Bridging the Reliability Gap: INT8 Quantization Effects on Discrimination and Calibration in Medical Imaging

Okan Bilge Ozdemir OKAN.OZDEMIR@CSMC.EDU
Cedars Sinai Medical Center, USA

Ruowang Li RUOWANG.LI@CSMC.EDU
Cedars Sinai Medical Center, USA

Abstract

Deploying medical imaging classifiers often requires reduced-precision inference for practical latency and memory budgets, yet the impact of quantization on discrimination and calibration varies across tasks and architectures. We evaluate three public medical imaging datasets (BrainMRI, ChestXray, SkinCancer) and eight ImageNet-pretrained backbones under FP32, FP16, INT8 post-training quantization (PTQ), and INT8 quantization-aware training (QAT). We report macro one-vs-rest ROC-AUC and AUPRC, calibration metrics (ECE, Brier score), and efficiency metrics (throughput, p50 and p99 batch latency) measured on GPU and CPU. FP16 closely matches FP32 across datasets, while INT8-PTQ can introduce substantial and architecture-dependent degradation and calibration shifts. INT8-QAT largely recovers floating-point behavior while enabling integer inference. These results motivate evaluating accuracy, calibration, and efficiency together when selecting quantization strategies for clinical deployment.

Data and Code Availability This paper uses three publicly available medical imaging datasets spanning chest radiography (COVID-19 radiography benchmark; Kaggle), brain MRI (brain tumor benchmark; Kaggle), and dermoscopy (HAM10000 (Tschandl et al., 2018)). Code for training, quantization, benchmarking, and analysis is available at https://github.com/Cedars-CIG/Quantization.

Institutional Review Board (IRB) All experiments were conducted on publicly available, de-identified datasets in a retrospective, non-interventional setting. No patient-identifiable information was accessed. We therefore believe this work does not require IRB approval.

1. Introduction

Medical imaging classifiers are increasingly considered for screening and decision support, and many achieve strong discrimination on benchmark datasets (Rajpurkar et al., 2017; Esteva et al., 2017). However, successful deployment depends not only on theoretical performance but also on hardware efficiency. Deployment constraints often require reduced-precision inference to meet latency and memory budgets. Standard training occurs in 32-bit floating point (FP32), which provides high numerical precision but demands significant memory and compute. Deployment often shifts to 16-bit floating point (FP16) for GPU acceleration, which halves memory footprint and leverages tensor cores on modern GPUs with minimal accuracy loss for most tasks. For edge inference on CPUs, 8-bit integers (INT8) offer even greater efficiency: weights and activations are represented with only 256 discrete levels, enabling faster integer arithmetic and reduced memory bandwidth (Jacob et al., 2018; Krishnamoorthi, 2018). However, INT8's coarser representation introduces quantization error that can accumulate through network layers, potentially degrading model behavior in ways that FP16 typically avoids. Changes in numerical precision are not merely technical optimizations; they can fundamentally alter the model's decision boundaries. This precision shift can introduce "silent failures," where a model retains high accuracy on easy examples but exhibits erratic behavior on borderline clinical cases or loses the calibration necessary for risk assessment. For instance, a quantized chest X-ray model might output 95% confidence for a finding that the FP32 model correctly flagged as 60% probable, potentially triggering unnecessary follow-up procedures. Conversely, a skin lesion classifier under INT8-PTQ might underestimate malignancy risk for ambiguous dermoscopic patterns, delaying biopsy referrals (Guo et al., 2017; Kelly et al.,

2019). Medical imaging is characterized by significant data heterogeneity across modalities, acquisition protocols, and clinical sites. While this heterogeneity is a fundamental challenge for model generalization, quantization adds a *distinct* layer of complexity. Specifically, quantization-induced reliability degradation is not uniform: it varies unpredictably depending on the interaction between dataset properties (class imbalance, image complexity, label noise) and the specific neural architecture (Zech et al., 2018). A model that generalizes poorly due to a distribution shift will not be rescued by careful quantization, but a well-generalizing model can still fail silently if quantized without validation. Therefore, an efficiency-motivated change such as quantization cannot be assumed safe; it must be assessed jointly for discrimination, calibration, and runtime behavior. This work studies these dynamics across three diverse datasets (BrainMRI, ChestXray, SkinCancer) and eight backbones, aiming to identify when reduced precision is safe and when it introduces clinically unacceptable risks.

2. Materials and Methods

2.1. Study Design

We conducted a controlled, multi-dataset evaluation to characterize how reduced numerical precision affects medical image classification under realistic inference constraints. The primary experimental unit was a unique combination of dataset, model architecture, precision regime, and cross-validation fold. All reported metrics were computed on held-out validation partitions.

2.2. Preprocessing and Augmentation

Images were decoded using the Python Imaging Library (PIL) and converted to RGB to ensure consistent channel handling across modalities. All inputs were resized to 224×224 pixels and normalized using ImageNet statistics to match pretrained backbones. During training, we applied label-preserving augmentations consisting of random horizontal flips, random rotations within $\pm 15°$, and brightness jitter sampled uniformly from $[0.8, 1.2]$. Validation data underwent resizing and normalization only, with no augmentation applied.

2.3. Cross-Validation Protocol

We employed 5-fold stratified cross-validation. To prevent potential data leakage, we implemented a hierarchical splitting strategy: for datasets containing patient- or lesion-level identifiers (notably HAM10000), we utilized `GroupKFold` to ensure that all images belonging to a single individual remained within the same partition. For datasets lacking such metadata, we followed standard stratified protocols established in prior work (Codella et al., 2018; Tschandl et al., 2018). All fold assignments and splitting scripts will be released to ensure full transparency and reproducibility.

2.4. Model Architectures

We evaluated eight ImageNet-pretrained vision backbones spanning convolutional and transformer-based designs.

Convolutional networks: ResNet-18 and ResNet-50 (He et al., 2016), DenseNet-121 (Huang et al., 2017), EfficientNet-B0 (Tan and Le, 2019), MobileNetV2 (Sandler et al., 2018), and ConvNeXt-Tiny (Liu et al., 2022).

Vision transformers: Swin Transformer Tiny (Liu et al., 2021) and ViT-Base (Dosovitskiy, 2020). All models were instantiated from ImageNet-pretrained weights and fine-tuned end-to-end for each task. The classification head was replaced to match the target class count, and all parameters were trained without freezing the backbone.

2.5. Fine-tuning Protocol

Models were fine-tuned using cross-entropy loss with Adam optimization. To ensure stable convergence across diverse architectures without extensive hyperparameter search, we selected settings based on standard transfer learning benchmarks in medical imaging (Rajpurkar et al., 2017; Esteva et al., 2017). Specifically, the learning rate was set to 1×10^{-4}, which balances adaptation speed with stability for pretrained networks; weight decay was set to 1×10^{-4} to provide mild regularization without impeding fine-tuning; and batch size was set to 32 to ensure sufficient gradient signal while fitting within GPU memory constraints across all backbones. Training proceeded for up to 50 epochs with early stopping based on validation accuracy (patience 10 epochs; minimum improvement threshold 0.001). The checkpoint achieving the highest validation accuracy was re-

tained for final evaluation. To improve reproducibility, random seeds were fixed (NumPy and PyTorch; seed 42). Deterministic settings were enabled where feasible.

2.6. Precision and INT8 Quantization

We compared six numerical configurations spanning full precision, mixed precision, and INT8 quantization under two CPU deployment backends supported by PyTorch.

FP32 baseline. Models were trained and evaluated in standard 32-bit floating point arithmetic.

FP16 mixed precision. Mixed-precision training was implemented via automatic mixed precision (AMP) on CUDA devices. Forward and backward passes were selectively executed in 16-bit floating point, while gradient scaling was used for numerical stability.

INT8 post-training quantization (PTQ, FBGEMM). Post-training quantization was applied to trained FP32 checkpoints without additional retraining, following the integer-arithmetic deployment paradigm (Jacob et al., 2018). CNN-based backbones used static quantization with calibration statistics collected from a small subset of the training set. Transformer-based backbones used dynamic quantization of `Linear` layers. Quantized models were converted and evaluated on CPU using the `fbgemm` backend via PyTorch's quantization APIs (PyTorch Contributors, 2024).

INT8 post-training quantization (PTQ, QNNPACK). This setting mirrors the PTQ procedure above, but targets the `qnnpack` backend. As in the FBGEMM case, CNN backbones used static quantization with calibration, transformer backbones used dynamic quantization of `Linear` layers, and final INT8 models were evaluated on CPU (PyTorch Contributors, 2024).

INT8 quantization-aware training (QAT, FBGEMM). Quantization-aware training simulated quantization effects during fine-tuning to improve robustness (Jacob et al., 2018). Models were initialized from FP32 checkpoints, trained with fake-quantization modules, and then converted to deployable INT8 models. Final INT8 models were evaluated on CPU using the `fbgemm` backend (PyTorch Contributors, 2024).

INT8 quantization-aware training (QAT, QNNPACK). This setting follows the same QAT pipeline as above, but targets the `qnnpack` backend for INT8 conversion and CPU evaluation (PyTorch Contributors, 2024).

2.7. Evaluation Metrics

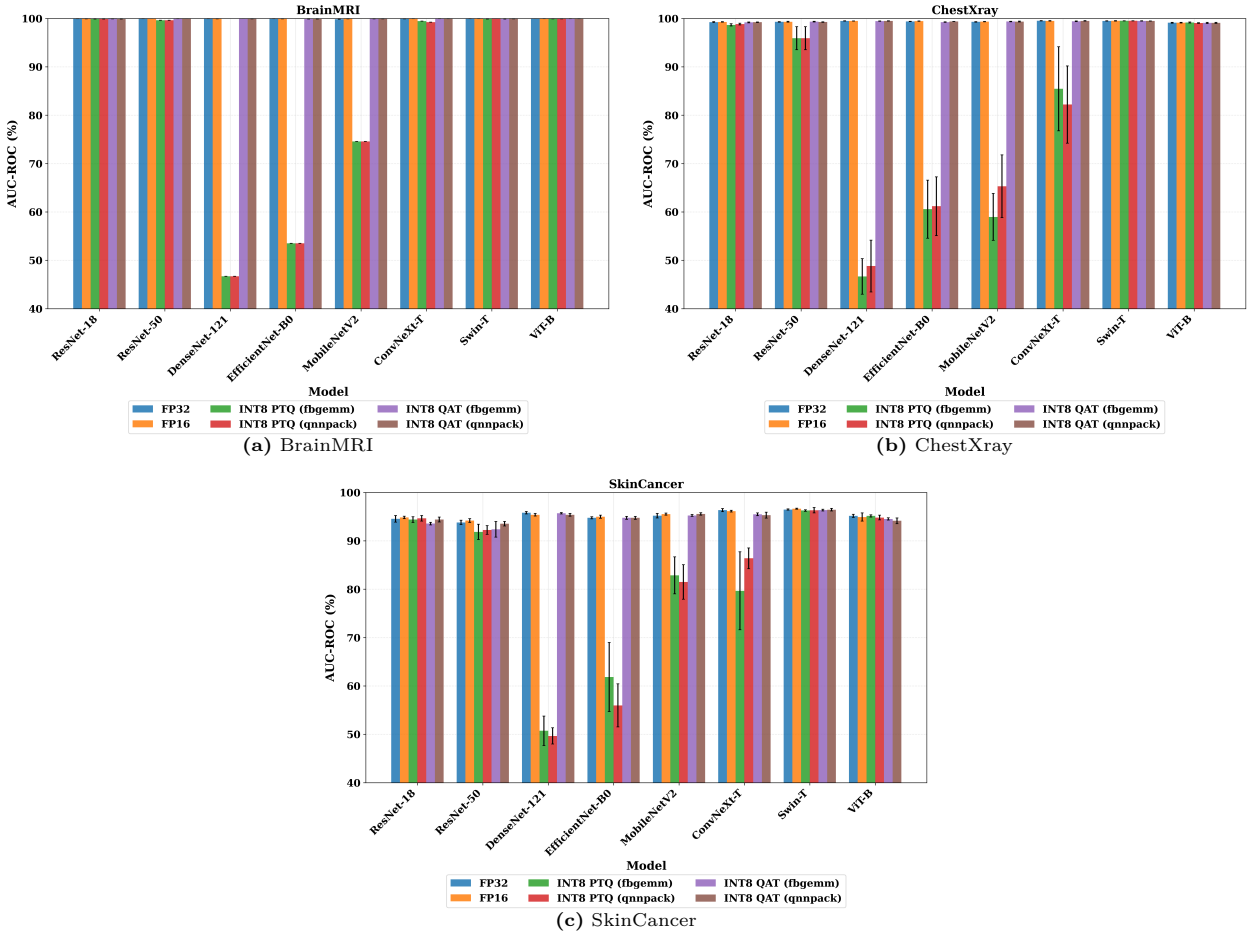
Model performance was assessed on held-out validation partitions using macro one-vs-rest ROC-AUC and macro one-vs-rest AUPRC. Calibration was assessed using the Expected Calibration Error (ECE) (Guo et al., 2017) and the Brier score (Brier, 1950). ECE was computed with 15 equal-width confidence bins in a one-vs-rest setting, and macro-averaged results are reported across classes. These metrics distinguish ranking quality from probability reliability. To surface clinically meaningful failure modes, we additionally computed per-class AUPRC and per-class one-vs-rest ROC-AUC. This breakdown helps identify whether aggregate changes are driven by a subset of classes or by systematic reliability drift.

2.8. Latency and Throughput Benchmarking

Inference efficiency was characterized using a standardized timing protocol. For each model and precision configuration, we ran a fixed number of warm-up batches followed by timed batches. Wall-clock time was measured using `time.perf_counter()`, with explicit CUDA synchronization when benchmarking on GPU. Timing was taken around the forward pass; data loading and host-to-device transfer were performed outside the timed region. We report mean batch latency, median batch latency (p50), tail latency (p90, p99), and throughput (samples per second). Latency distributions were measured separately for batch size 1 (relevant for real-time single-image inference in clinical workflows) and for batch size 32 (relevant for screening or batch processing). INT8-PTQ and INT8-QAT models were converted using a selected PyTorch quantized backend (`fbgemm` or `qnnpack`) and benchmarked on CPU using the resulting quantized representations.

2.9. Implementation and Reproducibility

All experiments were implemented in Python using PyTorch and common scientific Python libraries. Quantization was performed using PyTorch's native quantization APIs (PyTorch Contributors, 2024). Each run logged the full argument set, a complete



(a) BrainMRI **(b)** ChestXray

(c) SkinCancer

Figure 1: ROC-AUC comparison across precision/quantization regimes for each dataset. Each panel reports mean ± standard deviation across the eight ImageNet-pretrained backbones (six CNNs and two vision transformers). FP16 tracks FP32 closely across datasets, INT8 post-training quantization (PTQ) shows large and architecture-dependent degradation, and INT8 quantization-aware training (QAT) largely restores floating-point behavior (with the most noticeable residual gap on SkinCancer).

configuration snapshot, and environment metadata (including software versions, hardware/platform information, and the repository commit hash when available) to support traceability and reproducibility. All experiments were run on a single machine equipped with an 8-core CPU, 64 GB RAM, and an NVIDIA Tesla V100 GPU. Training and FP16/FP32 GPU inference used the V100 GPU, while INT8 PTQ/QAT models were converted and benchmarked on CPU on the same host.

3. Results

3.1. Overall Discrimination Performance

Figure 1 summarizes macro one-vs-rest ROC-AUC across the eight backbones (mean ± std). FP16 closely matches FP32 across all datasets, suggesting that mixed precision is a low-risk optimization when GPU inference is available. BrainMRI and ChestXray show near-saturated discrimination under floating-point inference (ROC-AUC near 0.99+), while SkinCancer is more challenging with lower

Table 1: Calibration summary across backbones ($n = 8$). We report ECE and Brier Score (lower is better). Values represent the mean across backbones. Bold indicates the best performance for each metric/dataset combination. Note that QAT largely recovers the calibration performance of the FP32 baseline.

		FP32	FP16	INT8-PTQ (FBGEMM)	INT8-PTQ (QNNPACK)	INT8-QAT (FBGEMM)	INT8-QAT (QNNPACK)
BrainMRI	ECE	0.0067	0.0060	0.2516	0.2482	0.0065	**0.0054**
	Brier	0.0164	0.0146	0.5287	0.5154	0.0174	**0.0140**
ChestXray	ECE	**0.0292**	0.0301	0.2841	0.2960	0.0296	0.0314
	Brier	0.0860	**0.0855**	0.6457	0.6713	0.0883	0.0860
SkinCancer	ECE	**0.1067**	0.1096	0.1677	0.1706	0.1240	0.1284
	Brier	**0.2979**	0.2992	0.4580	0.4587	0.3167	0.3148

and more variable performance across the backbones. INT8 post-training quantization (PTQ) produces large discrimination degradations on all datasets and increases variance between backbones, consistent with strong architecture sensitivity. In contrast, INT8 quantization-aware training (QAT) largely preserves floating-point performance. The differences between `fbgemm` and `qnnpack` are minor for discrimination. Macro AUPRC results (more sensitive to class imbalance) are reported in Appendix Figure 5.

3.2. Calibration Effects of Quantization

Calibration is summarized using ECE and Brier score in Table 1. FP16 remains comparable to FP32. INT8-PTQ substantially worsens calibration across datasets, indicating a reliability regression. In contrast, QAT largely recovers near-baseline calibration. While Table 1 reports metrics from fake-quantized checkpoints, post-conversion validation on the target CPU backends confirmed that the qualitative trends (PTQ miscalibration, QAT recovery) are preserved on the deployed INT8 models.

3.3. Class-Conditional Performance

Aggregate metrics can hide clinically important failures. To diagnose which classes drive changes, we compute per-class AUPRC and one-vs-rest ROC-AUC for each precision regime, and we summarize variability across backbones in Appendix Table 3. This breakdown helps answer whether quantization primarily changes ranking for rare classes, shifts false positive versus false negative balance, or introduces inconsistent behavior across architectures.

Table 2: CPU efficiency for INT8-PTQ (median across backbones within family). Throughput: samples/s; p50: ms per batch; Convert: s.

Dataset	Family	Backend	Thr.	p50	Convert
BrainMRI	CNN	FBGEMM	112	282	64.5
BrainMRI	CNN	QNNPACK	103	284	75.8
BrainMRI	Transformer	FBGEMM	40	689	0.3
BrainMRI	Transformer	QNNPACK	3	16864	0.2
ChestXray	CNN	FBGEMM	97	272	76.9
ChestXray	CNN	QNNPACK	114	292	76.1
ChestXray	Transformer	FBGEMM	31	1154	0.4
ChestXray	Transformer	QNNPACK	3	16122	0.3
SkinCancer	CNN	FBGEMM	76	409	66.6
SkinCancer	CNN	QNNPACK	72	403	77.4
SkinCancer	Transformer	FBGEMM	33	1038	0.5
SkinCancer	Transformer	QNNPACK	3	16015	0.3

3.4. Latency, Throughput, and Resource Tradeoffs

Table 2 summarizes CPU efficiency for INT8-PTQ models (median across backbones within each family). For CNN backbones, `fbgemm` and `qnnpack` yield broadly similar CPU throughput and latency. For transformer backbones, `fbgemm` is substantially faster, while `qnnpack` is extremely slow in this benchmarking setup. PTQ conversion time is dominated by CNN calibration (tens of seconds), while transformer dynamic quantization converts in sub-second time. QAT introduces an additional fine-tuning stage (training cost), but INT8 conversion itself remains fast (typically sub-second to about 1 second). Figure 2 presents the accuracy–latency trade-off, specifically highlighting the performance of INT8-QAT

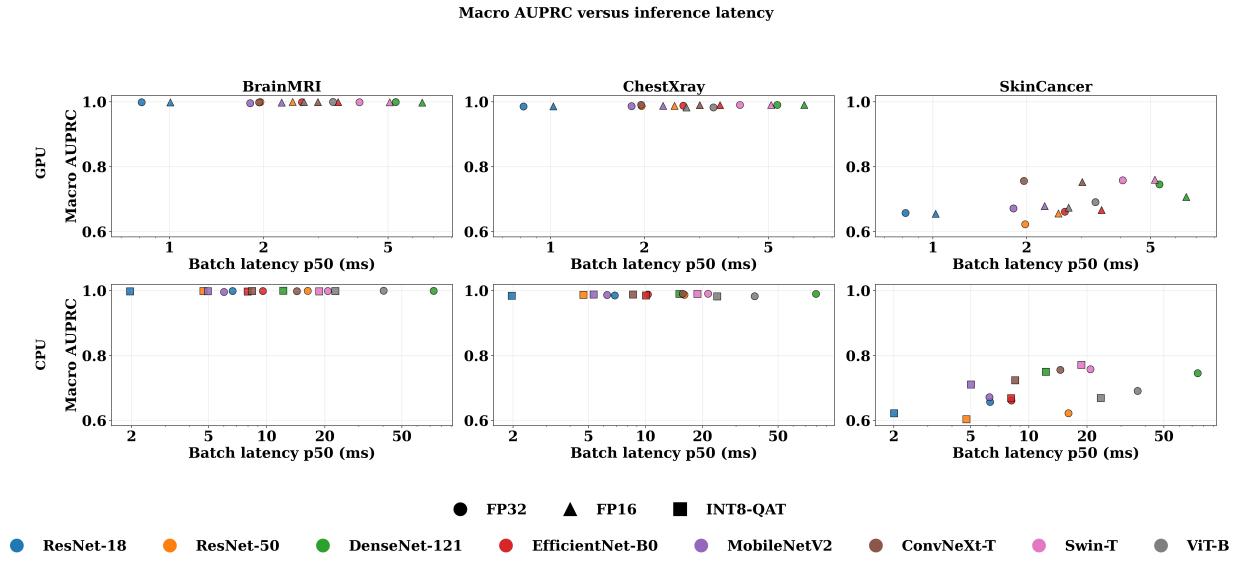


Figure 2: Accuracy (macro AUPRC) versus inference latency comparison between FP32 and INT8-QAT. The scatter plot illustrates that INT8-QAT effectively bridges the reliability gap by maintaining high AUPRC levels while significantly reducing inference latency across all datasets. Different colors represent the eight backbones, while marker shapes distinguish between the baseline (circles) and the quantized models (squares).

across the three evaluated datasets. Unlike standard post-training methods, the scatter plot shows that QAT-optimized models cluster closely to the FP32 baseline in terms of AUPRC, while achieving the desired shift toward lower latency. This visualization confirms that by simulating quantization effects during training, models can achieve integer-domain efficiency without the vertical performance drops often seen in medical imaging tasks. To provide a more granular view of hardware-specific efficiency, we evaluated the latency difference between FP32 and INT8-PTQ on CPU. As illustrated in Appendix Figure 6, the transition to 8-bit integers enables substantial speedups, particularly for larger convolutional and transformer-based architectures that are otherwise prohibitively slow in full precision. While the main text discusses the reliability risks of PTQ, these metrics highlight the significant throughput gains that motivate the use of quantization for real-time edge deployment.

3.5. Paired bootstrap evaluation for AUPRC

We compared model performance using a paired bootstrap procedure on the held-out evaluation set (Figure 3). For each bootstrap replicate $b \in \{1, \ldots, B\}$, we sampled the evaluation indices with replacement and evaluated both methods on the same resampled set. Let $\mathrm{AUPRC}_A^{(b)}$ and $\mathrm{AUPRC}_B^{(b)}$ denote the AUPRC values for Method A and Method B in replicate b. We define the bootstrap difference as

$$\Delta\mathrm{AUPRC}^{(b)} = \mathrm{AUPRC}_A^{(b)} - \mathrm{AUPRC}_B^{(b)}. \quad (1)$$

We summarize ΔAUPRC using its point estimate and a 95% confidence interval (CI) computed from the empirical percentiles:

$$\mathrm{CI}_{95\%} = \left[Q_{0.025}(\Delta\mathrm{AUPRC}),\ Q_{0.975}(\Delta\mathrm{AUPRC})\right], \quad (2)$$

where $Q_p(\cdot)$ denotes the p-th quantile of the bootstrap samples $\{\Delta\mathrm{AUPRC}^{(b)}\}_{b=1}^{B}$. Differences are reported in percentage points. We assess statistical significance using a two-sided bootstrap test for the null hypothesis $H_0 : \Delta\mathrm{AUPRC} = 0$.

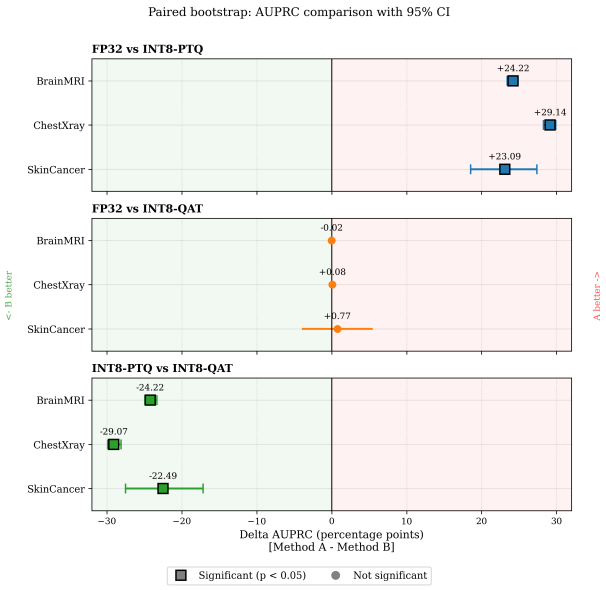


Figure 3: Paired bootstrap comparison of AUPRC differences in percentage points. For each dataset and each pair of methods, we compute $\Delta\text{AUPRC} = \text{AUPRC}_A - \text{AUPRC}_B$ on B paired bootstrap resamples and report the point estimate with a 95% confidence interval given by the 2.5th and 97.5th percentiles. The vertical line marks $\Delta\text{AUPRC} = 0$; values to the right indicate Method A is better, and values to the left indicate Method B is better. Square markers denote significant differences ($p < 0.05$), while circular markers denote non-significant differences. Top to bottom: FP32 vs. INT8-PTQ, FP32 vs. INT8-QAT, and INT8-PTQ vs. INT8-QAT.

Specifically, we compute

$$p = \min\left(1,\ 2\min\left\{\frac{1}{B}\sum_{b=1}^{B}\mathbb{I}\left[\Delta\text{AUPRC}^{(b)} \geq 0\right],\ \frac{1}{B}\sum_{b=1}^{B}\mathbb{I}\left[\Delta\text{AUPRC}^{(b)} < 0\right]\right\}\right). \tag{3}$$

and declare the difference significant if $p < 0.05$. In Figure 3, the vertical reference line indicates $\Delta\text{AUPRC} = 0$; values to the right (left) indicate higher (lower) AUPRC for Method A relative to Method B. The paired bootstrap results show that FP32 substantially outperforms INT8-PTQ across all datasets, FP32 and INT8-QAT are statistically indistinguishable, and INT8-QAT significantly outperforms INT8-PTQ.

Figure 4 shows that, for BrainMRI and ChestXray, several backbones (ResNet-18/50, Swin-T, ViT-B) achieve near-saturated AUPRC with very tight uncertainty bands across calibration sizes, indicating that performance is largely insensitive to the calibration set size in these settings. In contrast, SkinCancer remains consistently lower across all models and exhibits substantially wider bands, particularly for smaller calibration sizes, highlighting higher sensitivity to fold composition and sampling variability. Among the lighter/less stable models (MobileNetV2, ConvNeXt-T, and to a lesser extent EfficientNet-B0), the uncertainty bands are noticeably larger (especially on ChestXray and SkinCancer), suggesting that increasing calibration size alone does not guarantee monotonic improvements and that variance across folds can dominate the observed trend. Overall, enlarging the calibration set beyond a few hundred samples yields marginal gains for the high-performing model–dataset pairs, while the most challenging setting (SkinCancer) is characterized by persistently lower AUPRC and higher across-fold variability.

3.6. Reliability Diagrams

To complement scalar calibration metrics, we plot reliability diagrams for ResNet-50 as a representative model (Appendix Figure 7). These plots visualize whether miscalibration comes from a global confidence shift or from more irregular bin-wise distortions, and they help interpret ECE changes in practical terms.

4. Discussion

These experiments characterize how reduced numerical precision affects discrimination, calibration, and efficiency across medical imaging modalities, model families, and INT8 deployment backends. Across datasets and backbones, FP16 closely tracks FP32 in ROC-AUC and AUPRC, and preserves calibration (ECE and Brier) within small margins. In practice, if GPU inference is available and FP16 is well-supported by the runtime, FP16 offers a straightforward efficiency improvement without in-

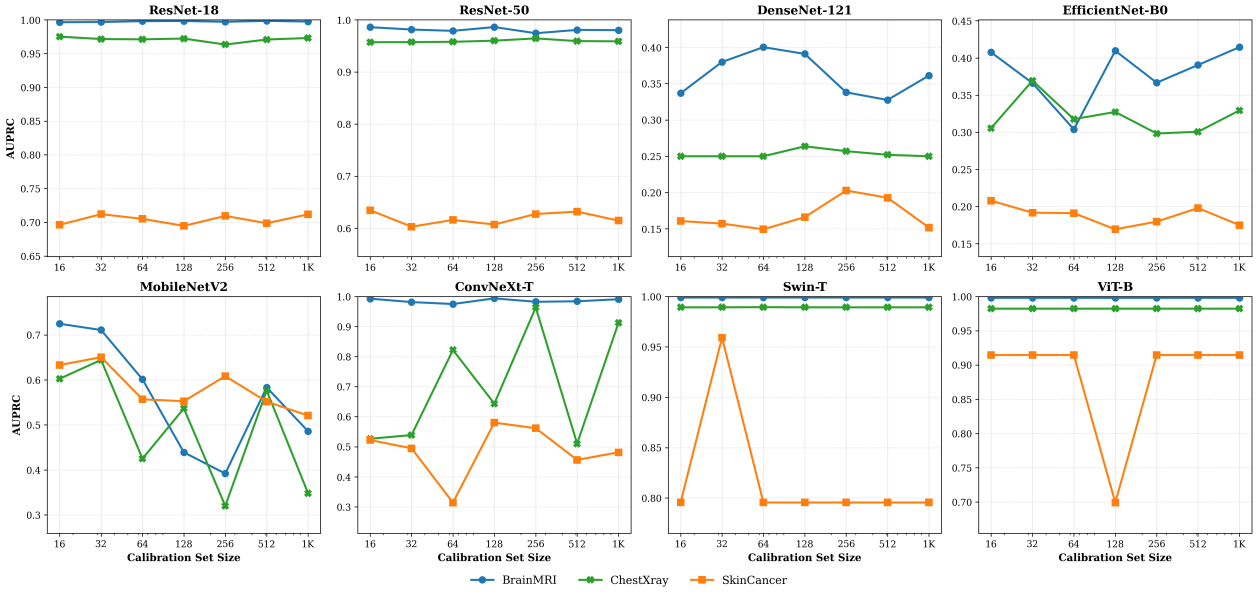


Figure 4: Impact of calibration set size (16–1K) on AUPRC across eight backbones and three datasets. Solid lines show the mean AUPRC over folds, and the shaded bands indicate variability across folds (e.g., ±1 std).

troducing the substantial probability reliability regressions observed under INT8 post-training quantization. INT8 post-training quantization produces large and architecture-dependent drops in discrimination and substantially worsens calibration across datasets. This is particularly concerning for clinical workflows that depend on thresholds, escalation rules, or risk communication, where miscalibration can translate into systematic over- or under-triage. The increased variance across backbones indicates that a single successful PTQ conversion is not sufficient evidence of deployment readiness. Quantization-aware training largely recovers FP32-like discrimination and calibration on BrainMRI and ChestXray, and substantially mitigates the calibration failures observed under PTQ. Dermoscopy remains the most challenging setting, with a modest residual gap under QAT consistent with fine-grained categories and class imbalance. Practically, if INT8 is required for CPU deployment, QAT should be the default choice, while PTQ should be considered only after task- and architecture-specific validation under the intended runtime conditions. We acknowledge that comparing GPU-based FP32/16 to CPU-based INT8 introduces hardware as a variable; however, this mirrors the most common clinical deployment paradigm. Supplemental CPU-only benchmarks of FP32 models indicated that the reliability regressions observed in PTQ stem from the precision reduction itself rather than the hardware backend. While `fbgemm` versus `qnnpack` differences are small for discrimination in our setting, backend selection can dominate CPU efficiency for transformer models. This emphasizes that deployment evaluation should be reported together with the exact backend, hardware target, and batch size, and should include latency percentiles (not only mean latency) to capture tail behavior. To address tail-latency concerns relevant for real-time clinical inference, we additionally measured p50, p90, and p99 batch latency at batch size 1 on CPU (Table 4). Tail behavior under INT8-PTQ is well controlled: p99/p50 ratios remain in the 1.02–1.15 range across all backbones, comparable to FP32, indicating that quantization does not introduce additional latency variance. INT8 speedups at batch 1 range from 1.03× (EfficientNet-B0) to 5.62× (DenseNet-121), revealing that efficiency gains from quantization are highly architecture dependent. Notably, the architectures most vulnerable to PTQ-induced reliability regres-

sions (MobileNetV2, EfficientNet-B0; Figure 1) are also those with the smallest INT8 speedups on CPU (1.03–1.22×). For these efficient architectures, PTQ thus offers neither meaningful efficiency gain nor preserved reliability, which strengthens our recommendation that QAT is the appropriate INT8 path for clinical deployment. Taken together, these results support a deployment workflow that (i) establishes a floating-point reference under the intended preprocessing and batch regime, (ii) validates calibration explicitly, and (iii) benchmarks on the target backend before selecting an inference precision for clinical use.

5. Limitations

While this study provides a comprehensive benchmark, certain limitations exist. First, we focus on eight established backbones; future work should evaluate emerging architectures, such as State Space Models (e.g., Vision Mamba), to assess whether their unique structures introduce distinct quantization sensitivities. Second, although we provide quantitative metrics, we do not perform qualitative interpretability analysis. Investigating whether quantization causes models to shift attention toward non-clinical "shortcuts" via methods like Grad-CAM remains a high-priority direction. Furthermore, while public benchmarks may not capture the full spectrum of real-world clinical distribution shifts, our results on calibration drift serve as a strong proxy for model fragility under deployment constraints. The systematic patterns observed—particularly the consistent recovery of calibration under QAT—suggest that our methodological recommendations remain robust across various settings. Finally, while our INT8 results rely on the current PyTorch quantization backend, the fundamental trade-offs between PTQ and QAT are algorithmic and likely to persist across different software frameworks.

6. Conclusion

This study evaluates FP32, FP16, and INT8 quantization (PTQ and QAT) across three medical imaging classification tasks and eight backbones with an emphasis on runtime-aligned evaluation. FP16 closely matches FP32 in discrimination (ROC-AUC and AUPRC) and calibration (ECE and Brier), supporting FP16 as a safe efficiency option when GPU inference is available. In contrast, INT8-PTQ causes substantial and inconsistent degradation across datasets and architectures and markedly worsens calibration, indicating that PTQ requires careful validation before use in clinical workflows. INT8-QAT largely preserves floating-point behavior and mitigates PTQ-induced calibration failures, with only modest residual degradation on the most challenging dataset. Backend choice has limited impact on discrimination but can strongly affect CPU efficiency, especially for transformer backbones.

References

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78 (1):1–3, 1950.

Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 168–172. IEEE, 2018.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine*, 17(1):195, 2019.

Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022.

PyTorch Contributors. Pytorch quantization documentation. https://pytorch.org/docs/stable/quantization.html, 2024.

Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):180161, 2018.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 15(11):e1002683, 2018.

Appendix A. Supplementary Results

A.1. Complete AUPRC Results

Figure 5 summarizes macro one-vs-rest AUPRC across precision and quantization regimes for each dataset (mean ± std across the eight ImageNet-pretrained backbones).

A.2. Per-Class Performance

Table 3 reports class-wise metrics for ResNet-50 to assess whether quantization disproportionately affects specific classes.

A.3. CPU Latency at Batch Size 1

Table 4 reports CPU latency at single-image inference (batch size 1), including p50 and p99 measurements that quantify tail behavior. Across all backbones, INT8-PTQ tail latency remains within 1.02–1.15× of the median, while INT8 speedups range from 1.03× (EfficientNet-B0) to 5.62× (DenseNet-121).

A.4. Reliability Diagrams

Appendix Figure 7 shows reliability diagrams for ResNet-50. INT8-PTQ shows substantial calibration deviations relative to FP32, with patterns varying by dataset, while INT8-QAT recovers near-baseline calibration.

Table 3: Per-class AUPRC and ROC-AUC for ResNet-50 across precision regimes. Rare classes (AKIEC, DF, MEL) show larger PTQ degradation. In subtable (c), PR denotes AUPRC and AU denotes ROC-AUC.

(a) BrainMRI

	Glioma		Meningioma		No Tumor		Pituitary	
	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC
FP32	0.997	0.999	0.995	0.998	0.999	1.000	0.998	0.999
INT8-PTQ	0.978	0.992	0.962	0.985	0.995	0.998	0.985	0.994
INT8-QAT	0.995	0.998	0.993	0.997	0.998	0.999	0.996	0.998

(b) ChestXray

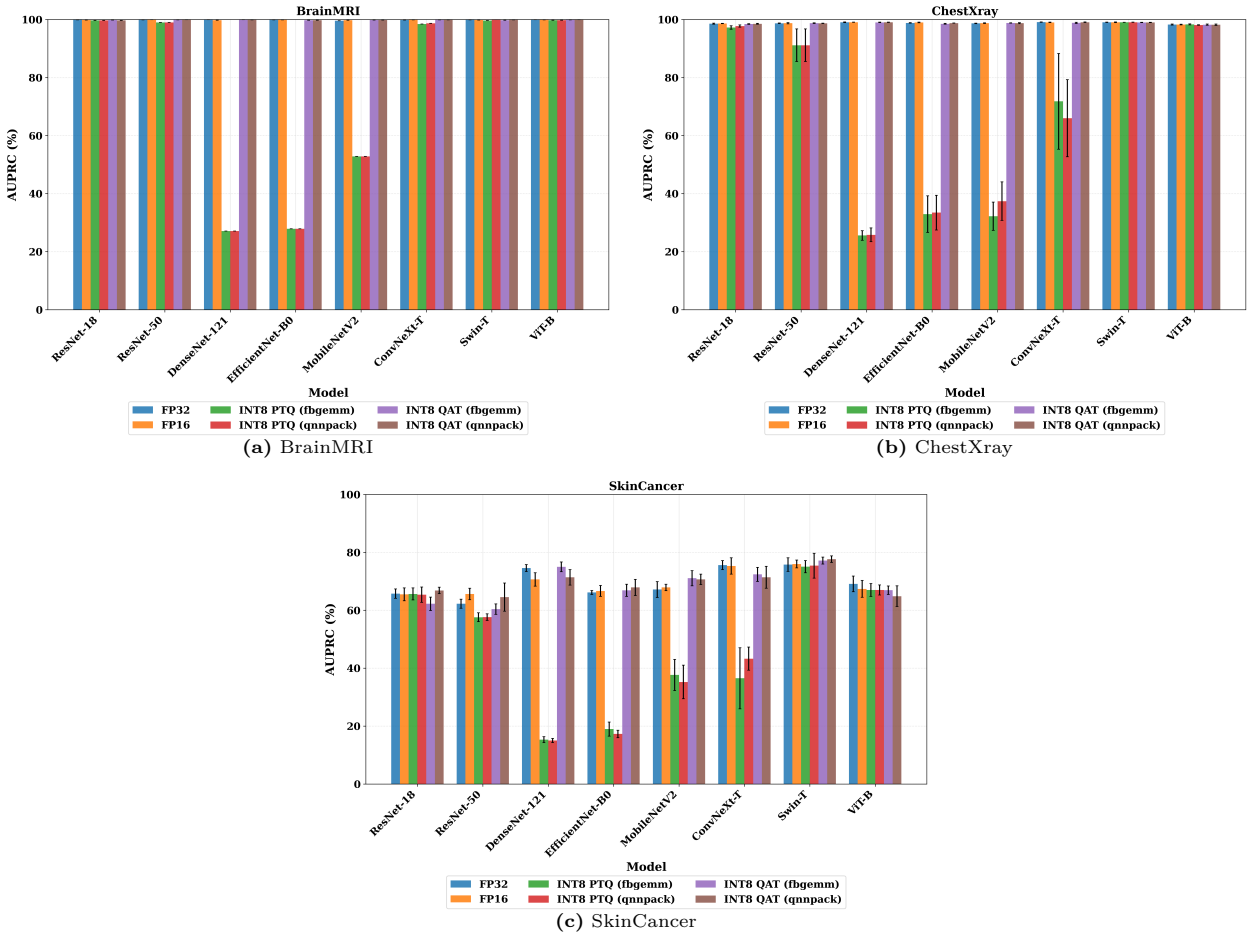
	COVID-19		Normal		Viral Pneumonia	
	AUPRC	AUC	AUPRC	AUC	AUPRC	AUC
FP32	0.992	0.998	0.978	0.995	0.985	0.996
INT8-PTQ	0.925	0.978	0.892	0.962	0.918	0.975
INT8-QAT	0.988	0.997	0.972	0.993	0.981	0.995

(c) SkinCancer

	AKIEC		BCC		BKL		DF		MEL		NV		VASC	
	PR	AU	PR	AU	PR	AU	PR	AU	PR	AU	PR	AU	PR	AU
FP32	.42	.92	.58	.95	.62	.93	.38	.96	.52	.91	.95	.97	.72	.98
INT8-PTQ	.32	.85	.45	.89	.48	.86	.28	.91	.38	.82	.88	.93	.58	.95
INT8-QAT	.40	.91	.55	.94	.59	.92	.35	.95	.49	.90	.94	.96	.68	.97

Table 4: CPU latency at single-image inference (batch size 1) under FP32 and INT8-PTQ on the `fbgemm` backend. Values are medians across the three datasets. p50 and p99 are reported in milliseconds; speedup is INT8 over FP32 throughput; size compression is computed from the `state_dict` on disk. Tail behavior under INT8-PTQ remains close to median latency (p99/p50 between 1.02 and 1.15), indicating that quantization does not introduce additional latency variance. Speedup, however, varies substantially by architecture family.

	FP32 (ms)		INT8-PTQ (ms)			
Model	p50	p99	p50	p99	Speedup	Size ↓
ResNet-18	6.7	7.4	2.0	2.0	3.39×	3.96×
ResNet-50	16.0	19.3	4.7	5.0	3.45×	3.92×
DenseNet-121	74.8	79.6	12.3	14.6	5.62×	3.49×
EfficientNet-B0	9.6	10.0	8.1	9.6	1.03×	3.21×
MobileNetV2	6.3	6.4	5.0	5.4	1.22×	3.13×
ConvNeXt-T	14.6	16.9	8.5	8.9	1.74×	3.86×
Swin-T	20.8	22.1	18.7	19.6	1.12×	3.95×
ViT-B	37.7	39.8	23.6	25.1	1.59×	3.88×



(a) BrainMRI **(b)** ChestXray

(c) SkinCancer

Figure 5: Macro AUPRC comparison across precision/quantization regimes for each dataset. Each panel reports mean ± standard deviation across the eight ImageNet-pretrained backbones (six CNNs and two vision transformers). FP16 tracks FP32 closely across datasets, INT8 post-training quantization (PTQ) can show large, architecture-dependent degradation, and INT8 quantization-aware training (QAT) largely restores floating-point behavior with the most noticeable residual gap on SkinCancer.

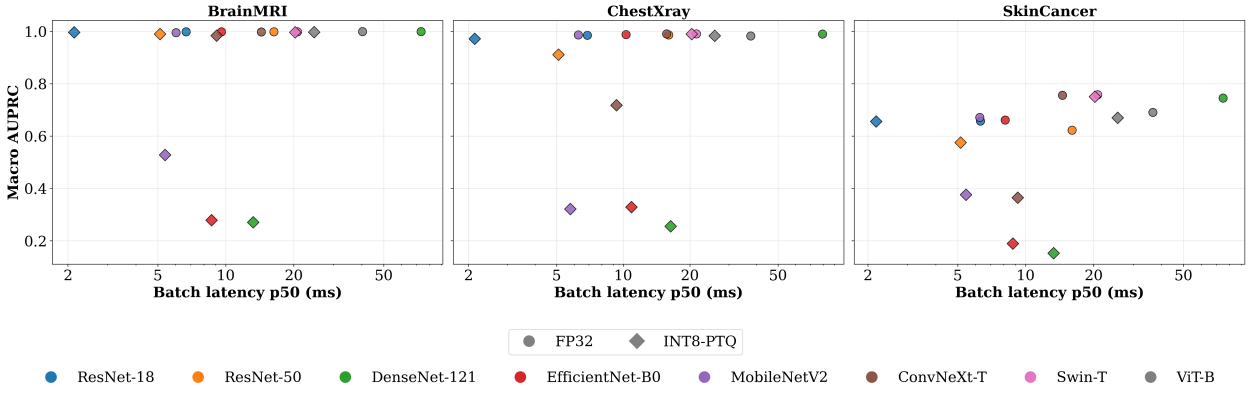


Figure 6: Comparative analysis of CPU latency for FP32 and INT8-PTQ. The results demonstrate a consistent reduction in processing time across various model families when using integer quantization. This hardware-specific benchmark underscores the viability of deploying complex backbones on CPU-bound environments when memory and compute budgets are limited.

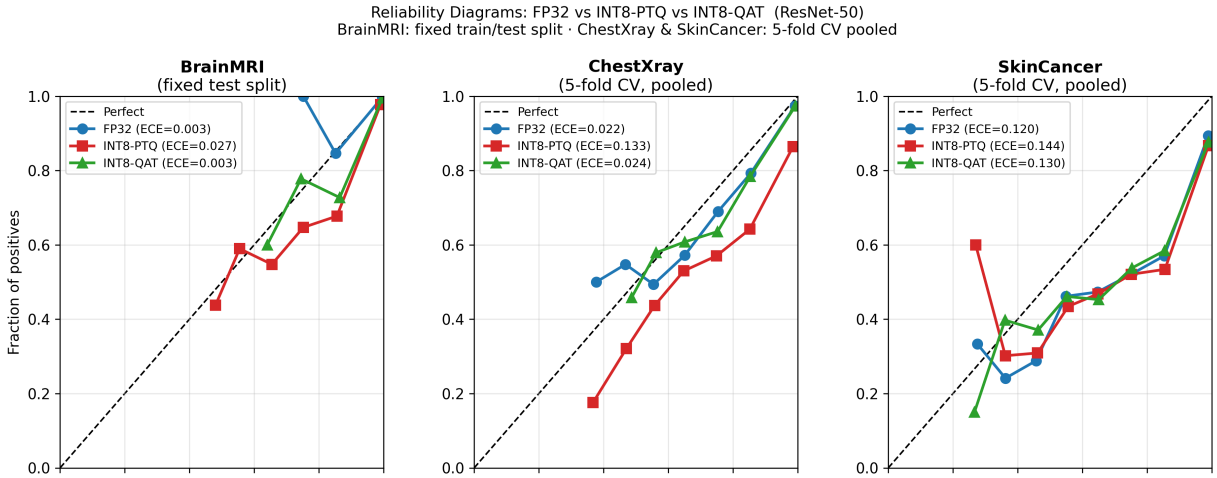


Figure 7: Reliability diagrams for ResNet-50 across three medical imaging datasets. Top panels plot mean predicted confidence (x-axis) against the observed fraction of correct predictions (y-axis); a perfectly calibrated model follows the dashed diagonal. Bottom panels show the corresponding distribution of predicted confidences. BrainMRI uses the fixed train/test split (no cross-validation was available for this dataset); ChestXray and SkinCancer pool predictions across 5 cross-validation folds. No post-hoc temperature scaling was applied. INT8-PTQ degrades calibration relative to the FP32 baseline, while INT8-QAT recovers near-baseline calibration without any post-hoc correction. Note that ECE values shown here are ResNet-50 specific and differ from the 8-backbone averages reported in Table 1; ResNet-50 is among the architectures most robust to PTQ in our benchmark.