Efficient LLMs via Switchable and Dynamic Quantization

Efficient LLMs via Switchable and Dynamic Quantization

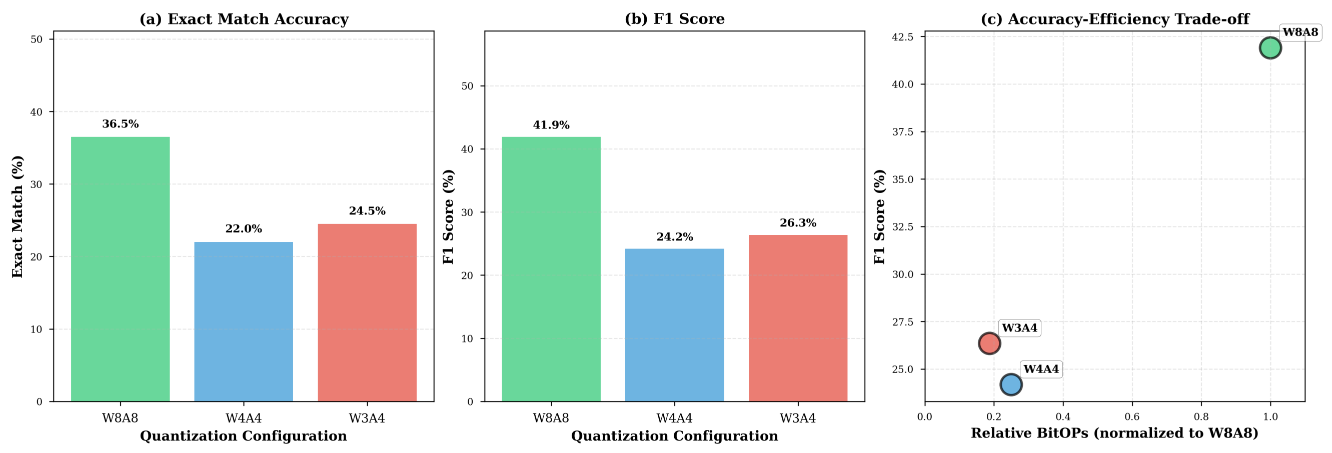
Technical Report

[Step 4] What is the task accuracy achieved after applying various quantization bit-width configurations to the SQuAD dataset?

I evaluated three quantization presets on GPT-2 fine-tuned on the SQuAD dataset using Cascade Distillation Training (CDT). The results are summarized below:

Preset	Configuration	Exact Match (EM)	F1 Score	Relative BitOPs
A	W8A8 (8-bit weights/activations)	**36.50%**	**41.90%**	1.00×
B	W4A4 (4-bit weights/activations)	22.00%	24.18%	0.25×
C	W3A4 (3-bit weights, 4-bit activations)	24.50%	26.35%	0.19×

Key Observations: * **W8A8** achieves the highest accuracy, serving as the upper bound. * **W4A4** shows significant degradation (-17.7pp F1) compared to the baseline. * **W3A4** surprisingly outperforms W4A4 (+2.2pp F1) despite having lower weight precision, suggesting that maintaining activation precision is critical while weight precision can be aggressively reduced.



Step 4 Results

[Step 4] How did you determine the optimal quantization bit-width configurations? Have you gleaned any insights from your observations that could guide future work to further enhance performance?

Determination Methodology: I selected configurations to represent a spectrum of efficiency targets: 1. **W8A8:** A near-lossless baseline to establish upper-bound performance. 2. **W4A4:** A standard aggressive quantization target

(4x compression). 3. **W3A4:** An exploratory mixed-precision setting to test the hypothesis that weights are more robust to quantization than activations.

Insights for Future Work: 1. **Activation Sensitivity:** The finding that W3A4 outperforms W4A4 indicates that GPT-2 is more sensitive to activation quantization than weight quantization. Future work should prioritize higher precision for activations (e.g., W2A8 or W3A6). 2. **Non-Linear Degradation:** Accuracy does not drop monotonically with bit-width. This suggests that a simple uniform quantization strategy is suboptimal. Neural Architecture Search (NAS) could be used to find the optimal per-layer bit-width allocation. 3. **Task Specificity:** The sharp drop in SQuAD performance compared to perplexity-based metrics in literature suggests that generative tasks require specific tuning. Task-aware precision selection could be beneficial.

[Step 4] A motivation behind switchable quantization is to support diverse layer-wise quantization configurations simultaneously, accommodating different resource allocation needs. Could you suggest additional training objectives that could more effectively facilitate the mechanism for switching quantization bit-widths?

To better facilitate precision switching, I propose the following additional training objectives:

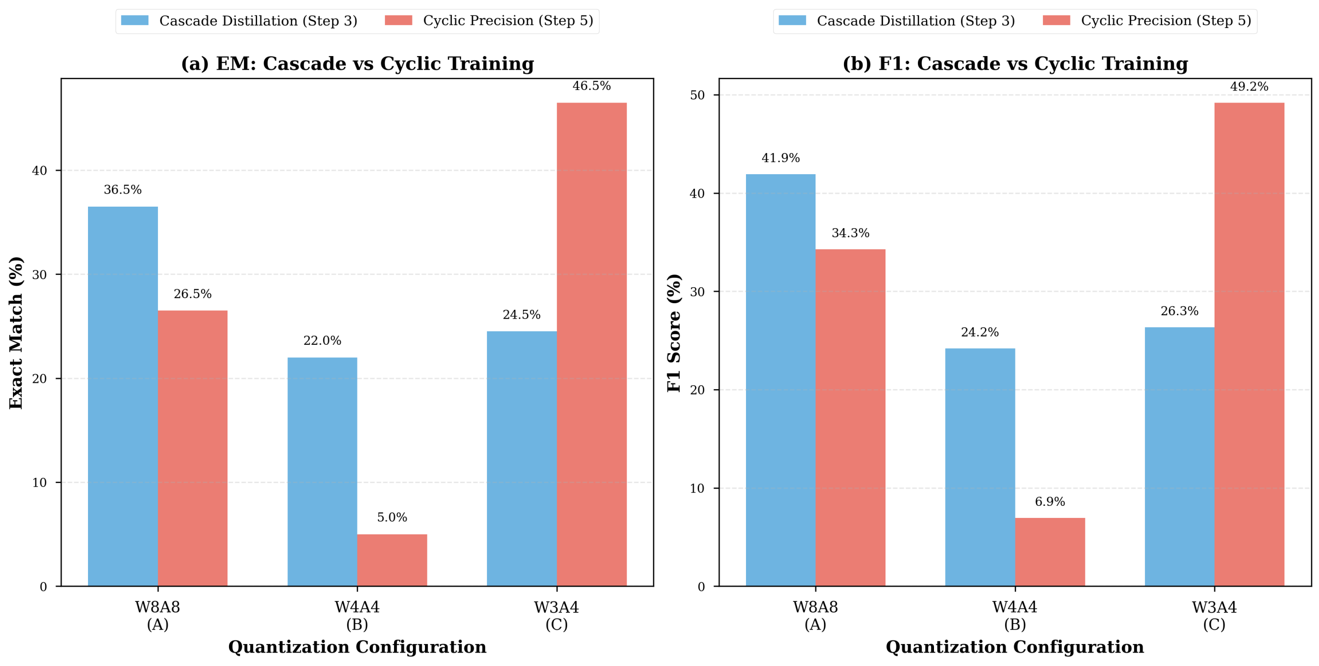
1. **Consistency Regularization:**
   - *Objective:* Minimize the KL divergence between the output distributions of different presets (e.g., $KL(P_{W8A8}||P_{W4A4})$).
   - *Benefit:* Ensures that lower-precision models mimic the behavior of the high-precision “teacher” more closely, smoothing the transition between bit-widths.
2. **Feature Alignment Loss:**
   - *Objective:* Minimize the Mean Squared Error (MSE) between the intermediate hidden states of different presets.
   - *Benefit:* Aligns the internal representations, not just the final output, ensuring that the model learns robust features that persist across quantization levels.
3. **Switchability Penalty:**
   - *Objective:* Add a penalty term based on the norm of the difference between LoRA adapters for different presets ($||W_{LoRA}{}^{A} - W_{LoRA}{}^{B}||$).
   - *Benefit:* Encourages the adapters to stay close to each other in parameter space, potentially reducing the “shock” to the model when switching presets at runtime.

[Step 5] Does this phenomenon align with the observations in CPT (ICLR'21)? If not, what could be the potential reasons?

No, my results do not align with the observations in the CPT paper.

In the CPT paper, Cyclic Precision Training improved or matched baseline performance. In my experiments, CPT caused severe degradation for presets A (W8A8) and B (W4A4), while massively boosting preset C (W3A4).

Potential Reasons for Divergence: 1. **Architecture Mismatch (CNN vs. Transformer):** CPT was designed for CNNs (ResNet). Transformers have different dynamics (LayerNorm vs. BatchNorm, global attention), which may be destabilized by the aggressive noise injection of cyclic precision. 2. **Insufficient Cycle Duration:** The original paper used long training cycles (thousands of steps). My implementation used short cycles (~31 steps) within a 1000-step budget. This rapid oscillation likely prevented the model from converging at any specific precision. 3. **LoRA Adapter Interference:** Unlike full fine-tuning, I trained precision-specific LoRA adapters. The cyclic schedule updates these adapters non-uniformly (spending more time at the "peaks" and "valleys" of the cosine schedule), leaving the middle preset (W4A4) undertrained.



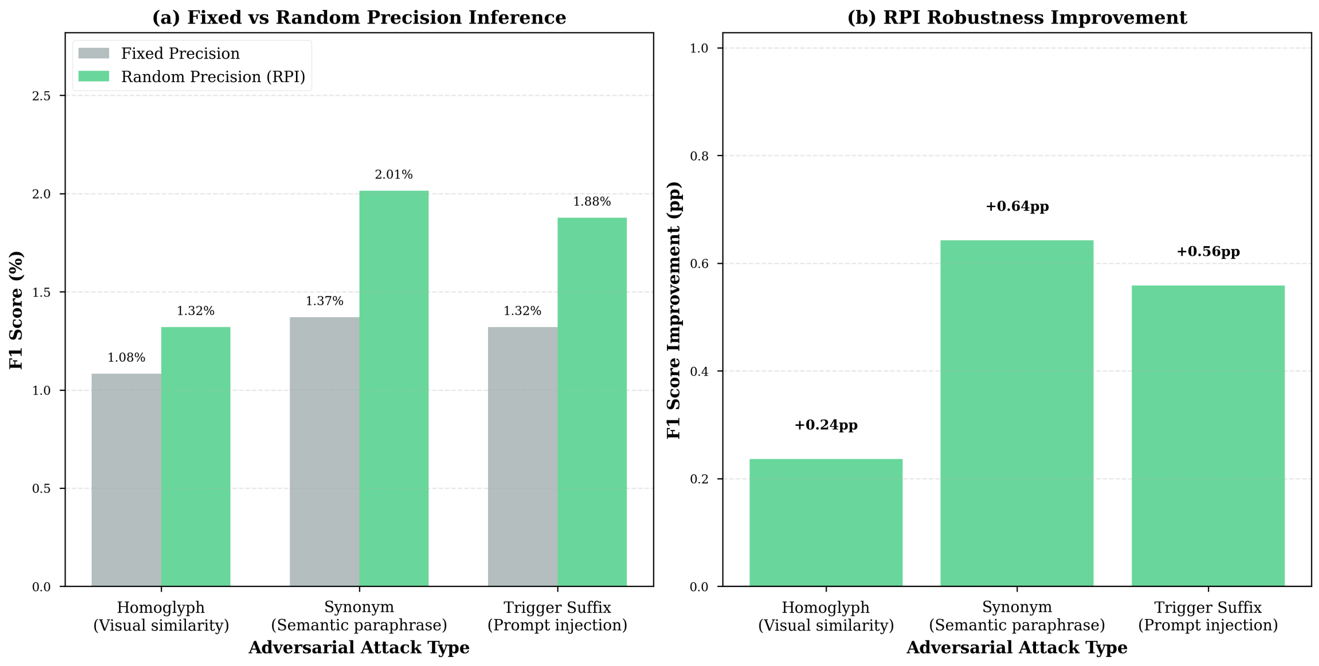
Step 5 Results

[Step 6] Does this phenomenon align with the observations in Double-Win Quant (ICML'21)? If not, what could be the potential reasons?

Qualitatively, yes; Quantitatively, no.

My results align with the *qualitative* finding that Random Precision Inference (RPI) improves adversarial robustness. However, the *quantitative* impact differs: I observed large relative gains (+38%) but on a very low absolute baseline (<2% F1), whereas the paper reported meaningful absolute gains on ImageNet.

Potential Reasons for Divergence: 1. **Task Complexity (QA vs. Classification):** SQuAD is a generative task with a vast output space. Adversarial perturbations in text (discrete token changes) behave differently than pixel noise. The base model's robustness on SQuAD is already near-zero, making "improvement" less meaningful. 2. **Attack Transferability:** Text adversarial attacks (like synonyms) rely on semantic preservation. Quantization alters the embedding space, breaking the specific transferability of these attacks, which explains why RPI helps. However, the attacks used may be too strong for the model to handle regardless of precision. 3. **Model Capacity:** GPT-2 (124M) is relatively small. The Double-Win Quant paper used ResNets which may have more redundant capacity to absorb quantization noise without collapsing on legitimate samples.



Step 6 Results

Based on your explorations of switchable and dynamic quantization, could you propose some promising research directions or questions for further integrating them with LLMs?

Based on this work, I propose the following research directions:

1. **Quantization-Aware Attention:** Standard quantization degrades attention mechanisms. Research into **mixed-precision attention** (e.g., keeping Key/Query at 8-bit while quantizing Value to 4-bit) could unlock efficiency without destroying context understanding.
2. **Prompt-Aware Dynamic Precision:** Different prompts require different levels of reasoning. Developing a lightweight **"precision router"** that predicts the necessary bit-width based on input complexity could optimize efficiency per-request.
3. **Adversarial Training with Random Precision (RPT):** RPI alone is a passive defense. Integrating random precision *during training* (RPT) specifically on adversarial examples could teach the model to be robust to both quantization noise and adversarial perturbations simultaneously.
4. **Hardware-Aware Co-Design:** Theoretical BitOPs often don't translate to wall-clock speedups due to memory bandwidth bottlenecks. Future work should focus on **co-designing quantization schemes with specific hardware accelerators** (e.g., optimizing for Tensor Core precision support) to realize actual latency gains.