Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 572-589
DOI: https://doi.org/10.51903/jtie.v4i3.492

# A Lightweight Medical Multi-Task Backbone for Cross-Modal Pretraining and Parameter-Efficient Few-Shot Transfer on MedMNIST

Gaotian Mi*[1], Tong Ye[2], Dan Wood[3]
Email: gaotian522@gmail.com
[1]*Biomedical Engineering, Johns Hopkins University, MD, USA*
[2]*Computer Science, Northeastern University, CA, USA*
[3]*Computer Engineering, Dartmouth College, NH, USA*
*Corresponding Author

**Abstract**
*Medical imaging has rapidly adopted pre-trained backbones, yet many transfer-learning pipelines remain expensive to train and difficult to adapt when data, compute, or privacy constraints limit full fine-tuning. We present STMedFM, a lightweight medical multi-task backbone baseline designed for fast prototyping across 2D images and 3D volumes. STMedFM uses modality-specific convolutional stems (2D and 3D) and a shared low-depth encoder, and it supports parameter-efficient transfer via Low-Rank Adaptation (LoRA) and bottleneck adapters. We pretrain STMedFM with supervised multi-task learning on four MedMNIST tasks (PathMNIST, BloodMNIST, DermaMNIST, and OrganMNIST3D) using official train/validation/test splits. We then compare (i) training from scratch, (ii) full fine-tuning from the multi-task checkpoint, and (iii) parameter-efficient fine-tuning (LoRA or adapters) that updates only a small fraction of parameters. Under a fixed compute budget (200 pretraining steps; 120 fine-tuning steps for 2D tasks; 50 steps for the 3D task), multi-task pretraining improved performance on PathMNIST (test accuracy 0.568 → 0.634; macro AUROC 0.886 → 0.914) and preserved most gains under PEFT (LoRA AUROC 0.909; Adapter AUROC 0.913) while training only 4,041–5,225 parameters versus 160,105 for full fine-tuning. For DermaMNIST, pretraining increased macro AUROC from 0.746 (Scratch, weighted) to 0.756 (Pretrain+Full), with similar AUROC under LoRA (0.760) and Adapter (0.763). In contrast, BloodMNIST and OrganMNIST3D showed mixed behavior, including cases where Scratch outperformed pretrained variants, indicating that transfer in this compact shared encoder is task-dependent and budget-sensitive. Calibration results were similarly non-monotonic: methods with better AUROC did not always achieve lower ECE. Overall, our results show that a small cross-modal multi-task model can serve as a practical MedMNIST-scale transfer baseline and that LoRA/adapters offer substantial parameter savings when task alignment is favorable. STMedFM should therefore be viewed as a lightweight supervised multi-task backbone on benchmark-scale tasks rather than a broadly general medical foundation model.*

**Keywords**: *Lightweight Medical Backbone, Multi-Task Learning, Medmnist, Parameter-Efficient Fine-Tuning, Calibration.*

## I. INTRODUCTION

Medical imaging workflows increasingly rely on deep learning systems for screening, diagnosis support, and quantitative measurement (Mahfazza et al., 2025; Melyani et al., 2024; Sholekhah & Noviar, 2025; Willie, 2025). In practice, however, building a reliable model for a new task often requires careful dataset curation, substantial computation, and repeated cycles of fine-tuning and validation. These costs are amplified by the heterogeneity of medical data: imaging modalities (histopathology, dermoscopy, microscopy, CT/MRI), dimensionality (2D versus 3D), acquisition protocols, and patient populations vary substantially. At the same time, privacy constraints and labeling costs limit the ability to share or centrally aggregate data. These conditions motivate models that are both transferable and efficient to adapt (Zhou et al., 2021; Thurston et al., 2025).

*Received on October, 2025; Revised on November, 2025; Accepted on December, 2025; Published on December, 2025*

The "foundation model" paradigm—models pretrained on broad data and adapted to many downstream tasks—has transformed natural language processing and computer vision. In medical imaging, however, many such systems rely on larger backbones and broader pretraining corpora than are practical for rapid prototyping. Our goal in this work is narrower: rather than claiming a general medical foundation model, we study whether a compact supervised multi-task backbone can provide useful transfer under tight compute limits on MedMNIST-scale tasks (Bommasani et al., 2021; Dosovitskiy et al., 2021; He et al., 2022).

Two technical challenges are especially relevant in this setting. First, medical tasks are often data-limited or imbalanced, so a shared representation must reuse features efficiently without overfitting to majority classes. Second, full fine-tuning of all parameters is frequently undesirable when compute, storage, or privacy constraints limit the cost of adaptation. Parameter-efficient fine-tuning (PEFT) methods such as Low-Rank Adaptation (LoRA) and adapter modules offer a practical alternative by updating only small trainable modules while freezing most of the backbone. However, PEFT is not guaranteed to work uniformly across tasks, particularly when the source and target domains differ or when 3D volumes require modality-specific representations (Hu et al., 2022; Houlsby et al., 2019; Han et al., 2024).

In this paper, we design and evaluate STMedFM, a lightweight medical multi-task backbone baseline that supports cross-modal (2D+3D) supervised multi-task pretraining and parameter-efficient downstream transfer. STMedFM is intentionally compact: it uses modality-specific convolutional stems and a low-depth shared encoder, enabling rapid training on commodity hardware. The model is pretrained only on four MedMNIST tasks and is therefore intended as a benchmark-scale transfer study rather than a claim of broad medical generalization. We choose the MedMNIST benchmark suite because it provides standardized biomedical classification tasks with official splits, allowing reproducible comparisons across diverse domains and dimensionalities (Caruana, 1997; Tang et al., 2023).

Our contributions are as follows. (1) We propose STMedFM, a compact architecture with modality-specific stems and a shared encoder designed for cross-modal multi-task learning. (2) We perform supervised multi-task pretraining on four MedMNIST tasks and evaluate transfer under three adaptation strategies: full fine-tuning, LoRA, and adapters. (3) We report task-by-task results including accuracy, macro-F1, macro AUROC, calibration (ECE and Brier score), and trainable parameter counts, explicitly highlighting both gains and regressions. (4) We evaluate few-shot transfer with K=1/5/10 samples per class on both a 2D task (DermaMNIST) and a 3D task (OrganMNIST3D). (5) We include ablations on class-weighted loss and on the effect of

excluding the 3D task during pretraining to diagnose cross-modal transfer under tight compute budgets (Hu et al., 2022; Houlsby et al., 2019; Guo et al., 2017; Brier, 1950).

Although our experiments are conducted under a fixed compute budget rather than with full-convergence training, this design choice aligns with the lightweight prototyping motivation. At the same time, the resulting conclusions are budget-dependent and should be interpreted as comparisons under limited optimization, not as full-convergence rankings. The paper therefore serves as an empirical benchmark for compact transfer on MedMNIST-scale tasks rather than as a claim about broad behavior of foundation-models in medicine (Tang et al., 2023).

## II. LITERATURE REVIEW

### *A. Benchmarks and lightweight medical datasets*

Standardized benchmarks play a central role in reproducible medical imaging research. MedMNIST provides small (28×28) 2D and 3D biomedical classification datasets with predefined train/validation/test splits, enabling fast experimentation and consistent evaluation across tasks. MedMNIST v2 has therefore become a useful benchmark for rapid prototyping, and MedMNIST+ further expands this setting to broader benchmark-scale experiments (Tang et al., 2023; Tang et al., 2024).

### *B. Foundation models and transferable visual pretraining*

In general computer vision, large-scale pretraining has enabled strong transfer performance. Vision Transformers, masked image modeling, and vision-language pretraining show that pretrained representations can improve downstream performance and reduce annotation requirements. Promptable segmentation models further illustrate how broadly pretrained visual backbones can support adaptation across tasks (Dosovitskiy et al., 2021; He et al., 2022; Radford et al., 2021; Kirillov et al., 2023).

### *C. Medical pretraining and transferable representations*

In medical imaging, transfer can come from supervised pretraining on curated medical datasets, self-supervised learning on unlabeled scans, and vision-language pretraining using reports. Prior work such as Models Genesis, MedCLIP, and BioViL-T suggests that domain-specific pretraining can improve transferability, while recent surveys in ophthalmology and segmentation-based studies highlight the promise and the limits of medical foundation-style modeling (Zhou et al., 2021; Shen et al., 2023; Vikram et al., 2023; Thurston et al., 2025; Ma et al., 2024).

### *D. Multi-task learning and cross-domain transfer*

Multi-task learning aims to improve generalization by training a shared representation across tasks, often providing implicit regularization and enabling knowledge transfer. In medical contexts, shared low-level structure can be beneficial, but heterogeneous tasks can also compete for capacity and cause negative transfer, especially when modalities differ substantially (Caruana, 1997).

### *E. Parameter-efficient fine-tuning (PEFT)*

PEFT methods address the practical challenge of adapting pretrained models without updating all parameters. Adapter modules insert small bottleneck networks into a frozen backbone, and LoRA learns low-rank updates to weight matrices. These methods reduce memory and training cost, but their effectiveness can depend on task similarity, hyperparameters, and the capacity of the shared backbone (Houlsby et al., 2019; Hu et al., 2022; Han et al., 2024).

### *F. Calibration and reliability*

In medical decision support, calibrated probabilities are often as important as top-1 accuracy. Calibration metrics such as Expected Calibration Error (ECE) and the Brier score quantify how well predicted confidences match empirical correctness. Because modern neural networks can be miscalibrated, calibration analysis is essential when evaluating transferable models intended for medical use (Guo et al., 2017; Brier, 1950).

This literature motivates the central question of our work: can a compact cross-modal multi-task backbone provide useful transfer on MedMNIST-scale tasks, and can PEFT preserve that transfer under strict compute constraints on diverse 2D and 3D medical classification tasks (Bommasani et al., 2021; Tang et al., 2023).

## III. RESEARCH METHOD

### *A. Datasets*

We used four MedMNIST tasks provided as NumPy archives with official splits: PathMNIST (9 classes), BloodMNIST (8 classes), DermaMNIST (7 classes), and OrganMNIST3D (11 classes). All images/volumes have an in-plane resolution of 28×28; OrganMNIST3D provides 28×28×28 voxel volumes. Table 1 summarizes dataset characteristics and split sizes (Tang et al., 2023; Tang et al., 2024).

### *B. Preprocessing and Augmentation*

We normalized the inputs to have approximately zero mean and a unit range by mapping uint8 pixel values to [0,1] and applying a linear transformation to [-1,1]. For training, we applied

random flips (horizontal/vertical for 2D and axis-aligned flips for 3D). Validation and test splits were evaluated without augmentation (Tang et al., 2023).

### *C. Model Architecture (STMedFM)*

STMedFM contains two modality-specific stems: a 2D ConvNet stem for 3-channel images and a 3D ConvNet stem for single-channel volumes. Both stems output a 64-dimensional feature vector via global average pooling. A shared encoder consisting of 2 residual MLP blocks (LayerNorm → Linear(64→128) → GELU → Linear(128→64) → residual) produces the final representation, which is passed to a task head (Linear(64→C)). Figure 1 and Table 2 detail the architecture and indicate where PEFT modules are inserted (Ba et al., 2016; Dosovitskiy et al., 2021).

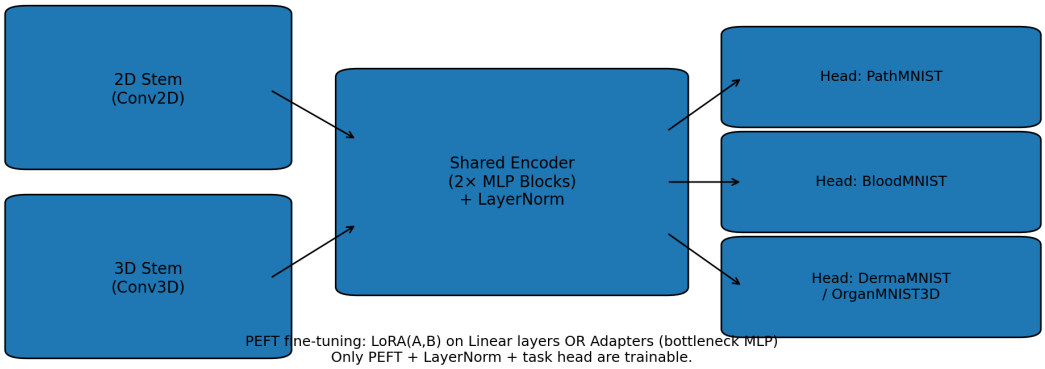


**Figure 1.** STMedFM Architecture Overview and PEFT Insertion Points

**Table 1.** Dataset Characteristics and Split Sizes

| Dataset | Modality | Input shape | Channels | Classes | Train | Val | Test |
|---|---|---|---|---|---|---|---|
| pathmnist | 2D | (28, 28, 3) | 3 | 9 | 89996 | 10004 | 7180 |
| bloodmnist | 2D | (28, 28, 3) | 3 | 8 | 11959 | 1712 | 3421 |
| dermamnist | 2D | (28, 28, 3) | 3 | 7 | 7007 | 1003 | 2005 |
| organmnist3d | 3D | (28, 28, 28) | 1 | 11 | 971 | 161 | 610 |

### *D. Multi-Task Pretraining*

We pretrained STMedFM with supervised multi-task learning across all four tasks. Each training step sampled one task proportionally to its training set size, drew a batch from that task, and updated the shared backbone and that task's head using cross-entropy loss. Under the split sizes in Table 1, this corresponds to approximately 81.9% PathMNIST, 10.9% BloodMNIST, 6.4% DermaMNIST, and 0.9% OrganMNIST3D sampling probability, so OrganMNIST3D is seen only about 1–2 times in expectation over 200 pretraining steps. We retain this simple schedule as a minimal-compute baseline, but it likely disadvantages the 3D task and is important for interpreting later negative-transfer results. To mitigate imbalance within each sampled task, we used inverse-frequency class weights, clipped to [0.1, 5.0], and renormalized them. We ran 200

multi-task steps using AdamW (learning rate 0.002, weight decay 0.0001). Figure 2 shows the pretraining loss curve (Caruana, 1997; Loshchilov & Hutter, 2019).

### *E. Downstream Adaptation Strategies*

For each task, we compared four strategies:

1. Scratch: train the same architecture from random initialization using the task's training split.
2. Pretrain+Full: initialize from the multi-task checkpoint and fine-tune all backbone parameters and the task head.
3. Pretrain+LoRA: insert LoRA modules (rank r=4) on the encoder Linear layers, freeze the backbone weights, and train only LoRA parameters, LayerNorm parameters, and the task head.
4. Pretrain+Adapter: insert bottleneck adapters (bottleneck dimension 16) after the encoder MLP blocks, freeze the backbone weights, and train only adapter parameters, LayerNorm parameters, and the task head (Hu et al., 2022; Houlsby et al., 2019).

### *F. Training Protocol and Compute Budget*

We trained each 2D downstream model for 150 steps from Scratch and 120 steps for fine-tuning (Full/LoRA/Adapter). For OrganMNIST3D (3D), Scratch training used 80 steps and fine-tuning used 50 steps due to higher compute cost. Training used a batch size of 32 for 2D and 8 for 3D, while evaluation used larger batch sizes (256 for 2D; 32 for 3D) to improve efficiency. Table 3 lists all hyperparameters. We fixed random seeds for sampling and initialization to ensure reproducibility (Loshchilov & Hutter, 2019; Kingma & Ba, 2015). Because the study targets lightweight prototyping rather than full-convergence optimization, all comparisons should be interpreted as budget-dependent. Unless otherwise noted, we report descriptive single-run results and do not claim statistical significance.

### *G. Few-Shot Evaluation*

We evaluated few-shot transfer on DermaMNIST (2D) and OrganMNIST3D (3D) by constructing balanced training subsets with K=1, 5, or 10 samples per class drawn from the original training split. For each K, we adapted the pretrained model using Full, LoRA, or Adapter fine-tuning for 40 steps on DermaMNIST and 10 steps on OrganMNIST3D, then evaluated on the full test set. To assess stability, each K-shot subset construction was repeated over three random class-balanced splits. We retain the single-split AUROC results in Tables 9–10 for direct comparison with Figures 4–5, and we additionally report the three-split mean ± standard deviation in Table 10b. This variance analysis is especially important for OrganMNIST3D because the

dataset is small and 3D adaptation uses very few optimization steps (Hu et al., 2022; Houlsby et al., 2019).

### *H. Evaluation Metrics*

We report Accuracy, Macro-F1, and Macro AUROC on the test set. For calibration, we report Expected Calibration Error (ECE) with 15 equal-width confidence bins and the multiclass Brier score computed as the mean squared error between predicted probabilities and one-hot labels. We also provide a reliability diagram and a confusion matrix for the PathMNIST task to illustrate calibration and error patterns (Guo et al., 2017; Brier, 1950). We report raw, uncalibrated probabilities; post hoc temperature scaling was not applied in the present study, so ECE and Brier reflect each method's native calibration. Because each configuration was run once under the fixed compute budget, calibration differences are interpreted descriptively rather than inferentially.

**Table 2.** STMedFM Architecture Specification (Lightweight Cross-Modal Backbone)

| Component | Layers | Output |
|---|---|---|
| 2D Stem | Conv2d 3→32 (3×3,s1) + ReLU; Conv2d 32→64 (3×3,s2)+ReLU; Conv2d 64→64 (3×3,s2)+ReLU; GlobalAvgPool | 64-dim vector |
| 3D Stem | Conv3d 1→16 (3×3×3,s1)+ReLU; Conv3d 16→32 (3×3×3,s2)+ReLU; Conv3d 32→64 (3×3×3,s2)+ReLU; GlobalAvgPool | 64-dim vector |
| Shared Encoder | 2 × [LayerNorm(64) → Linear(64→128) → GELU → Linear(128→64) → Residual] | 64-dim vector |
| PEFT modules | LoRA(r=4) on encoder Linear layers OR Adapter(bottleneck=16) after encoder MLP | Residual update |
| Task head | Linear(64→C) | Logits |

**Table 3.** Training and Fine-Tuning Hyperparameters Used in All Experiments

| Stage | Steps | Train batch (2D/3D) | Optimizer | LR | Weight decay | Class weights | Augmentation |
|---|---|---|---|---|---|---|---|
| Multi-task pretraining | 200 | 32 / 8 | AdamW | 0.0020 | 0.0001 | Inverse frequency (clipped 0.1–5) | Random flips |
| Scratch training (2D) | 150 | 32 / 8 | AdamW | 0.0020 | 0.0001 | Inverse frequency (clipped 0.1–5) | Random flips |
| Scratch training (3D) | 80 | 32 / 8 | AdamW | 0.0020 | 0.0001 | Inverse frequency (clipped 0.1–5) | Random flips |
| Fine-tune Full (2D) | 120 | 32 / 8 | AdamW | 0.0015 | 0.0001 | Inverse frequency (clipped 0.1–5) | Random flips |
| Fine-tune PEFT (2D) | 120 | 32 / 8 | AdamW | 0.0020 | 0.0001 | Inverse frequency (clipped 0.1–5) | Random flips |
| Fine-tune (3D) | 50 | 32 / 8 | AdamW | 0.0015 (Full) / 0.002 (PEFT) | 0.0001 | Inverse frequency (clipped 0.1–5) | Random flips |

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 4 No. 3 December 2025**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 572-589
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

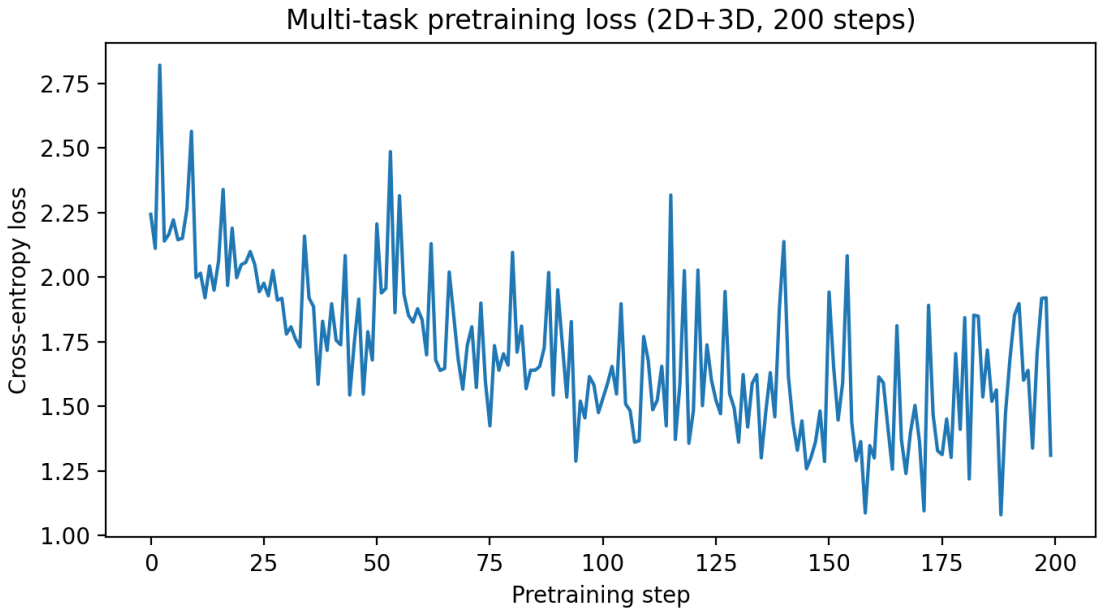


**Figure 2. Multi-Task Pretraining Loss Curve (Supervised)**

## IV. RESULT AND DISCUSSION

### *A. Overall Performance Across Tasks*

Tables 4–7 summarize parameter counts and test-set performance under Scratch, Pretrain+Full, Pretrain+LoRA, and Pretrain+Adapter. The results are task-dependent rather than uniformly positive. PathMNIST benefits most clearly from multi-task pretraining: Pretrain+Full improves accuracy from 0.568 to 0.634 and macro AUROC from 0.886 to 0.914, while LoRA and Adapter retain most of the AUROC gain (0.909 and 0.913) with far fewer trainable parameters. DermaMNIST shows a smaller but still positive transfer signal in AUROC (0.746 for Scratch vs. 0.756 for Pretrain+Full). In contrast, accuracy remains mixed and both PEFT variants slightly underperform Scratch in top-1 accuracy under the weighted-loss setting (Tang et al., 2023).

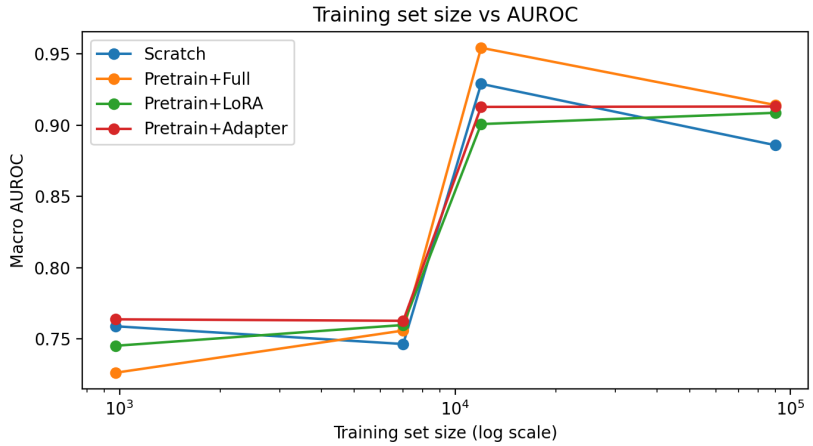


**Figure 3.** Training Set Size Versus Test Macro AUROC Across Methods

BloodMNIST shows that pretraining is most helpful when all parameters are updated: Pretrain+Full achieves the best AUROC (0.954), but Scratch still exceeds Adapter (0.913) and LoRA (0.901), indicating that PEFT does not fully preserve transfer on this task. OrganMNIST3D shows the clearest negative-transfer behavior: Scratch outperforms Pretrain+Full in accuracy (0.205 vs. 0.157), macro-F1 (0.125 vs. 0.088), and AUROC (0.759 vs. 0.726), while Adapter slightly improves AUROC to 0.764 but not accuracy or macro-F1. Taken together, STMedFM is best understood as a lightweight transfer baseline whose benefits depend on task alignment, adaptation strategy, and metric choice, not as a uniformly dominant pretrained model.

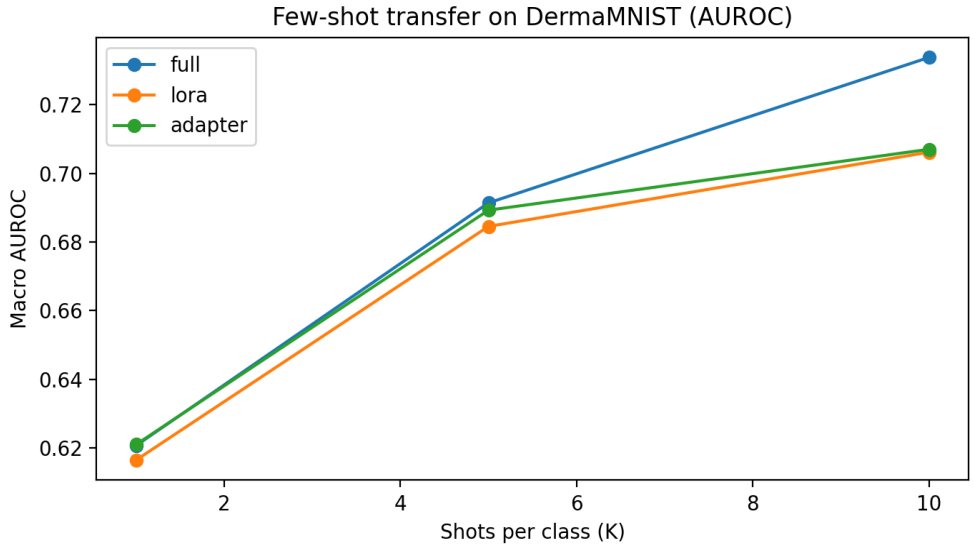


**Figure 4.** Few-Shot AUROC Versus K on Dermamnist (Full vs LoRA vs Adapter)

**Table 4.** Total and Trainable Parameter Counts for Each Adaptation Strategy (Single-Task Heads)

| Task | Method | Total Params | Trainable Params |
|---|---|---|---|
| pathmnist | Scratch | 160105 | 160105 |
| bloodmnist | Scratch | 160040 | 160040 |
| dermamnist | Scratch | 159975 | 159975 |
| organmnist3d | Scratch | 160235 | 160235 |
| pathmnist | Pretrain+Full | 160105 | 160105 |
| pathmnist | Pretrain+LoRA | 163177 | 4041 |
| pathmnist | Pretrain+Adapter | 164361 | 5225 |
| bloodmnist | Pretrain+Full | 160040 | 160040 |
| bloodmnist | Pretrain+LoRA | 163112 | 3976 |
| bloodmnist | Pretrain+Adapter | 164296 | 5160 |
| dermamnist | Pretrain+Full | 159975 | 159975 |
| dermamnist | Pretrain+LoRA | 163047 | 3911 |
| dermamnist | Pretrain+Adapter | 164231 | 5095 |
| organmnist3d | Pretrain+Full | 160235 | 160235 |
| organmnist3d | Pretrain+LoRA | 163307 | 4171 |
| organmnist3d | Pretrain+Adapter | 164491 | 5355 |

### *B. Parameter Efficiency*

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 4 No. 3 December 2025**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 572-589
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

Table 4 shows that LoRA and adapters substantially reduced trainable parameters. On PathMNIST, Full fine-tuning trained 160,105 parameters, while LoRA trained 4,041 parameters and adapters trained 5,225 parameters. Similar reductions held across other tasks. These savings are valuable for privacy-sensitive settings (fewer weights need to be updated or shared) and for rapid iteration (Hu et al., 2022; Houlsby et al., 2019; Han et al., 2024).

### *C. Calibration*

Table 8 reports ECE and Brier score across tasks. Calibration does not correlate monotonically with pretraining or AUROC. For Full fine-tuning, pretraining reduces ECE on DermaMNIST (0.156 → 0.082) and OrganMNIST3D (0.109 → 0.085), but worsens ECE on BloodMNIST (0.053 → 0.061) and PathMNIST (0.081 → 0.116). Brier scores show a partly different pattern, improving for BloodMNIST (0.523 → 0.348), DermaMNIST (0.581 → 0.478), and PathMNIST (0.606 → 0.538), but not for OrganMNIST3D (0.860 → 0.886).

**Table 5.** Test Accuracy Across Datasets and Methods

| Task | Pretrain+Adapter | Pretrain+Full | Pretrain+LoRA | Scratch |
|---|---|---|---|---|
| bloodmnist | 0.594 | 0.730 | 0.519 | 0.585 |
| dermamnist | 0.568 | 0.621 | 0.560 | 0.599 |
| organmnist3d | 0.203 | 0.157 | 0.207 | 0.205 |
| pathmnist | 0.602 | 0.634 | 0.605 | 0.568 |

Thus, better ranking performance does not necessarily imply better probability calibration in this compact setting. Figure 6 is therefore interpreted qualitatively as a diagnostic rather than as evidence of uniform calibration improvement: it shows that confidence and empirical accuracy are not perfectly aligned across bins for PathMNIST under Pretrain+Full, which is consistent with the higher AUROC but worse ECE relative to Scratch. We did not apply post hoc temperature scaling, so the reported values reflect native calibration only. Because each configuration was evaluated once under the fixed budget, these differences should be interpreted descriptively rather than as statistically significant (Guo et al., 2017; Brier, 1950).

### *D. Few-Shot Transfer*

Tables 9–10 and Figures 4–5 show the single-split few-shot AUROC trends, while Table 10b reports mean ± standard deviation over three random class-balanced splits. On DermaMNIST, average AUROC generally increases from approximately 0.54–0.56 at K=1 to approximately 0.66–0.67 at K=10, and the standard deviations are modest, indicating reasonably stable few-shot adaptation in the 2D setting. On OrganMNIST3D, the results are less stable, especially for Full fine-tuning (0.624 ± 0.033, 0.636 ± 0.055, and 0.628 ± 0.044 at K=1/5/10). LoRA and Adapter are somewhat more stable in this setting, but the overall 3D few-shot comparison remains budget-

sensitive. These results support interpreting the OrganMNIST3D few-shot results as trend indicators rather than definitive method rankings (Hu et al., 2022; Houlsby et al., 2019).

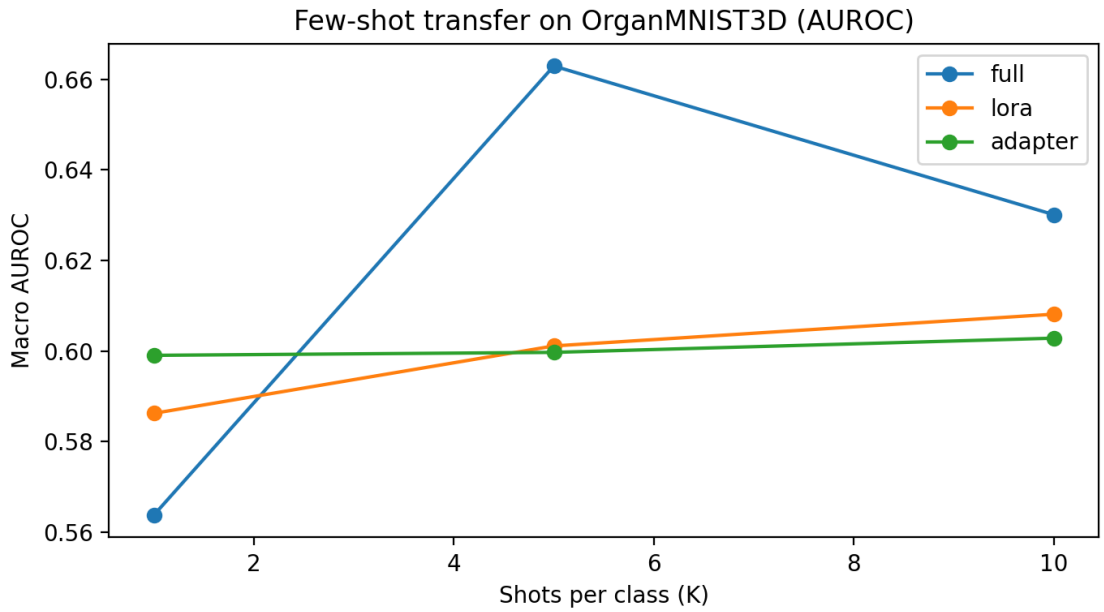


**Figure 5.** Few-Shot AUROC Versus K on OrganMNIST3D (Full vs LoRA vs Adapter)

### *E. Extended-Training Sanity Check*

To test whether the mixed transfer pattern is solely a consequence of the strict step budget, we additionally repeated the OrganMNIST3D comparison with a longer optimization schedule, increasing Scratch training from 80 to 240 steps and Pretrain+Full fine-tuning from 50 to 150 steps. Table 12b reports the longer-budget results; compared with the short-budget results in Tables 5–8, both methods improved, but Scratch remained stronger than Pretrain+Full on OrganMNIST3D (accuracy: 0.352 vs. 0.266; macro-F1: 0.281 vs. 0.196; AUROC: 0.816 vs. 0.794). This suggests that the observed negative transfer is not solely an undertraining artifact and also confirms that the absolute rankings are partly compute-budget-dependent.

**Table 6.** Test Macro-F1 Across Datasets and Methods

| Task | Pretrain+Adapter | Pretrain+Full | Pretrain+LoRA | Scratch |
|---|---|---|---|---|
| bloodmnist | 0.577 | 0.684 | 0.499 | 0.574 |
| dermamnist | 0.263 | 0.182 | 0.206 | 0.167 |
| organmnist3d | 0.109 | 0.088 | 0.091 | 0.125 |
| pathmnist | 0.526 | 0.565 | 0.524 | 0.496 |

**Table 7.** Test Macro AUROC Across Datasets and Methods

| Task | Pretrain+Adapter | Pretrain+Full | Pretrain+LoRA | Scratch |
|---|---|---|---|---|
| bloodmnist | 0.913 | 0.954 | 0.901 | 0.929 |
| dermamnist | 0.763 | 0.756 | 0.760 | 0.746 |
| organmnist3d | 0.764 | 0.726 | 0.745 | 0.759 |
| pathmnist | 0.913 | 0.914 | 0.909 | 0.886 |

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 4 No. 3 December 2025**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 572-589
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

### *F. Error Analysis*

Figure 7 shows a confusion matrix for PathMNIST under Pretrain+Full. Table 13 reports per-class precision/recall/F1, revealing that certain tissue classes are more easily confused, likely due to shared visual texture. This analysis helps identify where additional data or task-specific inductive biases could improve performance (Tang et al., 2023).

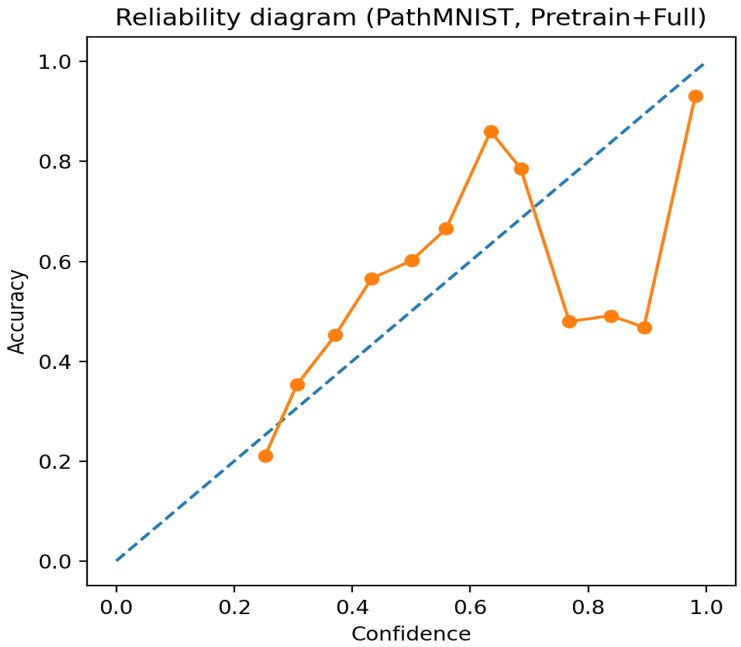


**Figure 6.** Reliability Diagram for PathMNIST Under Pretrain+Full (15 Bins)

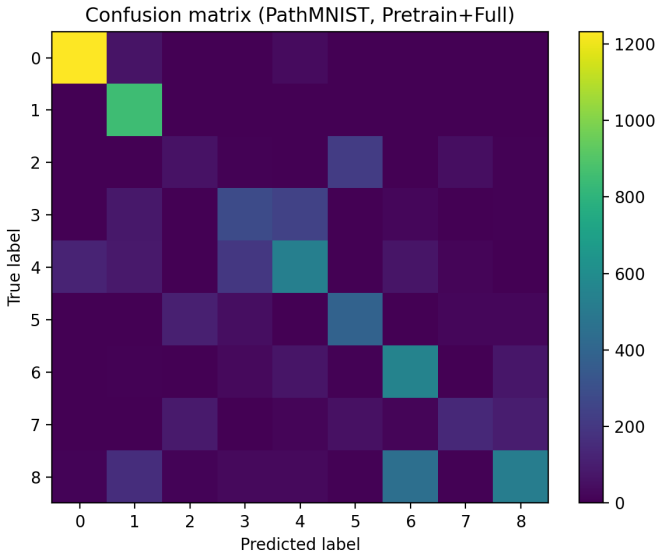


**Figure 7.** Confusion Matrix for PathMNIST Under Pretrain+Full

### *G. Ablations*

Table 11 studies the effect of class-weighted loss on DermaMNIST. Weighted loss generally improves macro-F1 by placing greater emphasis on minority classes, but it can reduce top-1 accuracy and sometimes AUROC; the preferred setting depends on whether the target objective prioritizes class balance or overall accuracy.

**Table 8.** Expected Calibration Error (ECE; Lower is Better) Across Datasets and Methods

| Task | Pretrain+Adapter | Pretrain+Full | Pretrain+LoRA | Scratch |
|---|---|---|---|---|
| bloodmnist | 0.047 | 0.061 | 0.055 | 0.053 |
| dermamnist | 0.115 | 0.082 | 0.105 | 0.156 |
| organmnist3d | 0.076 | 0.085 | 0.076 | 0.109 |
| pathmnist | 0.073 | 0.116 | 0.079 | 0.081 |

**Table 8b**. Multiclass Brier Score (Lower is Better) Across Datasets and Methods

| Task | Pretrain+Adapter | Pretrain+Full | Pretrain+LoRA | Scratch |
|---|---|---|---|---|
| bloodmnist | 0.526 | 0.348 | 0.585 | 0.523 |
| dermamnist | 0.553 | 0.478 | 0.556 | 0.581 |
| organmnist3d | 0.886 | 0.886 | 0.904 | 0.860 |
| pathmnist | 0.547 | 0.538 | 0.535 | 0.606 |

**Table 9.** Few-Shot AUROC on DermaMNIST for K=1/5/10 per Class

| K | adapter | full | lora |
|---|---|---|---|
| 1.000 | 0.621 | 0.621 | 0.617 |
| 5.000 | 0.689 | 0.691 | 0.685 |
| 10.000 | 0.707 | 0.734 | 0.706 |

**Table 10.** Few-Shot AUROC on OrganMNIST3D for K=1/5/10 per Class

| K | adapter | full | lora |
|---|---|---|---|
| 1.000 | 0.599 | 0.564 | 0.586 |
| 5.000 | 0.600 | 0.663 | 0.601 |
| 10.000 | 0.603 | 0.630 | 0.608 |

**Table 10b.** Few-Shot Macro AUROC (Mean ± Std Over Three Random Class-Balanced Splits) on DermaMNIST and OrganMNIST3D

| Dataset | K | Pretrain + Full | Pretrain + LoRA | Pretrain + Adapter |
|---|---|---|---|---|
| DermaMNIST | 1 | 0.535 ± 0.075 | 0.559 ± 0.029 | 0.550 ± 0.039 |
| DermaMNIST | 5 | 0.664 ± 0.043 | 0.639 ± 0.022 | 0.646 ± 0.008 |
| DermaMNIST | 10 | 0.659 ± 0.009 | 0.662 ± 0.008 | 0.669 ± 0.011 |
| OrganMNIST3D | 1 | 0.624 ± 0.033 | 0.669 ± 0.012 | 0.668 ± 0.027 |
| OrganMNIST3D | 5 | 0.636 ± 0.055 | 0.697 ± 0.038 | 0.685 ± 0.025 |
| OrganMNIST3D | 10 | 0.628 ± 0.044 | 0.658 ± 0.009 | 0.667 ± 0.021 |

Table 12 provides a more important finding: cross-modal pretraining does not uniformly help OrganMNIST3D. The most plausible explanation is not a single factor but the interaction of limited shared capacity, modality mismatch, and unbalanced task sampling. First, STMedFM uses a very small shared encoder, so jointly representing 2D histopathology/dermoscopy/microscopy and 3D organ volumes may exceed the shared component's capacity. Second, because pretraining samples tasks in proportion to training-set size, OrganMNIST3D is heavily underrepresented during pretraining. Under this schedule, joint 2D+3D training can dilute modality-specific 3D representation rather than strengthen it.

*Journal of Technology Informatics and Engineering (JTIE)*
**Vol. 4 No. 3 December 2025**
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 572-589
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

Third, Table 12 shows that the effect depends on the adaptation method. With LoRA, 2D+3D pretraining improves AUROC over 2D-only pretraining (0.745 vs. 0.675), but with Adapter, 2D-only pretraining improves accuracy and macro-F1 (0.259/0.166 vs. 0.203/0.109) while slightly lowering AUROC (0.745 vs. 0.764). We therefore interpret Table 12 as evidence that cross-modal sharing is useful only when sampling and capacity are better matched to the 3D task; in the present compact, budget-limited setting, negative transfer on OrganMNIST3D is a real and central finding rather than an anomaly (Caruana, 1997).

**Table 11.** DermaMNIST Ablation: Effect of Class-Weighted Loss (Weights On vs Off)

| Stage | Method | Weights | Accuracy | Macro-F1 | AUROC | ECE | Brier |
|---|---|---|---|---|---|---|---|
| Scratch | Full | On | 0.599 | 0.167 | 0.746 | 0.156 | 0.581 |
| Scratch | Full | Off | 0.653 | 0.208 | 0.806 | 0.103 | 0.471 |
| Finetune | Full | On | 0.621 | 0.182 | 0.756 | 0.082 | 0.478 |
| Finetune | Full | Off | 0.673 | 0.128 | 0.771 | 0.075 | 0.458 |
| Finetune | LoRA | On | 0.560 | 0.206 | 0.760 | 0.105 | 0.556 |
| Finetune | LoRA | Off | 0.670 | 0.152 | 0.763 | 0.047 | 0.460 |
| Finetune | Adapter | On | 0.568 | 0.263 | 0.763 | 0.115 | 0.553 |
| Finetune | Adapter | Off | 0.669 | 0.115 | 0.767 | 0.074 | 0.465 |

**Table 12.** OrganMNIST3D Ablation: 2D+3D Pretraining vs 2D-Only Pretraining

| Pretraining | Finetune | Accuracy | Macro-F1 | AUROC | ECE | Brier |
|---|---|---|---|---|---|---|
| 2D+3D multitask | LoRA | 0.207 | 0.091 | 0.745 | 0.076 | 0.904 |
| 2D-only multitask | LoRA | 0.143 | 0.057 | 0.675 | 0.013 | 0.899 |
| 2D+3D multitask | Adapter | 0.203 | 0.109 | 0.764 | 0.076 | 0.886 |
| 2D-only multitask | Adapter | 0.259 | 0.166 | 0.745 | 0.131 | 0.878 |

**Table 12b.** OrganMNIST3D Extended-Training Sanity Check (Longer Optimization Budget)

| Method | Steps | Accuracy | Macro-F1 | AUROC | ECE | Brier |
|---|---|---|---|---|---|---|
| Scratch | 240 | 0.352 | 0.281 | 0.816 | 0.044 | 0.782 |
| Pretrain + Full | 150 | 0.266 | 0.196 | 0.794 | 0.071 | 0.815 |

**Table 13.** PathMNIST per-Class Precision/Recall/F1 for PRETRAIN+Full

| Class | Support | Precision | Recall | F1 |
|---|---|---|---|---|
| 0.000 | 1338.000 | 0.899 | 0.921 | 0.910 |
| 1.000 | 847.000 | 0.683 | 1.000 | 0.812 |
| 2.000 | 339.000 | 0.221 | 0.177 | 0.196 |
| 3.000 | 634.000 | 0.464 | 0.443 | 0.453 |
| 4.000 | 1035.000 | 0.572 | 0.513 | 0.541 |
| 5.000 | 592.000 | 0.572 | 0.655 | 0.611 |
| 6.000 | 741.000 | 0.492 | 0.741 | 0.592 |
| 7.000 | 421.000 | 0.609 | 0.340 | 0.436 |
| 8.000 | 1233.000 | 0.711 | 0.423 | 0.531 |

## V. CONCLUSION AND RECOMMENDATION

We developed STMedFM, a lightweight medical multi-task backbone baseline that supports cross-modal supervised pretraining and parameter-efficient adaptation with LoRA or adapters. Using four MedMNIST tasks spanning histopathology, blood cell microscopy, dermoscopy, and 3D organ volumes, we evaluated transfer under a fixed compute budget. Multi-task pretraining improved downstream performance most clearly on PathMNIST and more modestly on DermaMNIST, while PEFT preserved much of the PathMNIST AUROC gain with far fewer trainable parameters. At the same time, the results are mixed rather than uniformly positive: BloodMNIST benefits mainly from full fine-tuning, and OrganMNIST3D exhibits clear negative-transfer behavior under several settings. The paper should therefore be read as a benchmark-scale study of budget-limited transfer on MedMNIST, not as evidence of broad behavior in medical foundation models.

The experiments also clarify why negative transfer can arise in this setting. The shared encoder is intentionally small, the pretraining signal is supervised and dataset-limited, and task sampling is heavily dominated by the largest 2D dataset, leaving very little 3D exposure during pretraining. These design choices make PEFT attractive for efficiency gains, but they also limit the extent of cross-modal sharing that can be expected. An additional extended-training sanity check on OrganMNIST3D showed that longer optimization improved both Scratch and Pretrain+Full, but did not reverse the ranking: Scratch remained stronger in both accuracy and AUROC. Calibration results are similarly mixed: better AUROC does not always correspond to better ECE, so post-hoc calibration should be considered before deployment. Because the comparisons are step-limited, our conclusions are budget-dependent and should not be interpreted as full-convergence rankings.

### *Recommendations for practitioners*

First, when rapid adaptation and a low trainable parameter count are critical (e.g., in privacy-preserving or on-device settings), LoRA or adapters provide strong baselines for tasks that align with the pretraining distribution. Second, for heavily imbalanced datasets, class-weighted loss can improve macro-F1, but it should be selected based on the clinical objective (minority-class sensitivity versus overall accuracy). Third, calibration should be evaluated explicitly; if ECE is high, post-hoc calibration (such as temperature scaling) should be applied before using probabilities in decision-making (Han et al., 2024; Guo et al., 2017).

### *Future work*

Several directions can strengthen lightweight medical transfer backbones: (1) incorporating self-supervised or vision-language pretraining to improve generality under limited labels; (2)

improving cross-modal fusion via modality-specific experts or gated routing to reduce negative transfer; (3) scaling the encoder depth modestly while maintaining efficiency; (4) integrating explicit calibration objectives; and (5) testing transfer on higher-resolution and real-world clinical datasets beyond the 28×28 benchmark setting (He et al., 2022; Radford et al., 2021; Shen et al., 2023; Vikram et al., 2023; Kirillov et al., 2023; Ma et al., 2024; Zhou et al., 2021; Thurston et al., 2025).

## REFERENCES

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer Normalization. *arXiv Preprint arXiv:1607.06450*. https://doi.org/10.48550/arxiv.1607.06450

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., … Liang, P. (2021). On The Opportunities And Risks Of Foundation Models. *arXiv Preprint arXiv:2108.07258*. https://doi.org/10.48550/arxiv.2108.07258

Brier, G. W. (1950). Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review, 78*(1), 1–3. https://doi.org/10.1175/1520-0493(1950)078

Caruana, R. (1997). Multitask Learning. *Machine Learning, 28*, 41–75. https://doi.org/10.1023/a:1007379606734

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., … Houlsby, N. (2021). An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arxiv.2010.11929

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. https://doi.org/10.48550/arxiv.1706.04599

Han, X., Zhu, B., Wang, Y., Wu, J., Zhang, R., & Liu, Y. (2024). Parameter-Efficient Fine-Tuning Methods for Pre-Trained Language Models: A Critical Review and Assessment. *arXiv Preprint arXiv:2402.12148*. https://doi.org/10.48550/arxiv.2402.12148

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.48550/arxiv.2111.06377

Houlsby, N., Giurgiu, A., Jastrzębski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-Efficient Transfer Learning For NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. https://doi.org/10.48550/arxiv.1902.00751

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arxiv.2106.09685

Kingma, D. P., & Ba, J. (2015). Adam: A Method For Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arxiv.1412.6980

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., … Girshick, R. (2023). Segment Anything. *arXiv Preprint arXiv:2304.02643*. https://doi.org/10.48550/arxiv.2304.02643

Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arxiv.1711.05101

Ma, J., He, Y., Li, F., Han, L., You, C., & Wang, B. (2024). Segment Anything in Medical Images. *arXiv Preprint arXiv:2304.12306*. https://doi.org/10.48550/arxiv.2304.12306

Mahfazza, E. C., Amrozi, Y., & Muslihul Amin, F. (2025). Enhancing Information Security and Risk Governance in Hospital Electronic Medical Record Systems. *Jurnal Ilmiah Sistem Informasi*, *11*(2), 210–225. https://doi.org/10.51903/00wfhv86

Melyani, M., Prasetyo, T. F., Rahadjeng, I. R., Mufid, Z., Rafik, A., Shaura, R. K., Daniel, D., & Emita, I. (2024). Design Framework of Expert System Program in Otolaryngology Disease Diagnosis Use Extreme Programming (XP) Method (Case Study in THB Bekasi Hospital). *Journal of Technology Informatics and Engineering*, *3*(3), 397–416. https://doi.org/10.51903/jtie.v3i3.209

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., … Sutskever, I. (2021). Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. https://doi.org/10.48550/arxiv.2103.00020

Shen, S., Lin, Z., Gao, K., Wang, B., Tang, X., Liu, X., & Wang, J. (2023). MedCLIP: Medical Knowledge Enhanced Language-Image Pre-Training. *arXiv Preprint arXiv:2301.02228*. https://doi.org/10.48550/arxiv.2301.02228

Sholekhah, D. Z., & Noviar, D. (2025). Integrative Deep Learning Architecture for High-Accuracy Medical Image Segmentation: Combining U-Net, ResNet, and Transformers. *Journal of Technology Informatics and Engineering*, *4*(1), 115–134. https://doi.org/10.51903/jtie.v4i1.288

Tang, Y., Yang, J., Chen, X., Ge, C., Yu, Z., Hong, L., Li, G., & Duan, L. (2023). MedMNIST v2: A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification. *Scientific Data, 10*(1), 41. https://doi.org/10.1038/s41597-022-01721-8

Tang, Y., Yang, J., Chen, X., Ge, C., Yu, Z., Hong, L., Li, G., & Duan, L. (2024). MedMNIST+. *Zenodo*. https://doi.org/10.5281/zenodo.11044450

Tang, Y., Yang, J., Chen, X., Ge, C., Yu, Z., Hong, L., Li, G., & Duan, L. (2024). Rethinking Model Prototyping Through the MedMNIST+ Database. *arXiv Preprint arXiv:2404.15786*. https://doi.org/10.48550/arxiv.2404.15786

Thurston, T. E., et al. (2025). Foundation Models in Ophthalmology. *Ophthalmology Science, 5*(4), 100848. https://doi.org/10.1016/j.xops.2025.100848

Vikram, D. S., Kalaycı, T., Sikonja, C., & Indurkhya, N. (2023). BioViL-T: Learning to Exploit Temporal Structure for Biomedical Vision-Language Processing. *Microsoft Research*. https://www.microsoft.com/en-us/research/publication/learning-to-exploit-temporal-structure-for-biomedical-vision-language-processing/

Willie, M. M. (2025). Value-Based Administration Services and Value-Based Care: Aligning Administrative Efficiency With Patient Outcomes. *Journal of Management and Informatics*, *4*(3), 1032–1042. https://doi.org/10.51903/jmi.v4i3.308