Real-Time Cell Sorting with Scalable In Situ FPGA-Accelerated Deep Learning

Supplementary Material

S1 Benchmark Results

Table S1 shows a comprehensive comparison of classification performance on the LymphoMNIST and MNIST datasets, including the test accuracy achieved by common machine learning classifiers across a range of hyperparameter configurations. Notably, there is a large performance disparity between the two datasets which reflects the increased complexity of LymphoMNIST. For instance, the tree-based methods such as DecisionTreeClassifier performed significantly worse on LymphoMNIST compared to MNIST across the entire hyperparameter search space. GradientBoostingClassifier shows significant improvement in accuracy on MNIST as the model complexity (e.g., number of estimators) increases, while achieving modest gains on LymphoMNIST relative to DecisionTreeClassifier.

Furthermore, KNeighborsClassifier and RandomForestClassifier also achieve superior accuracy on MNIST, while their performance on LymphoMNIST demonstrates their difficulty in capturing complex features. LogisticRegression and LinearSVC also achieve only moderate accuracy on LymphoMNIST, further highlighting the necessity for specialized approaches. This comparative analysis shows the value of LymphoMNIST as a challenging benchmark for advancing ML models and provides a complementary perspective to MNIST's utility in assessing general-purpose algorithmic performance. This exploration provides a foundation for future investigation into improving ML efficacy on datasets like LymphoMNIST.

S2 FPGA Resource-Latency Trade-offs and Scalability Limits

The reuse factor hyperparameter in `hls4ml` controls the degree of resource sharing in the synthesized hardware, directly affecting both latency and resource consumption. This trade-off can be visualized as a Pareto frontier, where design points represent different balance points between parallelization (low reuse factor, low latency, high resources) and resource efficiency (high reuse factor, high latency, low resources). Figure S1 illustrates this trade-off for neural network FPGA implementations of comparable complexity to our Student 2 model. As reuse factor increases from 2 to 1024, latency increases from tens of microseconds to milliseconds while resource consumption (measured as percentage of available LUTs, registers, BRAMs, and DSPs on the KU035 FPGA) decreases dramatically. At reuse factor = 2, DSP utilization approaches 100%, representing the low-latency extreme of the feasible design space. Our Student 2 implementation operates at this frontier, using reuse factors of 1–2 for convolutional layers and 25 for dense layers, achieving 14.5 µs inference latency while consuming approximately 95% of available DSPs.

Several factors constrain scalability to larger models or multi-class extensions on the target KU035 device. First, DSP exhaustion is the primary bottleneck: our design already consumes ∼95% of available DSP blocks at near-minimum reuse factors. Extending to deeper networks or higher-resolution inputs (e.g., 64×64 instead of 48×48) would require either increasing reuse factors—thereby trading latency for feasibility—or adopting LUT-based arithmetic, which increases LUT consumption and may reduce maximum clock frequency. Second, memory bandwidth and capacity impose practical limits: aggressive parallelization demands more concurrent memory ports and greater BRAM partitioning, which can exceed on-chip resources for large models. Third, routing congestion becomes significant in highly parallelized designs, potentially degrading $F_{\max}$ such that even reduced cycle counts do not proportionally improve end-to-end latency. Finally, multi-class extensions increase computational and memory demands: for example, a 10-class classifier would require approximately 5× more DSPs and parameters in the final layer (a consequence of adding 8 more neurons) compared to our binary design. Our Student 2 architecture and heterogeneous reuse factor strategy represent a carefully optimized design point that achieves ultra-low latency (14.5 µs inference, 24.7 µs total detection-to-trigger) while remaining within the strict resource envelope of the KU035 FPGA. This positions our implementation at the practical performance limit for this device class and demonstrates the necessity of co-designing model architecture and hardware configuration to meet real-time sorting constraints.

Table S1. Comparison of Classification Accuracy on LymphoMNIST and MNIST Datasets.

Classifier	Parameters	Test Accuracy	
		Our Dataset	MNIST
DecisionTreeClassifier	criterion=gini, max_depth=10, splitter=best	0.498	0.866
	criterion=gini, max_depth=50, splitter=best	0.453	0.877
	criterion=gini, max_depth=100, splitter=best	0.452	0.879
	criterion=gini, max_depth=10, splitter=random	0.498	0.853
	criterion=gini, max_depth=50, splitter=random	0.449	0.873
	criterion=gini, max_depth=100, splitter=random	0.457	0.875
	criterion=entropy, max_depth=10, splitter=best	0.498	0.873
	criterion=entropy, max_depth=50, splitter=best	0.454	0.886
	criterion=entropy, max_depth=100, splitter=best	0.457	0.886
	criterion=entropy, max_depth=10, splitter=random	0.502	0.861
	criterion=entropy, max_depth=50, splitter=random	0.452	0.883
	criterion=entropy, max_depth=100, splitter=random	0.451	0.881
GradientBoostingClassifier	max_depth=3, n_estimators=10, loss=deviance	0.502	0.846
	max_depth=10, n_estimators=10, loss=deviance	0.555	0.933
	max_depth=50, n_estimators=10, loss=deviance	0.498	0.888
	max_depth=3, n_estimators=50, loss=deviance	0.532	0.926
	max_depth=10, n_estimators=50, loss=deviance	0.610	0.964
	max_depth=3, n_estimators=100, loss=deviance	0.553	0.949
	max_depth=10, n_estimators=100, loss=deviance	0.628	0.969
KNeighborsClassifier	weights=uniform, n_neighbors=1, p=1	0.489	0.955
	weights=uniform, n_neighbors=1, p=2	0.490	0.943
	weights=distance, n_neighbors=1, p=1	0.489	0.955
	weights=distance, n_neighbors=1, p=2	0.490	0.943
	weights=uniform, n_neighbors=5, p=1	0.506	0.957
	weights=uniform, n_neighbors=5, p=2	0.499	0.944
	weights=distance, n_neighbors=5, p=1	0.527	0.959
	weights=distance, n_neighbors=5, p=2	0.517	0.945
	weights=uniform, n_neighbors=9, p=1	0.523	0.955
	weights=uniform, n_neighbors=9, p=2	0.522	0.943
	weights=distance, n_neighbors=9, p=1	0.536	0.955
	weights=distance, n_neighbors=9, p=2	0.533	0.944
LinearSVC	loss=squared_hinge, C=1.0, penalty=l2, multi_class=ovr	0.531	0.912
	loss=squared_hinge, C=10.0, penalty=l2, multi_class=ovr	0.531	0.885
	loss=squared_hinge, C=100.0, penalty=l2, multi_class=ovr	0.531	0.873
LogisticRegression	C=1.0, penalty=l2, multi_class=ovr	0.545	0.917
	C=10.0, penalty=l2, multi_class=ovr	0.542	0.916
RandomForestClassifier	criterion=gini, max_depth=10, n_estimators=10	0.535	0.930
	criterion=gini, max_depth=50, n_estimators=10	0.525	0.948
	criterion=gini, max_depth=100, n_estimators=10	0.523	0.948
	criterion=entropy, max_depth=10, n_estimators=10	0.532	0.933
	criterion=entropy, max_depth=50, n_estimators=10	0.528	0.949
	criterion=entropy, max_depth=100, n_estimators=10	0.531	0.949
	criterion=gini, max_depth=10, n_estimators=50	0.543	0.945
	criterion=gini, max_depth=50, n_estimators=50	0.575	0.968
	criterion=gini, max_depth=100, n_estimators=50	0.578	0.967
	criterion=entropy, max_depth=10, n_estimators=50	0.543	0.947
	criterion=entropy, max_depth=50, n_estimators=50	0.578	0.967
	criterion=entropy, max_depth=100, n_estimators=50	0.575	0.968

S3 Model Calibration Analysis

Probability calibration measures whether a model's predicted confidence scores accurately reflect its true classification accuracy. A well-calibrated model assigns high probabilities to samples it classifies correctly and low probabilities to potential misclassifications. We evaluated calibration performance using Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) with 5-bin reliability analysis[2]. ECE measures the average difference between predicted confidence and actual accuracy across bins, while MCE captures the worst-case deviation. Temperature scaling[2] was applied as a post-hoc calibration technique to assess whether simple rescaling of the output logits could improve calibration metrics.

Figure S2 presents reliability diagrams for the Teacher and Student 1 models before and after temperature scaling. The

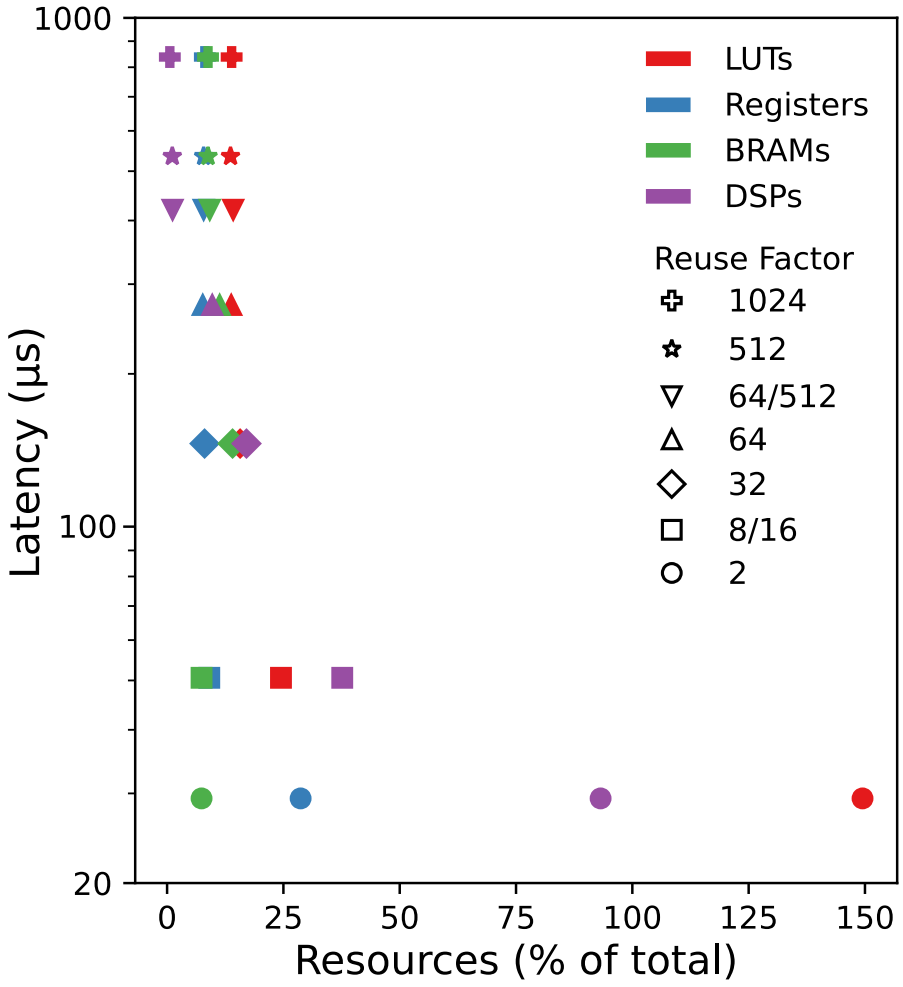

Figure S1. Pareto frontier of optimized model's resource-latency configurations with increasing levels of parallelization. Colors represent the types of resources available on an FPGA: LUTs, Registers, BRAMs, and DSPs. Shapes represent different reuse factors. Resource usages are shown as % of available resources on the KU035 FPGA. Adapted from Wei et al.[1].

Teacher model demonstrates excellent baseline calibration (ECE = 0.90%, MCE = 9.58%), with post-hoc temperature scaling (optimal temperature $T = 1.286$) further improving ECE to 0.41%—a 54.7% reduction. The Student 1 model exhibits moderate baseline calibration (ECE = 1.59%, MCE = 7.42%) and displays underconfidence across low-to-mid confidence bins, meaning it tends to assign lower probabilities than its actual accuracy warrants. This conservative behavior is advantageous for classification tasks where false positives carry higher costs than false negatives, as underconfident predictions naturally provide a safety margin. Temperature scaling reduces worst-case calibration error (MCE from 7.42% to 4.75%), but increases average calibration error (ECE to 2.51%), indicating that attempts to shift predictions toward perfect calibration disrupt the model's inherent conservative bias.

S4 Uncertainty-Aware Rejection and Throughput-Safety Trade-offs

Real-world deployment of machine learning models for cell sorting requires careful consideration of performance under domain shift and the ability to reject ambiguous predictions. In clinical settings, false routing of cells to incorrect collection channels can compromise downstream analyses or therapies. To address this critical deployment concern, we implemented and evaluated an uncertainty-aware rejection framework that quantifies the trade-off between throughput (fraction of samples processed) and safety (accuracy on processed samples).

Our rejection mechanism is based on confidence thresholding, where samples with maximum softmax probability below a threshold τ are rejected and not processed (directed to a discard channel in a physical sorting implementation). We evaluated seven threshold levels: $\tau \in \{0.50, 0.70, 0.80, 0.85, 0.90, 0.95, 0.99\}$ on both our Teacher and Student1 models. To simulate domain shift conditions that would be encountered when deploying the system across different microscopes, illumination

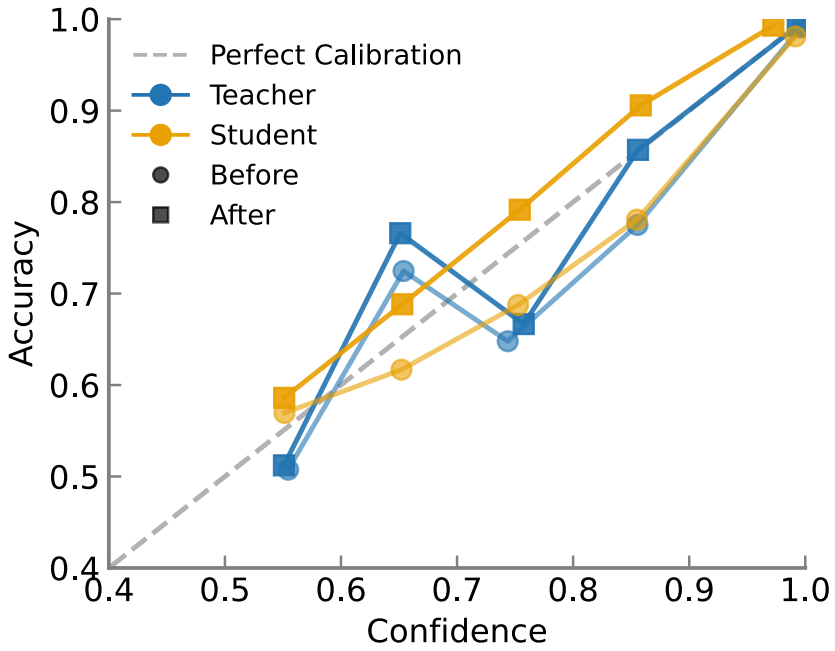

Figure S2. Reliability diagram comparing Teacher (blue) and Student 1 (orange) model calibration before (circles) and after (squares) temperature scaling. Gray dashed line indicates perfect calibration.

conditions, or sample preparation protocols, we applied realistic augmentation transforms to our test dataset. These transforms included color jitter with brightness and contrast variations of 0.4 (simulating illumination changes), random rotations up to 30 degrees (simulating sample orientation variations), random horizontal flips with 50% probability (simulating microscope positioning differences), and Gaussian blur (simulating optical variations). The test dataset consisted of 7,140 samples from the LymphoMNIST dataset, using a 70% test split with random state 42 for reproducibility.

Table S2. Uncertainty-aware rejection performance across confidence thresholds under clean and domain-shifted conditions.

Model	**Threshold** τ	**Coverage** (%)	**Clean Acc.** (%)	**Shift Acc.** (%)	**False-Route Rate** (Clean / Shift, %)
Teacher	0.50	100.0	97.4	94.6	2.6 / 5.4
	0.85	94.6	98.9	97.9	1.1 / 2.1
	0.95	89.4	99.5	99.0	0.5 / 1.0
Student1	0.50	100.0	94.5	84.4	5.5 / 15.6
	0.85	90.2	97.6	94.6	2.4 / 5.4
	0.95	83.3	98.8	97.4	1.2 / 2.6

Table S2 presents the comprehensive results of our uncertainty-aware rejection analysis. At the baseline threshold ($\tau = 0.50$), both models process 100% of samples. The Teacher model achieves 97.4% accuracy on clean data and 94.6% under domain shift, corresponding to a degradation of 2.8%. Student1 achieves 94.5% accuracy on clean data and 84.4% under shift, showing a 10.1% degradation. This baseline comparison confirms that the Teacher model exhibits superior robustness to distribution shift.

As we increase the confidence threshold, both models demonstrate the expected trade-off: coverage decreases while accuracy on processed samples increases. At the intermediate threshold ($\tau = 0.85$), the Teacher processes 94.6% of samples with 98.9% accuracy on clean data and 97.9% accuracy under shift. The false-route rate (percentage of misclassified samples among processed ones) increases from 1.1% on clean data to 2.1% under shift. Student1 at this threshold processes 90.2% of samples with 97.6% clean accuracy and 94.6% shifted accuracy, with false-route rates of 2.4% and 5.4% respectively.

In conservative mode ($\tau = 0.95$), the Teacher maintains 99.5% accuracy on clean data and 99.0% accuracy under domain shift while processing 89.4% of samples. The false-route rate remains below 1% (0.5% clean, 1.0% shifted), making this operating point particularly suitable for clinical applications where sorting accuracy is paramount. Student1 achieves 98.8%

accuracy on clean data and 97.4% under shift at 83.3% coverage, with false-route rates of 1.2% and 2.6% respectively. These results demonstrate that both models can operate in a mode where ambiguous cells are rejected, ensuring that over 98% of processed cells are correctly classified even under domain shift conditions.

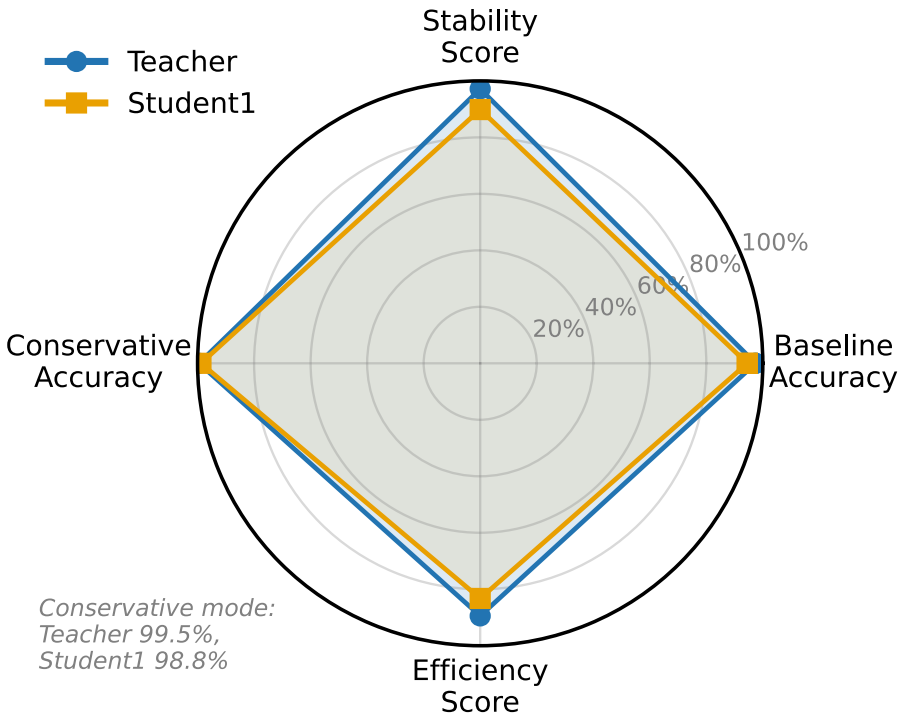

Figure S3. Radar chart visualization of model robustness across four dimensions: Baseline Accuracy (performance at $\tau = 0.50$), Stability Score (robustness to domain shift), Conservative Accuracy (performance at $\tau = 0.95$), and Efficiency Score (coverage at $\tau = 0.95$). The Teacher model (blue) demonstrates superior robustness to domain shift and maintains higher throughput in conservative mode, while Student1 (orange) achieves remarkable parameter efficiency with acceptable performance degradation under distribution shift.

Figure S3 synthesizes these findings in a radar chart visualization across four key dimensions: Baseline Accuracy (performance at $\tau = 0.50$ with 100% coverage), Stability Score (100% minus degradation percentage, quantifying robustness to domain shift), Conservative Accuracy (performance at $\tau = 0.95$ in high-confidence mode), and Efficiency Score (coverage maintained at $\tau = 0.95$). This multi-dimensional view clearly illustrates that while the Student model achieves remarkable parameter efficiency with a 5000-fold compression, the Teacher maintains superior robustness to domain shift and higher throughput in conservative mode.

From a hardware implementation perspective, the confidence threshold mechanism can be realized with minimal overhead. After the FPGA computes the softmax probabilities, a simple comparator operation checks whether the maximum probability exceeds the threshold τ. This comparison adds negligible latency (less than 0.1 microseconds) and requires minimal additional logic resources, ensuring that uncertainty-aware rejection does not compromise the sub-25 microsecond total latency budget reported in our main results. The threshold value can be configured as a runtime parameter, allowing system operators to adjust the throughput-safety trade-off based on specific application requirements.

These results establish a quantitative framework for deploying our models in clinical settings where both throughput and safety must be carefully balanced. Research applications prioritizing maximum sample throughput might operate at $\tau = 0.85$ (over 90% coverage with approximately 98% accuracy), while clinical diagnostic applications demanding highest accuracy would operate at $\tau = 0.95$ (over 83% coverage with over 98% accuracy). The explicit quantification of false-route rates under domain shift provides critical information for regulatory compliance and risk assessment in clinical deployment scenarios.

References

1. Wei, Y. *et al.* Low latency optical-based mode tracking with machine learning deployed on FPGAs on a tokamak. *The Rev. scientific instruments* **95**, DOI: 10.1063/5.0190354 (2024).

2. Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. On calibration of modern neural networks. *Int. Conf. on Mach. Learn.* **abs/1706.04599**, 1321–1330 (2017).