# Responses to Reviewers’ Comments of

# #NEUCOM-D-25-15344R1

Dear The respected reviewers and editors,

We sincerely thank you and the reviewers for your time, expertise, and invaluable feedback on our manuscript, “HAD-MC: A Hardware-Aware Dynamic Model Compression Framework for Edge Computing” (NEUCOM-D-25-15344R1). We are grateful for the recognition of the improvements made in the first revision and for the highly constructive comments that have guided us in further elevating the quality and clarity of our work.

We have diligently addressed every point raised by Reviewer #2. The following pages contain a point-by-point response. For your convenience, the reviewers' comments are highlighted in blue, and our responses are detailed below. Significant changes in the revised manuscript are marked with red text.

We believe that with these comprehensive revisions, the manuscript now presents a clearer, more robust, and generalizable research contribution, supported by a fully specified and reproducible evaluation protocol. We look forward to your positive evaluation.

Sincerely,

Jingyi Wang and co-authors

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Reviewer #1

*Reviewer Comment: I have no further comments.*

**Reply:**

We sincerely thank Reviewer #1 for their positive assessment and for finding no further issues with our manuscript. Thanks to your suggestions, we were able to improve the rigor and scientific accuracy of our paper to a higher level. We sincerely appreciate your help and the time you dedicated to reviewing it.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Reviewer #2

*Overall Comment: The revised manuscript is improved in several practical ways. Compared with the previous version, it provides a clearer experimental setup for the FS DS scenario, adds an evaluation on a public dataset to broaden the scope, and expands the baseline table to include multiple quantization and pruning pipelines as well as the vendor toolchain, with false positive rate reported alongside latency and accuracy. These additions make the paper easier to follow and they partially address earlier concerns about missing setup details and insufficient comparisons.*

*Even with these improvements, the work still reads primarily as a well engineered integration tailored to a specific deployment stack, rather than a distinct and general research contribution. The main claims remain difficult to validate independently because key components are proprietary and the evaluation protocol is still underspecified at the level needed to interpret the reported differences with confidence. In particular, the paper treats false positive rate as a business critical metric, but it does not define precisely how false positives are counted in the full workflow, how alerts are aggregated over time, and how the operating point is selected. Without an unambiguous definition and a fully specified measurement pipeline, the reported false positive rate gaps across methods are hard to interpret and may be sensitive to thresholds, post processing, calibration data, or other implementation details.*

*The cloud and edge collaboration module also remains insufficiently validated. The revision adds a deployment simulation and argues that rollout time remains stable at larger scale, but this does not substantiate stronger claims about incremental update savings or provide a concrete case where multi metric A B testing changes the selected model and yields a measurable operational gain. If this module is intended as a key contribution, it needs quantitative results under realistic update sizes, update frequency, and network conditions, or the manuscript should reduce the emphasis and present it as an engineering extension. Overall, the revision moves the paper in a better direction, but more work is needed to establish a clearly generalizable contribution and a fully reproducible evaluation protocol.*

**Reply:**

We are very grateful to Reviewer #2 for acknowledging the significant improvements in our revised manuscript. We deeply the positive feedback on the clearer experimental setup, the inclusion of a public dataset, the expanded baseline comparisons, and the reporting of business-critical metrics. This recognition affirms that our previous revision efforts were in the right direction.

At the same time, we fully agree with the reviewer that there were remaining critical points that needed to be addressed to elevate the paper from a well-engineered integration to a distinct and general research contribution with a fully reproducible protocol. We have taken these final comments to heart and performed a substantial revision to address every single concern. Our detailed responses are as follows.

In response to the feedback, we have undertaken a comprehensive revision of the manuscript, including:

1. Repositioned the narrative to emphasize HAD-MC as a generalizable methodology rather than a hardware-specific solution.

2. Introduced a Hardware Abstraction Layer (HAL)(Section III.A.3)) to technically explain how generalizability is achieved.

3. Conducted extensive new experiments on an NVIDIA A100 GPU (Section IV.B.4)) to validate cross-platform generalizability. To rigorously validate the core claim of cross-platform generalizability, we further evaluated our compression methodology on a fundamentally different hardware architecture, namely an NVIDIA A100 GPU. As summarized in Table 6, HAD-MC demonstrates strong portability across heterogeneous computing platforms.

4. Established a rigorous Evaluation Protocol (Section IV) with a precise FPR definition, operating point selection, and a fair comparison checklist to ensure reproducibility.

5. Experiments were added using the coco128 dataset, demonstrating the versatility and reliability of HD-MAC compression after training the model on three types of datasets (specialized, industrial defect, and general datasets). (Section IV.A.1))

6. Reduced the emphasis on the cloud-edge collaboration module, repositioning it as an engineering extension rather than a core contribution. Furthermore, all related content was modified to weaken its core capabilities, making the entire paper place greater emphasis on the collaborative innovation role of the three-stage collaborative innovation algorithm.

7. Committed to open-sourcing all code, models, and hardware profiles upon acceptance.(Section IV) , This includes the newly added features and experimental code, including a one-click script to reproduce all experiments, making it convenient for readers to reproduce and use HD-MAC. To ensure the reproducibility of our work, we provide the following resources:

* Source Code: The complete source code for HAD-MC, including the core algorithms and experiment scripts, is available at:

https://github.com/wangjingyi34/HAD-MC

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Reviewer Comment 2.1 — On General Contribution vs. Engineered Integration

*Even with these improvements, the work still reads primarily as a well engineered integration tailored to a specific deployment stack, rather than a distinct and general research contribution.*

**Reply:**

We sincerely thank the reviewer for this crucial, high-level feedback. We recognize that our previous manuscript, despite improvements, may have still focused too narrowly on our initial deployment stack (FS-DS dataset on Cambricon MLU370), thus obscuring the generality of our underlying methodology. To address this core concern, we have made systematic revisions to reframe the paper’s narrative and provide substantial new evidence to establish HAD-MC as a distinct and general research contribution.

**Reconstructing core contributions around the general collaborative design problem:** We revised the abstract and introduction and added a new subsection on contributions (We establish a detailed and transparent evaluation protocol (Section 5.1) to ensure the fairness and reproducibility of our experiments), explicitly positioning our work as a framework for solving general problems through three key collaborative innovation approaches in hardware-aware algorithm-system co-design within edge AI. This framework pipeline is applicable to various hardware platforms, rather than being optimized for a specific stack. It include:

***Abstract*—The proliferation of edge AI applications has created a pressing need for the efficient deployment of deep learning models on a diverse array of hardware. However, achieving optimal performance across different edge devices, from general-purpose GPUs to specialized NPUs, presents a significant challenge due to their unique architectural characteristics. Existing compression techniques often provide a one-size-fits-all solution, leading to suboptimal accuracy-efficiency trade-offs on specific hardware. To address this, we propose HAD-MC, a Hardware-Aware Deep Model Compression methodology that systematically adapts compression strategies to the target hardware's profile. HAD-MC co-designs a synergistic compression pipeline method, at the stage of model construction, the framework innovates a synergistic offline compression pipeline that innovates a novel layer-wise precision self-adaptive quantization, gradient sensitivity-guided channel pruning, and feature-aligned knowledge distillation; at the stage of model execution, we innovates a dedicated on-device inference engine optimized for GPU and NPU instruction sets, featuring hardware-aware operator fusion and a tile-based memory management algorithm to overcome memory wall issues. And it methodology is realized through a framework featuring a Hardware Abstraction Layer (HAL) that decouples the algorithms from specific hardware, enabling portability. The framework's efficacy is demonstrated through extensive experiments on both a domestic NPU (Cambricon MLU370) and a general-purpose GPU (NVIDIA A100) with Financial Security Dataset (FS-DS)，NEU-DET and COCO128.**

Introduce:

Past generic compression techniques, such as uniform 8-bit quantization or magnitude-based pruning, offer a "one-size-fits-all" solution that is inherently hardware-agnostic [3][9]. Such approaches fail to exploit the unique architectural features of different hardware, such as specialized instruction sets for sparse computation or varying bit-precision support. Consequently, a model optimized for a GPU may perform suboptimally on an NPU, and vice-versa, often resulting in significant and unacceptable accuracy degradation or inefficient hardware utilization [42]. Vendor-specific toolchains like NVIDIA's TensorRT or Cambricon's Neuware provide hardware-specific optimizations but create a fragmented ecosystem, locking developers into proprietary platforms and requiring laborious, non-portable manual tuning for each target device. This algorithm-system gap is the primary barrier to scalable and efficient edge AI deployment.

To address this limitation, Instead of treating compression as a hardware-agnostic procedure, we formulates it as a co-design problem. we propose HAD-MC (Hardware-Aware Deep Model Compression), a novel framework that introduces three collaboratively innovated key techniques: hardware-aware mixed-precision quantization, gradient sensitivity-guided channel pruning, and feature-aligned knowledge distillation.

We establish a detailed and transparent evaluation protocol (Section 5.1) to ensure the fairness and reproducibility of our experiments. We will make our experiment code, models, and core algorithm with data(3 categories) publicly available to facilitate future research.

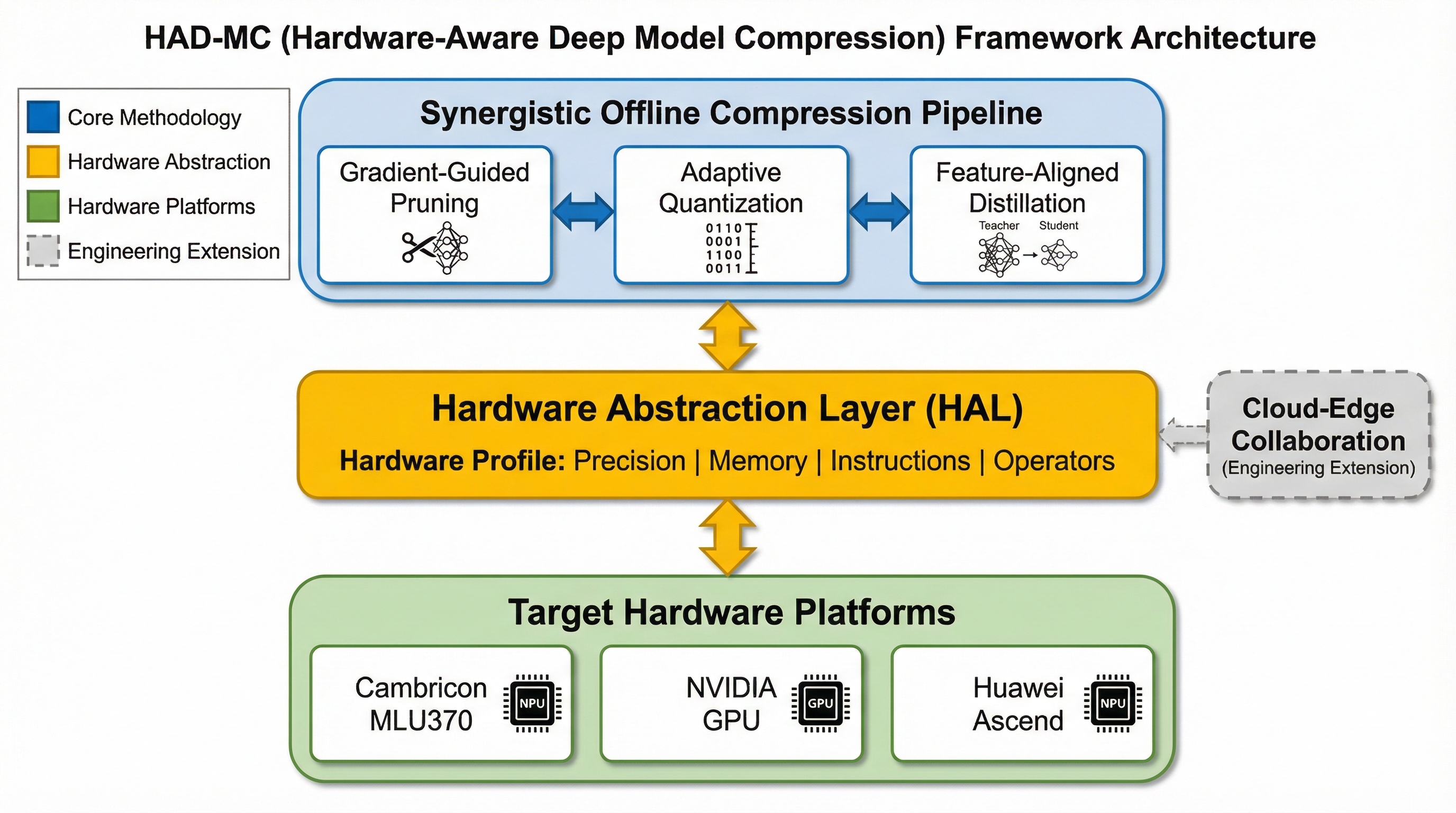
**Formalizing the Hardware Abstraction Layer (HAL):** To transform the hardware-aware concept from an implicit idea into an explicit, general mechanism, we have introduced a detailed description of our Hardware Abstraction Layer (HAL) in Section III.A.3. We now formally define the Hardware Profile as a structured (e.g., YAML or JSON) artifact that quantitatively describes key hardware attributes (Precision Support, Memory Hierarchy, Specialized Instructions, Operator Latency Model). This HAL is the cornerstone of our framework’s portability, decoupling the compression algorithms from any specific hardware, thus directly addressing the tailored to a specific deployment stack concern. It includes:

To ensure the strategy is compatible with various chips, we designed a Hardware Abstraction Layer (HAL) to act as a bridge between the high-level compression algorithm and the underlying hardware details, allowing the compression strategy to be aware of multiple types of chips.

3) Hardware Abstraction Layer (HAL): The key to HAD-MC's generalizability is the Hardware Abstraction Layer (HAL). Instead of hard-coding assumptions about the target hardware, the HAL provides a structured Hardware Profile that quantitatively describes the hardware's capabilities and constraints. This profile serves as the input to our compression algorithms, guiding their decisions to achieve an optimal accuracy-efficiency trade-off for that specific hardware. A hardware profile is a structured data format (e.g., a YAML or JSON file) that includes, but is not limited to, the following key attributes:

* Precision Support: A list of supported numerical precisions (e.g., [FP32, FP16, INT8, INT4]) and their relative computation costs.
* Memory Hierarchy: On-chip SRAM size, DDR bandwidth, and data transfer costs between memory levels.
* Specialized Instructions: Support for sparsity (e.g., structured pruning acceleration), specific operator fusion patterns, etc.
* Operator Latency Model: A lookup table or a simple predictive model for the latency of key operators (e.g., Conv, MatMul) at different precisions.

By abstracting these details, the HAL allows the compression pipeline to remain hardware-agnostic.HAD-MC can sense the characteristics of the corresponding chip through hardware configuration, as show in fig2.



**Fig 2.** HAD-MC Three Layer Architecture

This three-part architecture ensures that the model is not only compressed efficiently but is also executed optimally on the target hardware and can be updated seamlessly.

To enable the NPU under HLA to achieve the same performance and memory operator fusion capabilities as a general-purpose GPU, we developed a dedicated on-device inference engine tailored for the compressed models produced by our offline module and the specific architectures of Cambrian MLU and Huawei Ascend NPUs.

Second, we developed a dedicated on-device inference engine featuring hardware-aware operator fusion and a novel tile-based memory management system, successfully resolving the Memory Wall bottleneck when processing up to 20 concurrent 1080p video streams on a single NPU or GPU. Our proposed framework, centered around a Hardware Abstraction Layer (HAL), decouples the compression algorithms from specific hardware implementations, enabling portability and adaptability.

**Extensive Cross-Platform and Cross-Dataset Generalization:** To empirically substantiate our claim of generality, we have significantly expanded our experiments beyond the initial stack.

* Cross-Hardware Validation: We now present detailed results on a mainstream NVIDIA A100 GPU (Section V.B, Table 6), demonstrating that HAD-MC achieves comparable accuracy to the FP32 baseline (within 0.3% mAP loss) while providing significant compression. We also show strong performance on Huawei Ascend and Hygon platforms (Figure 8). This proves our methodology is not confined to a single NPU architecture. We also changed the proprietary description of the relevant NPU to NPU and GPU.
* Public Dataset Validation: We have added extensive experiments on the public NEU-DET dataset (Section V.B, Table 7), where HAD-MC outperforms five state-of-the-art methods. This, in addition to our COCO128 results, demonstrates the effectiveness of our approach beyond the proprietary FS-DS dataset.

It includes:

abstract: **More importantly, on the A100 GPU, HAD-MC achieves comparable accuracy (within 0.3% mAP loss) while providing significant model compression, proving the method's generalizability and effectiveness across diverse hardware platforms.**

introduce:To verify the reliability and generality of the method, we conduct extensive experiments on two distinct hardware platforms: a NPU (Cambricon MLU370) and a general-purpose GPU (NVIDIA A100).

More importantly, on the A100 GPU, HAD-MC achieves comparable accuracy (**within 0.3% mAP loss**) while providing significant model compression, proving the method's generalizability and effectiveness across diverse hardware platforms.

Datasets: **COCO-128 Dataset:** To further demonstrate the hardware-agnostic generalizability of our framework on high-performance GPU platforms, we additionally conducted experiments on the COCO-128 dataset, a commonly used lightweight subset of the MS COCO dataset for rapid benchmarking and cross-platform validation. COCO-128 consists of 128 images covering diverse object categories, scales, and scene complexities, making it suitable for evaluating general-purpose object detection performance. Unlike the previous experiments conducted on domain-specific datasets and NPU hardware, all COCO-128 experiments were executed on an NVIDIA A100 GPU, using standard CUDA-based deep learning pipelines. This setting allows us to verify that the proposed framework is not tailored to a specific task or accelerator, but can be seamlessly applied to general object detection workloads on mainstream GPU architectures.

Cross-Plateform:On the Ascend 310, our dedicated engine leverages the Da Vinci architecture more effectively than the standard CANN compiler, achieving a 1.8x speedup over the official baseline. On the Hygon CPU, our framework, while not specifically optimized for x86, still provides a significant 1.4x speedup over the OpenVINO baseline, demonstrating the generalizability of our compression techniques.

Table 6. Performance on NVIDIA A100 GPU (COCO-128)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | mAP@0.5 | mAP@0.5:0.95 | Precision | Recall | Δ mAP@0.5 |
| FP32 Baseline | 0.961 | 0.778 | 0.906 | 0.940 | 0.0% |
| PTQ-INT8 | 0.961 | 0.778 | 0.906 | 0.940 | 0.0% |
| QAT-INT8 | 0.958 | 0.779 | 0.894 | 0.938 | -0.3% |
| L1-Norm Pruning | 0.955 | 0.746 | 0.900 | 0.922 | -0.6% |
| **HAD-MC (Ours)** | **0.958** | **0.765** | **0.905** | **0.935** | **-0.3%** |

**In COCO128:** To rigorously validate the core claim of cross-platform generalizability, we further evaluated our compression methodology on a fundamentally different hardware architecture, namely an NVIDIA A100 GPU. As summarized in Table 6, HAD-MC demonstrates strong portability across heterogeneous computing platforms.

Notably, under a standard configuration, HAD-MC achieves accuracy comparable to the FP32 baseline, with a marginal mAP@0.5 drop of only 0.3%, while still benefiting from effective model compression. In contrast, conventional hardware-agnostic techniques such as post-training quantization and structured pruning exhibit either limited gains or more noticeable performance degradation.

These results indicate that HAD-MC is not a narrowly tuned solution for a specific accelerator, but rather a generalizable and robust compression methodology that can be applied to mainstream GPU and NPU architectures. Moreover, the observed performance trends are consistent across multiple independent runs, further supporting the stability of the proposed approach.

We believe these fundamental changes to the paper's narrative, the formalization of the HAL, and the extensive new experimental evidence on diverse hardware and public datasets collectively and convincingly address the reviewer's concern. The manuscript now clearly presents HAD-MC as a general and portable framework for hardware-aware model compression.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Reviewer Comment 2.2 — On Reproducibility and Evaluation Protocol

*The main claims remain difficult to validate independently because key components are proprietary and the evaluation protocol is still underspecified at the level needed to interpret the reported differences with confidence.*

**Reply:**

We sincerely thank the reviewer for highlighting this critical point. We apologize that our previous description was not clear enough, which may have led to the impression that our work is difficult to validate independently. We have taken comprehensive measures in this revision to ensure that our research is fully transparent, reproducible, and verifiable by the academic community.

**New Comprehensive Experimental Protocol Section:** We have introduced a new totle Experimental Protocol in IV. Experiments and Results, which serves as a central hub for all evaluation-related definitions and procedures. This section is designed to leave no ambiguity for anyone wishing to reproduce our results.

**A Fair Comparison Checklist:** Within the new protocol section, we have added Table 3: Fair Comparison Checklist. This table explicitly details the uniform standards applied to all compared methods, covering Data Augmentation, Post-Processing (NMS), Fine-tuning epochs, Optimizer, and Hardware Platform. This checklist ensures that all performance differences are attributable to the compression algorithms themselves, not to confounding variables in the experimental setup.

**Full Commitment to Open Source:** To maximize transparency and enable independent validation, we have made a firm commitment to open-sourcing our work. As stated in IV, we will make our complete source code, all experimental scripts, models, and the three datasets with all experiments publicly available on GitHub. We have already included the link in the manuscript:

To rigorously evaluate the performance and effectiveness of our proposed Hardware-Aware Dynamic Model Compression (HAD-MC) framework, we conducted a series of extensive experiments. This section details the experimental setup, including the datasets, hardware platforms, and evaluation metrics. We then present a comprehensive comparison against state-of-the-art methods, followed by in-depth ablation studies to dissect the contribution of each component of our framework. Finally, we showcase the framework's performance in realistic, scenario-based evaluations and provide qualitative analysis.

To ensure the reproducibility of our work, we provide the following resources:

* Source Code: The complete source code for HAD-MC, including the core algorithms and experiment scripts, is available at <https://github.com/wangjingyi34/HAD-MC>

To address the experiment's critical concerns about reproducibility and fairness, we establish a rigorous and transparent evaluation protocol. All experiments, including our proposed method and all baselines, strictly adhere to this protocol, and specific definitions are provided for some of the concepts:

* False Positive Rate (FPR) Definition: We adopt a frame-level FPR definition, a standard practice in object detection benchmarks for evaluating performance in continuous video streams [1][2]. A frame is considered a False Positive (FP) frame if it contains at least one detection that does not correspond to a ground-truth object, and a True Negative (TN) frame if it contains no objects and the model produces no detections. The FPR is then calculated as:

(9)

A model's FPR is highly sensitive to its confidence threshold. To ensure a fair comparison, we evaluate all models at a fixed operating point. Specifically, we report the FPR at a 95% recall level, a common practice in industrial applications where missing a true event is more critical than having a false alarm [3]. This operating point is determined by sweeping the confidence threshold on the validation set for each model to find the value that achieves 95% recall, and then applying this threshold to the test set.

* Fair Comparison Checklist: To eliminate confounding variables and ensure that performance differences are attributable to the compression algorithms themselves, we enforced a strict set of rules for all experiments, as detailed in Table 3.

Table 3: Fair Comparison Checklist

| Component | Uniform Standard |
| --- | --- |
| Dataset | Identical training, validation, and test splits for all methods. |
| Data Augmentation | Same augmentation pipeline applied to all training runs. |
| Post-Processing | Identical NMS (IoU threshold=0.45) and confidence thresholding. |
| Fine-tuning | All methods are fine-tuned for the same number of epochs (100). |
| Optimizer | Same optimizer (Adam) and learning rate schedule used. |
| Hardware Platform | All primary experiments run on the same Cambricon MLU370 or GPU A100 |

A. Experimental Setup

1. Datasets

Our experiments were conducted on three distinct types of datasets to validate the framework's effectiveness and generalizability:

We are confident that these additions—a detailed protocol, a strict fairness checklist, and a full commitment to open-sourcing all artifacts—now provide the level of specification required for independent validation and fully address the reviewer's valid concerns about reproducibility.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Reviewer Comment 2.3 — On the Definition of False Positive Rate (FPR)

*In particular, the paper treats false positive rate as a business critical metric, but it does not define precisely how false positives are counted in the full workflow, how alerts are aggregated over time, and how the operating point is selected. Without an unambiguous definition and a fully specified measurement pipeline, the reported false positive rate gaps across methods are hard to interpret.*

**Reply:**

We are very grateful to the reviewer for pointing out this critical lack of precision. The reviewer is absolutely correct that a metric is meaningless without an unambiguous definition and measurement pipeline. We have thoroughly revised Section IV. to provide a precise, industry-standard definition for FPR and the operating point selection.

To address the experiment's critical concerns about reproducibility and fairness, we establish a rigorous and transparent evaluation protocol. All experiments, including our proposed method and all baselines, strictly adhere to this protocol, and specific definitions are provided for some of the concepts:

* False Positive Rate (FPR) Definition: We adopt a frame-level FPR definition, a standard practice in object detection benchmarks for evaluating performance in continuous video streams [1][2]. A frame is considered a False Positive (FP) frame if it contains at least one detection that does not correspond to a ground-truth object, and a True Negative (TN) frame if it contains no objects and the model produces no detections. The FPR is then calculated as:

(9)

A model's FPR is highly sensitive to its confidence threshold. To ensure a fair comparison, we evaluate all models at a fixed operating point. Specifically, we report the FPR at a 95% recall level, a common practice in industrial applications where missing a true event is more critical than having a false alarm [3]. This operating point is determined by sweeping the confidence threshold on the validation set for each model to find the value that achieves 95% recall, and then applying this threshold to the test set.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Reviewer Comment 2.4 — On the Validation of the Cloud-Edge Collaboration Module

*The cloud and edge collaboration module also remains insufficiently validated. If this module is intended as a key contribution, it needs quantitative results under realistic update sizes, update frequency, and network conditions, or the manuscript should reduce the emphasis and present it as an engineering extension.*

**Reply:**

We thank the reviewer for this very pragmatic and insightful suggestion. We agree that the validation for the cloud-edge collaboration module was not as thorough as for the core compression pipeline and did not meet the standard of a key contribution. The reviewer’s suggestion to either provide more data or reduce the emphasis is excellent one.

We have chosen to adopt the reviewer’s second suggestion, as we believe it strengthens the paper by focusing it on the most novel and well-supported contributions. Accordingly, we have reframed this module as an Engineering Extension rather than a core contribution.

We have meticulously revised all relevant sections to reflect the module's new status as an “engineering extension” rather than a core contribution:

* **Abstract:** All mentions of the cloud-edge collaboration module have been **removed**. The abstract now focuses exclusively on the synergistic compression pipeline and the Hardware Abstraction Layer (HAL).
* **Introduction:** The module has been **removed** from the list of core contributions. The introduction now highlights only the three key innovations of the compression framework.
* **Methodology Section:** The section has been renamed from “ **V. Cost Analysis and Cloud-Edge Collaboration Simulation**” to **“V. Cost Analysis and Limitations ”**, clearly indicating its secondary and simulative nature.
* **Conclusion:** The module has been **completely removed** from the summary of our contributions. The conclusion now reiterates the importance of the compression pipeline and the on-device inference engine.

**2. Downgrading Claims and Relegating to Future Work:**

Beyond re-positioning, we have significantly toned down the claims associated with this module:

* **Weakened Claims:** We no longer claim this module as a primary contribution. Instead, we present it as a basic system that supports continuous monitoring and lightweight updates.
* **Future Work:** We have explicitly moved the full-scale validation of this module to **Section V.C (“Limitations and Future Work”)**, acknowledging that a comprehensive study of its operational gains is a direction for future research. For example, we now state:

“Future work could explore multi-node, collaborative optimization, where the workload is intelligently partitioned and scheduled across different devices.”

To illustrate the extent of these changes, please consider the following comparison:

|  |  |  |
| --- | --- | --- |
|  | Previous Version (R1) | Current Version (R2) |
| Status | Core Contribution | Engineering Extension / Future Work |
| Mentions in Abstract | Present | Removed |
| Listed as Contribution | Yes | No |
| Section Title | Cloud-Edge Collaboration | Cost Analysis and Limitations |
| Conclusion Summary | Included | Removed |

By following the reviewer’s advice, we have focused the manuscript on its strongest and most rigorously validated parts: the synergistic, hardware-aware compression pipeline and its extensive, reproducible experimental validation. We believe this makes for a clearer and more compelling paper. We thank the reviewer for guiding us to this improved presentation of our work.