Supplementary Material of Polymorph: Energy-Efficient Multi-Label Classification for Video Streams on Embedded Devices

Saeid Ghafouri[1] Mohsen Fayyaz[2] Xiangchen Li[3] Deepu John[4]
Bo Ji[3] Dimitrios S. Nikolopoulos[3] Hans Vandierendonck[1]

[1]Queen's University Belfast, United Kingdom [2]Microsoft, Berlin, Germany
[3]Virginia Tech, Blacksburg, VA, USA [4]University College Dublin, Ireland

s.ghafouri@qub.ac.uk mohsenfayyaz@microsoft.com lixiangchen@vt.edu deepu.john@ucd.ie
boji@cs.vt.edu dsn@vt.edu h.vandierendonck@qub.ac.uk

Table 1. Operational metrics for different methods.

Method	Retrained	Data needed	Update	E2E retune
CACTUS	per classifier	context-only	small	no
Larger model	full model	full dataset	large	yes
MoE	experts + router	full dataset	medium	partial
Polymorph	LoRA only	label subset	small	no

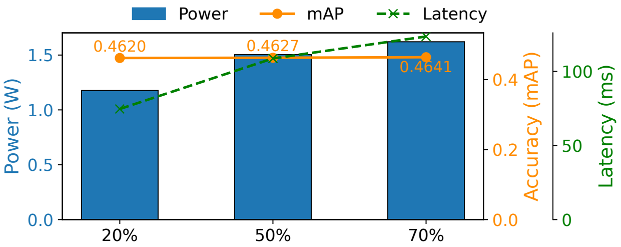

Figure 1. Ablation on the depth of LoRA application. We compare applying adapters to the last 20%, 50%, and 70% of layers.

1. Operational trade-offs

Beyond accuracy and latency, methods also differ in training and deployment overhead. Table 1 summarizes these aspects. CACTUS requires training and storing a separate classifier per context, which becomes inefficient as the label space grows and contexts need to be frequently updated. Larger monolithic models demand full retraining and distributing large binaries, making them costly to adapt and cumbersome to deploy at the edge. Mixture-of-Experts (MoE) approaches offer modularity but typically require co-training experts together with the router, so updating one expert often forces rebalancing or retraining the entire system. In contrast, Polymorph trains each LoRA independently, can add or remove them without affecting others, and deploys small incremental updates. This modularity not only reduces storage and retraining cost but also allows periodic re-grouping of the router when co-occurrence patterns shift. Such lightweight and flexible updates are particularly important for embedded video, where memory is limited and adaptation must remain fast and frequent.

2. Ablation Study

An important design decision in Polymorph is the depth at which LoRA adapters are applied. The main experiments used the final 20% of layers, motivated by the concentration of class-discriminative features in later transformer blocks. To evaluate the impact of varying this proportion, we conducted an ablation study where adapters were applied to 20%, 50%, and 70% of the layers, starting from the output end of the network. Figure 1 summarizes the trade-offs across energy, latency, and accuracy. These results confirm that applying LoRAs to deeper portions of the network increases computational cost with only marginal accuracy improvements. Expanding adaptation from 20% to 70% of layers increases latency by more than 65% and power by 37%, while mAP improves by less than 0.3 points. This validates our choice of restricting LoRA to the final 20% of layers: it achieves near-maximum accuracy while keeping energy and latency within real-time embedded budgets.

3. Effect of False Positives

Figure 2 illustrates the effect of false negatives in a single video sequence. The ground truth labels remain constant across frames, yet the base model frequently predicts non-existent classes (e.g., 1, 10), leading to false positives. In contrast, Polymorph maintains accurate predictions throughout. Due to using narrower, context-specific clas-

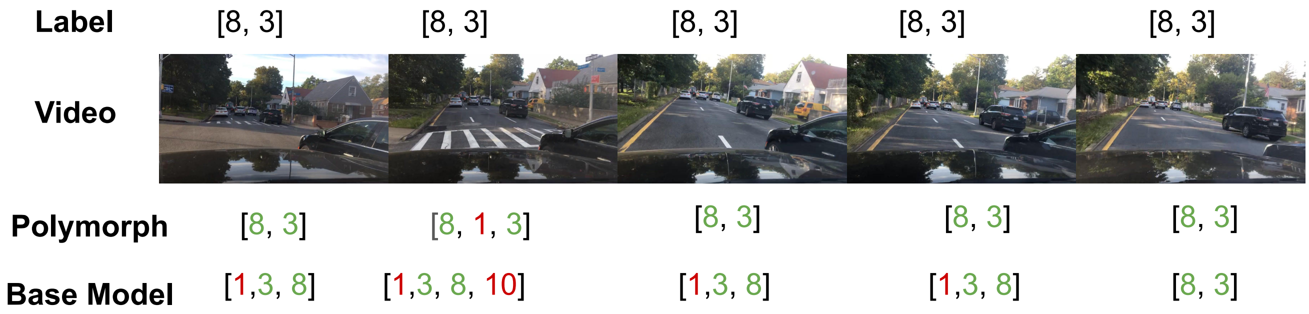

Figure 2. Polymorph avoids false positives by using narrower classifiers, unlike the base model which predicts non-existent classes (e.g., 1, 10). 1: Person, 3: Car, 8: Truck, 10: Traffic Light

sifiers that restrict the label space per frame, reducing the chance of spurious detections. By focusing on relevant label subsets, Polymorph avoids activating unrelated classes.