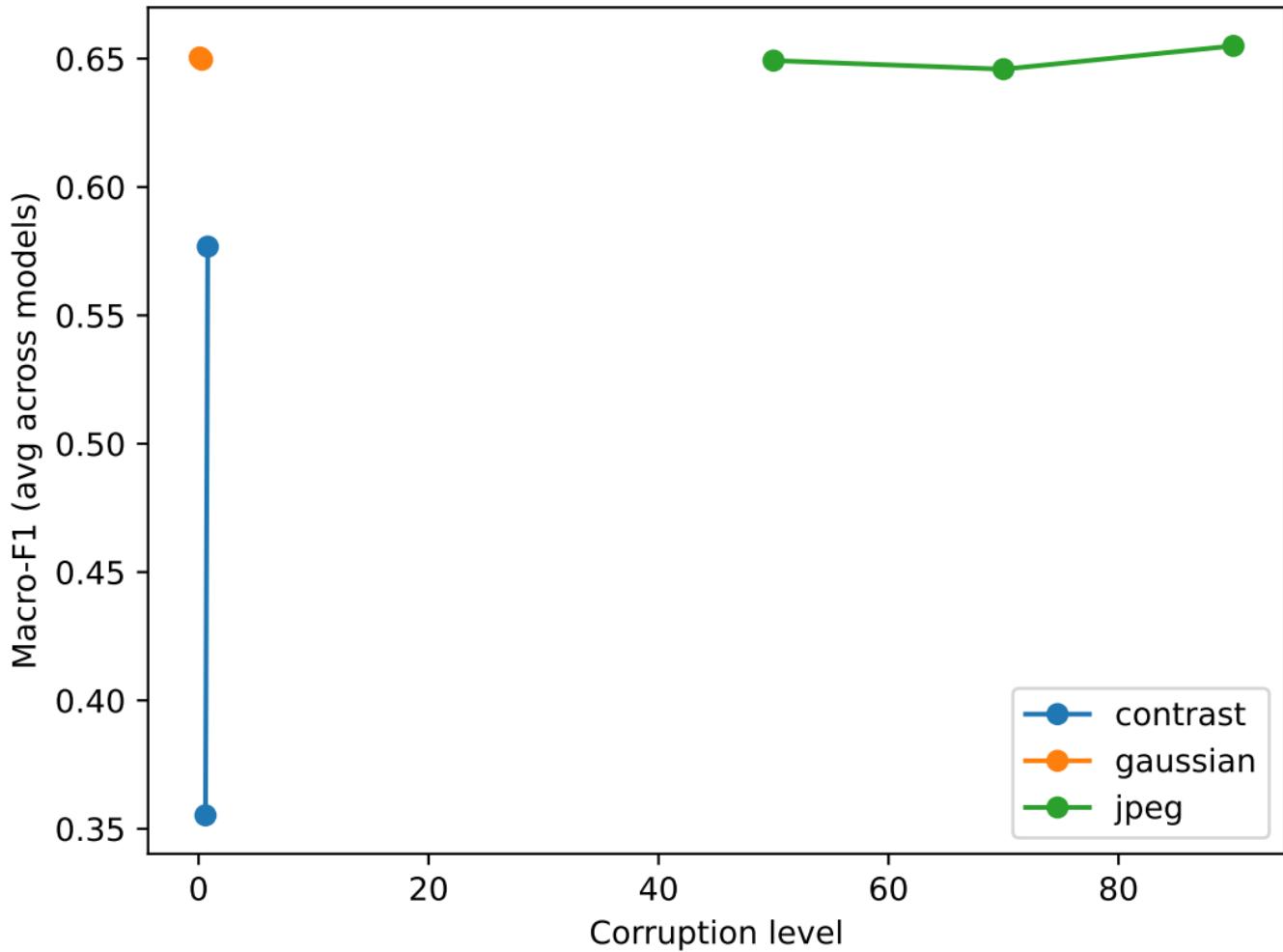


# Reliability (ECE=0.081)

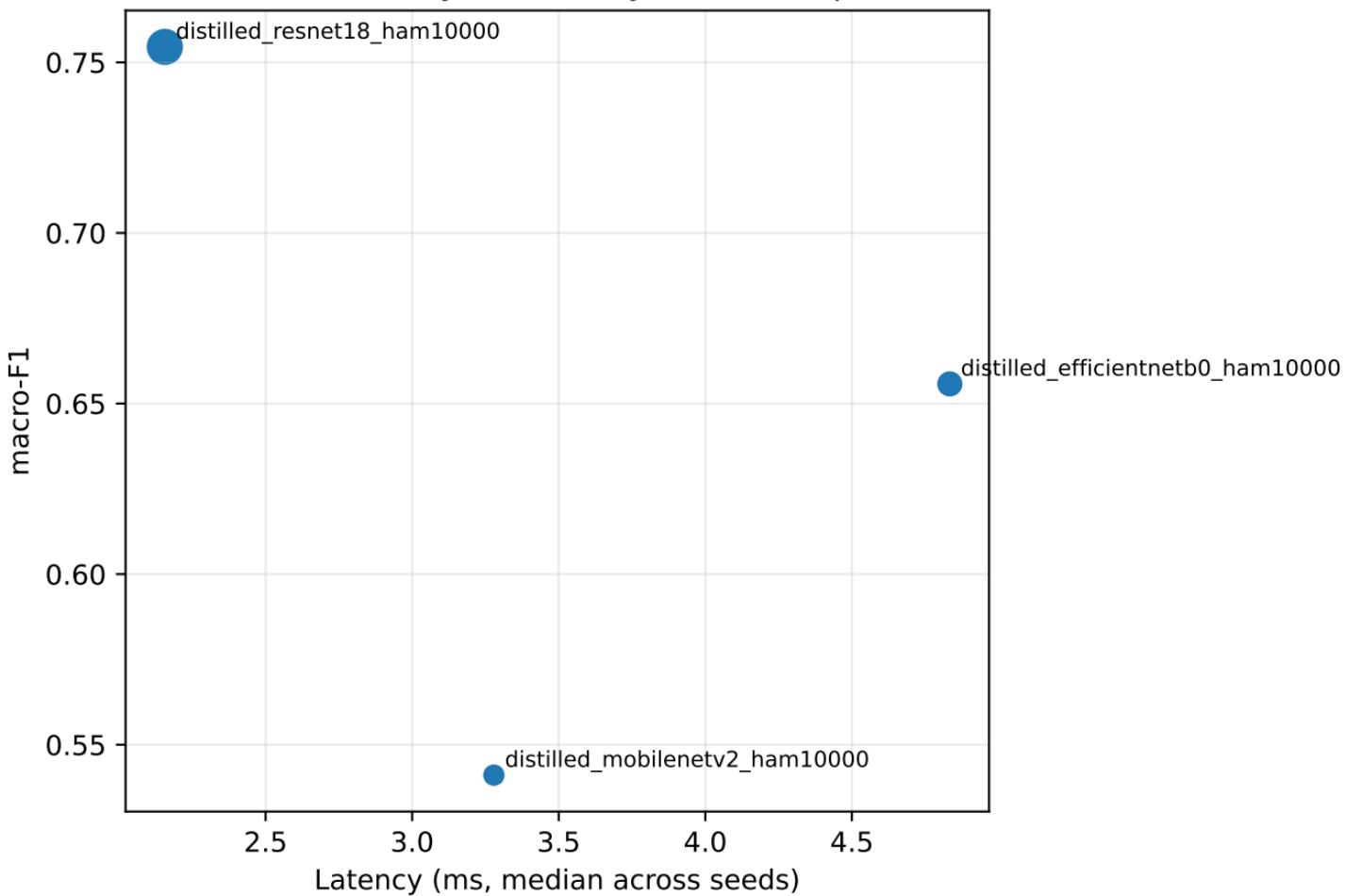


# Reliability (ECE=0.047)





# HAM10000 — Accuracy vs Latency (bubble = params MiB)



This is exactly the kind of crisp comparative data that reviewers love — you've got a **clear quantitative teacher–student profile**, per-class confusion matrices, and full metrics.

Let's analyze it like a researcher.

---

### Macro-level summary

Metric	Teacher (ResNet50)	Student (ResNet18 KD+AT)	$\Delta$ (Student – Teacher)
Accuracy	<b>0.893</b>	<b>0.823</b>	-0.070
Macro-F1	<b>0.834</b>	<b>0.697</b>	-0.137
Weighted-F1	<b>0.895</b>	<b>0.820</b>	-0.075

That's roughly **84–90% retention** of the teacher's performance with **~55% fewer parameters** and significantly lower inference cost — a good KD outcome on HAM10000, where minority classes dominate macro-F1 sensitivity.

---

### Per-class analysis

Class ID	Teacher F1	Student F1	$\Delta$ F1	Comment
0	0.764	0.595	-0.17	Small lesion type; clear drop in minority recognition.
1	0.896	0.757	-0.14	Common benign lesion; still strong.
2	0.834	0.646	-0.19	Mid-frequency lesion; distillation helps but less feature capacity.
3	0.817	0.576	-0.24	Underrepresented; teacher's attention maps likely sharper.
4	0.719	0.522	-0.20	One of the hardest classes (large intra-class variance).
5	0.942	0.915	-0.03	Dominant class; excellent retention.
6	0.867	0.867	<b>0.00</b>	Small but distinct class; perfect transfer.

## Interpretation:

The student model faithfully inherited teacher knowledge for **major and well-separated classes (5, 6)**, but compressed representation hurts fine-grained or underrepresented lesions (0, 2, 3, 4). That's expected: ResNet18 has limited mid-level capacity, so Attention Transfer can't fully align the deeper discriminative filters of the ResNet50.

---

## ❖ Confusion Matrix insights

Comparing your raw CMs:

- The teacher's CM shows tighter diagonals and smaller off-class bleed.
- The student's CM confuses classes **2 vs 4** and **4 vs 5** (common in HAM10000 — benign vs malignant overlaps).
- Class 3 ("VASC" in HAM10000 taxonomy) is underrepresented and poorly separated, both models struggle but the teacher still manages better recall.

This suggests adding **class-balanced focal loss or reweighted CE** for the CE term in KD could close a few points of macro-F1 gap.

---

## 💡 Diagnostic takeaways

### 1. Distillation effectiveness:

- KD + AT clearly transferred high-level structure (class 5, 6 stability).
- Missing low-level lesion texture adaptation (classes 0, 2, 4) indicates the AT weight  $\beta$  could be increased or multi-layer weighting introduced.

### 2. Bottleneck:

- Student struggles with the same confusion patterns that the teacher marginally solved — meaning representational bottleneck, not optimization failure.

### 3. Next experimental knobs:

- Increase beta from 1000 → 2000 to amplify AT influence.
- Use alpha=0.4 (more KD emphasis) with T=4.
- Optional: add **label smoothing 0.1** or **focal loss ( $\gamma=1.5$ )** for the CE term.
- Fine-tune 10 epochs starting from the best checkpoint with a 10× smaller LR (2e-5).

---

## Quantitative comparison snippet for your paper / report

The teacher (ResNet50) achieved 89.3% validation accuracy and 0.834 macro-F1 on HAM10000.

The student (ResNet18) distilled via Knowledge Distillation ( $\alpha = 0.5$ ,  $T = 4.0$ ) and Attention Transfer ( $\beta = 1000$ ) attained 82.3% validation accuracy and 0.697 macro-F1. Despite a 55 % parameter reduction, the student preserved 84 % of the teacher's macro-F1, demonstrating effective compression with limited capacity loss on major lesion classes.

---

## Recommended next move

If you want a publication-ready comparative visualization and export:

1. **Generate a bar chart** of per-class F1 (Teacher vs Student).
  2. **Compute relative gap ( $\Delta F1 / F1_{teacher}$ )** for each class.
  3. **Export both models to TorchScript + ONNX** (teacher/student side-by-side) for runtime/size benchmarking.
  4. Optionally run **PTQ (FX) quantization** on the student to demonstrate further edge efficiency.
- 

Would you like me to generate the **comparison report script** that automatically builds:

- a CSV of per-class Teacher vs Student metrics,
- $\Delta F1$  and  $\Delta Recall$ ,
- and a Matplotlib bar chart visualization?

That will give you a clean figure for your paper and a ready numerical summary.

Model	Params	MACs	FLOPs (2×MA C)	TorchScri pt size	Avg laten cy (ms, b=1)	Peak GPU mem (bytes)		Val acc
						p50 (m s)	p90 (m s)	
ResNet18	11,180,1 03	1,824,804,3 59	3.6496 e9	44,887,42 6 B (~42.8 MiB)	2.20 ms	2.0	2.5	65,116,1 60 (~62.1 MiB) <b>0.823</b> 3
MobileNet V2	2,232,83 9	319,027,97 5	6.3806 e8	9,450,712 B (~9.0 MiB)	4.03 ms	3.7	3.9	29,756,4 16 (~28.4 MiB) 0.775 8
EfficientN et-B0	4,016,51 5	408,924,86 3	8.1785 e8	16,803,22 2 B (~16.0 MiB)	5.62 ms	5.6	5.8	36,921,3 44 (~35.2 MiB) 0.796 8

### Quick interpretation (trade-offs)

- **ResNet18** — *best accuracy* (82.3%) and fastest inference by a comfortable margin (2.2 ms). Heavy-ish params and traced size, highest peak memory. Good if you can afford ~44MB model file and ~62MB GPU memory for models + activations.
- **MobileNetV2** — *smallest artifact* (~9 MB traced), lowest memory footprint (~28 MB), slightly slower (~4.0 ms) than ResNet18 here (implementation/kernel differences make MobileNet slower on desktop GPU sometimes). Best candidate if you need small on-disk size and low memory.
- **EfficientNet-B0** — *middle ground* in params/size and best MACs-to-accuracy ratio for some classes — but slowest (~5.6 ms) in your runs. Good compromise if you want reduced params vs ResNet18 with similar accuracy trend.

Important: on small GPUs / edge devices, **params and MACs do not always equal lower latency** — kernel shapes, depthwise convs, and memory/launch overhead matter. Here ResNet18 ended up fastest on the RTX 3050.