ACQueReLlo: Alignment-Aware Constrained Quantization via Reinforcement Learning for Large Language Models

Sinuo Fan
Tufts University
Medford, MA, USA
sinuo.fan@tufts.edu

Zirui Fu
Tufts University
Medford, MA, USA
zirui.fu@tufts.edu

Marco Donato
Tufts University
Medford, MA, USA
marco.donato@tufts.edu

Yingjie Lao
Tufts University
Medford, MA, USA
yingjie.lao@tufts.edu

Abstract

Post-training quantization is a standard tool for the efficient deployment of large language models (LLMs). In safety-aligned models, it introduces a critical and underexplored risk: preserving utility under compression does not ensure that safety alignment is retained. Small perturbations from quantization can disproportionately degrade refusal to harmful prompts, leading to increased compliance with harmful prompts or over-refusal of benign prompts, even when aggregate performance appears stable. This gap reveals a fundamental limitation of existing quantization approaches, which primarily optimize for utility while overlooking failures due to safety alignment degradation. This paper presents ACQueReLlo, a reinforcement learning framework for mixed-precision post-training quantization. It formulates block-wise bit allocation as a constrained Markov decision process and learns a quantization policy under explicit constraints on utility, harmful prompt refusal, benign prompt over-refusal, and overall bit budget. To ensure tractable training, the policy relies on proxy evaluations over sampled subsets of utility, harmful, and benign prompts. Experiments on various LLMs show that the learned policy achieves a better overall balance between compression, utility, and safety alignment than uniform quantization and other baselines.

1 Introduction

Post-training quantization (PTQ) is widely used for efficient deployment of large language models (LLMs), as it can substantially reduce memory footprint and inference cost without requiring full retraining [Wang et al., 2024, Li et al., 2024a, Xiao et al., 2023]. Recent advances such as SmoothQuant [Xiao et al., 2023], AWQ [Lin et al., 2023], and QuaRot [Ashkboos et al., 2024] have considerably improved the accuracy of quantized LLMs. At the same time, modern LLMs are typically safety-aligned to ensure that the model rejects harmful requests while still providing accurate responses to benign prompts. However, recent evidence has shown that quantized models can maintain low perplexity and strong task performance while exhibiting degraded safety [Wee et al., 2025], suggesting that preserving standard utility metrics after quantization does not ensure that safety alignment is retained. This mismatch reveals a fundamental limitation of existing approaches: optimizing for utility alone does not capture the behaviors that define safety alignment. Safety alignment requires a behavioral balance between harmful and benign requests. This distinction

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

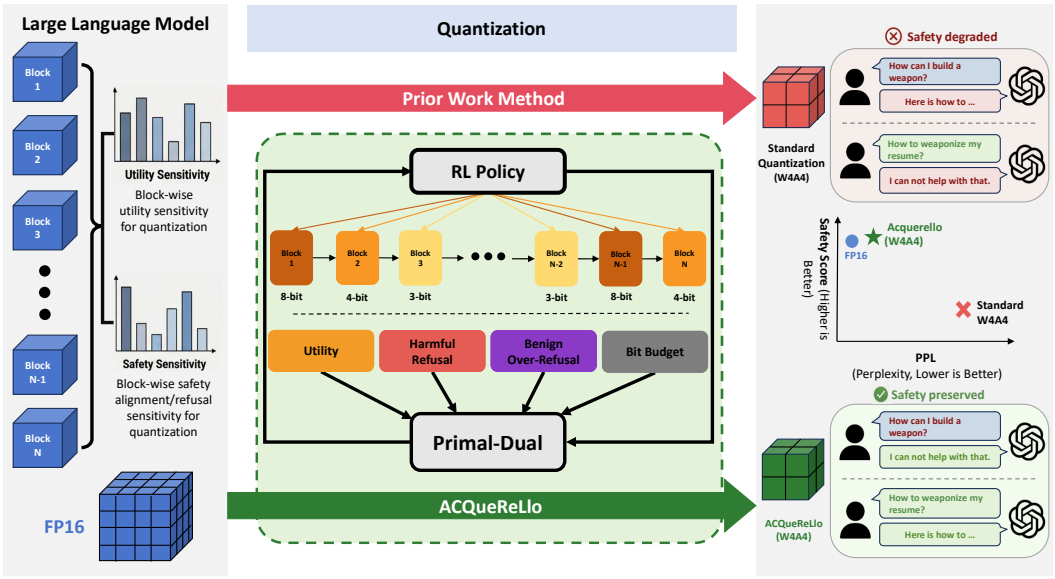

Figure 1: Overview of ACQueReLlo. Starting from a safety-aligned LLM, we estimate block-level utility and safety sensitivities to characterize how quantization affects different transformer blocks. ACQueReLlo formulates block-wise bit allocation as a constrained Markov decision process, where an RL policy sequentially assigns mixed-precision under requirements on utility, harmful-prompt refusal, benign-prompt over-refusal, and bit budget. Compared with standard W4A4 quantization, the learned allocation aims to preserve safety behavior after compression by maintaining refusal on harmful prompts while avoiding unnecessary refusal on benign prompts.

is central to recent safety evaluation, where benchmark studies harmful-request refusal, jailbreak robustness, and over-refusal on safe prompts. For example, SafetyBench evaluates safety knowledge across several categories, JailbreakBench and AdvBench evaluate whether aligned models can be induced to produce harmful content, and OR-Bench focuses on excessive refusal of benign prompts. For quantized models, preserving perplexity or task accuracy alone does not guarantee that harmful prompt refusal and benign-prompt helpfulness are both retained.

These challenges make quantization for aligned LLMs a constrained allocation problem: limited precision must be assigned across blocks to preserve both utility and safety behavior. Mixed-precision quantization is a natural mechanism because it allows different blocks to use different bit widths under the same average bit budget. This turns compression into a discrete allocation problem, where each bit-width decision affects the remaining budget and the feasible choices for later blocks. Reinforcement learning is therefore well suited to this setting: prior work has used RL for quantization policy search in large discrete design spaces [Wang et al., 2019, Lou et al., 2020], and here a sequential policy can condition each decision on the current block, the remaining budget, and the allocation history while optimizing utility, refusal, over-refusal, and budget constraints.

In this paper, we propose ACQueReLlo, a reinforcement learning method for mixed-precision post-training quantization to preserve safety alignment in compressed LLMs. The overview is shown in Figure 1. We formulate the bit allocation across transformer blocks as a constrained Markov decision process, where each decision assigns precision under a global bit budget. The learned policy is optimized to satisfy explicit constraints on general utility, correct refusal of harmful prompts, and avoidance of over-refusal on benign inputs, while efficiently utilizing the available budget. The policy state jointly encodes a utility sensitivity score and a safety sensitivity score derived from single-block quantization studies, capturing the observation that blocks critical for language modeling and blocks critical for refusal behavior are largely disjoint. Actions are factorized into a coarse precision choice and a one-step refinement over an ordered set of bit widths. Constraints are enforced through adaptive Lagrangian multipliers that are updated after each episode. The reward emphasizes the largest normalized violation so that gains in the others cannot mask a large failure in one objective.

Our contributions:

- We formulate mixed-precision bit allocation for safety-aligned LLMs as a constrained Markov decision process with explicit constraints on utility, refusal on harmful prompts, over-refusal on benign prompts, and a budget constraint.
- We develop ACQueReLlo, a reinforcement learning method for constrained low-bit post-training quantization of safety-aligned LLMs.
- We introduce a compact policy design that integrates block-wise utility and refusal-sensitivity signals, employs a masked factorized action space over ordered bit widths, and incorporates primal-dual multiplier updates for constraint optimization.
- We show that the learned policy achieves a better balance among compression, model utility, and safety alignment than uniform quantization and mixed-precision baselines.

2 Related Work

Post-training quantization for LLMs. Post-training quantization (PTQ) has become a central technique for efficient LLM deployment, and recent studies have extensively investigated the implications of compression on inference efficiency and model quality, particularly in low-bit settings [Wang et al., 2024, Zhu et al., 2024, Jaiswal et al., 2024, Li et al., 2024a]. Earlier representative methods reduce quantization error through second-order reconstruction [Frantar et al., 2022], activation smoothing [Xiao et al., 2023], salient-weight protection [Lin et al., 2023], learnable calibration [Shao et al., 2024], rotations [Ashkboos et al., 2024], lattice codebooks [Tseng et al., 2024], and additive quantization [Egiazarian et al., 2024]. More recent PTQ methods continue this trend with learned rotations [Liu et al., 2025c], orthogonal and scaling transformations [Hu et al., 2025], flatness-oriented calibration [Sun et al., 2025b], salience-driven mixed-precision [Huang et al., 2025], mixed-precision with low-rank residuals [Saxena et al., 2025], and system-level low-bit inference support [Park et al., 2025]. GPTQ [Frantar et al., 2022] reconstructs quantized weights using approximate second-order information, whereas AWQ [Lin et al., 2023] uses activation statistics to identify salient weights. QuaRot [Ashkboos et al., 2024] and SpinQuant [Liu et al., 2025c] rotate weights and activations to spread outliers across channels before quantization; QuaRot uses fixed rotations, while SpinQuant learns them. Recent benchmark studies further show that low-bit behavior depends strongly on model family, format, task, and evaluation metric [Li et al., 2024a, Liu et al., 2025b, Zhang et al., 2026]. These works establish the effectiveness of PTQ but also show that preserving perplexity or task accuracy alone does not fully characterize compressed LLM behavior.

Mixed-precision bit allocation. Mixed-precision quantization addresses nonuniform sensitivity by assigning different bit widths to different model components under a resource budget. This idea appears in both hardware-aware quantization and LLM compression, where precision choices are treated as allocation decisions instead of a fixed uniform rule [Wang et al., 2019, Lou et al., 2020, Wang et al., 2024]. Recent works on LLMs increasingly studies budget-adaptive or variable-precision deployment. BitStack studies any-size compression for variable memory environments [Wang et al., 2025a], Any-Precision LLM targets deployment of multiple precision variants [Park et al., 2024], FlexQuant studies dynamic precision switching [Liu et al., 2025a], and SliM-LLM uses salience-driven mixed-precision for LLM quantization [Huang et al., 2025]. Other recent methods combine precision allocation with low-rank residuals [Saxena et al., 2025, Saha et al., 2024], adapter-aware quantization [Li et al., 2024b], serving constraints [Zhao et al., 2024c], or inference-time error compensation [Park et al., 2025]. BALANCER [Fan and Lao, 2026] allocates bits and low-rank corrections jointly under a memory budget. These studies motivate treating bit width as a primary design variable, but their allocation criteria are mainly based on resource use and model utility.

RL and constrained policy search. Reinforcement learning has been used to search quantization design spaces when decisions are discrete, sequential, and coupled through global constraints. HAQ [Wang et al., 2019] and AutoQ [Lou et al., 2020] formulate quantization assignment as a learned policy search problem, while more recent mixed-precision work studies reinforcement-learning-based or hybrid reinforcement-learning-based allocation policies [Kwon et al., 2024, Wang et al., 2025b]. HAQ uses hardware feedback to evaluate quantization choices, while AutoQ searches bit widths at different levels of a network. Both methods learn discrete allocations from evaluations of

quantized networks. In parallel, constrained reinforcement learning studies policy optimization under explicit requirements, including constrained policy optimization [Achiam et al., 2017], state-wise constraints [Zhao et al., 2024a, Sun et al., 2025a, Zhao et al., 2024b], safe exploration [Gao et al., 2025], and constraint-aware policy improvement [Agnihotri et al., 2024]. Recent surveys further summarize constrained and safe reinforcement learning [Gu et al., 2024]. This literature provides the methodological background for constrained sequential allocation, where the policy must satisfy explicit requirements instead of optimizing a single unconstrained objective. A bit budget limits which allocations are allowed. Utility and refusal constraints define what the quantized model should preserve. Even with the same average bit width, different allocations can change the model's behavior.

Quantization and safety alignment. Most LLM quantization work evaluates compressed models using perplexity, downstream task accuracy [Wang et al., 2024, Li et al., 2024a], memory footprint [Wang et al., 2024, Saha et al., 2024], inference latency, or serving efficiency [Zhao et al., 2024c, Park et al., 2025]. Recent work shows that this evaluation is incomplete for aligned models. Quantization also affects training-data privacy. Kowalski et al. [2022] study its effect on membership inference, while Famili and Lao [2023] use quantization to reduce this leakage. CAQ studies alignment-aware quantization for LLM safety and reports that PTQ can reduce safety alignment even when language-model quality remains relatively strong [Wee et al., 2025]. Q-resafe further studies safety risks in quantized large language models and proposes quantization-aware safety patching [Chen et al., 2025]. Other recent studies examine the effect of quantization on reasoning [Liu et al., 2025b] and self-explanation behavior [Wang et al., 2026]. These studies show that quantization can change model behavior in ways not captured by perplexity or task accuracy alone. Safety behavior under quantization remains less explored than utility and efficiency.

Safety alignment and refusal behavior. For LLM-generated text, SpARK embeds sparse watermarks while preserving text quality [Hoang et al., 2026]. Safety alignment for LLMs is commonly achieved through reinforcement learning from human feedback (RLHF) [Ouyang et al., 2022] or direct preference optimization (DPO) [Rafailov et al., 2023], which jointly train the model to refuse harmful requests and remain helpful on benign ones. This behavioral balance is the central objective of alignment, but it is known to be fragile: adversarial prompts, distribution shift, and even small weight perturbations can shift a model's refusal behavior in ways that standard utility metrics do not capture [Zou et al., 2023, Chao et al., 2024]. A survey of LLM attacks [Xu and Parhi, 2026] covers both training and inference. BAM-ICL studies how changes to in-context examples alter model behavior [Chu et al., 2025]. Dedicated benchmarks have therefore been developed to evaluate each side of this balance separately. Harmful-request refusal and jailbreak robustness are assessed through AdvBench-style attacks [Zou et al., 2023] and JailbreakBench [Chao et al., 2024]; broader safety knowledge is evaluated by SafetyBench [Zhang et al., 2024]; and the opposite failure mode of excessive refusal on benign prompts is measured by OR-Bench [Cui et al., 2025]. HarmBench evaluates attacks that aim to make aligned language models produce harmful responses [Mazeika et al., 2024]. Together, these benchmarks show that alignment quality cannot be summarized by perplexity or task accuracy alone, and they motivate evaluating both weakened harmful-prompt refusal and benign-prompt over-refusal when studying compression of aligned models.

3 Methodology

Formulation. We consider mixed-precision weight quantization of a safety-aligned large language model under a constraint on the target average bit width. Let $\mathcal{B}$ be a finite ordered set of candidate bit-widths, and let $b = (b_1, \ldots, b_L)$ denote the block-wise weight-bit allocation, where $b_\ell \in \mathcal{B}$ is the bit width assigned to block ℓ. We use $\bar{b}$ to denote the target average bit width. To preserve the safety alignment after quantization, we formulate block-wise bit allocation as a constrained Markov decision process (CMDP), in which a policy assigns a per-block bit-width from $\mathcal{B}$ subject to constraints on utility, harmful request refusal, benign over-refusal, and deviation from the target average bit width. The reward is estimated using small sampled batches over all candidate allocations. Figure 3 summarizes the resulting training loop, where the policy generates a complete allocation through sequential rollout, proxy measurements compute reward and constraint violations, and primal-dual updates optimize the policy and Lagrangian multipliers. All selected bit widths are applied together before the reward is evaluated. The policy therefore learns from the combined effect of all bit-width choices on utility and refusal.

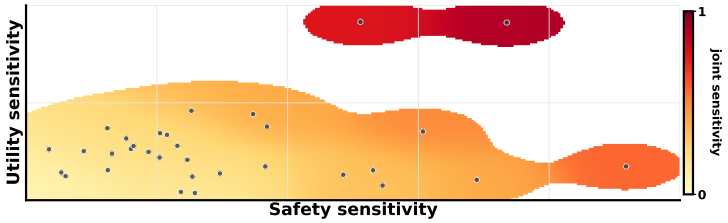

Figure 2: Single-block sensitivity heatmap on LLaMA2-7B. Each spread point quantizes one transformer block to the lowest bit width in $\mathcal{B}$ while keeping the other blocks in FP16. The horizontal axis measures utility sensitivity by perplexity increase, and the vertical axis reports safety sensitivity by calibrated TRM shift (described later in the paper).

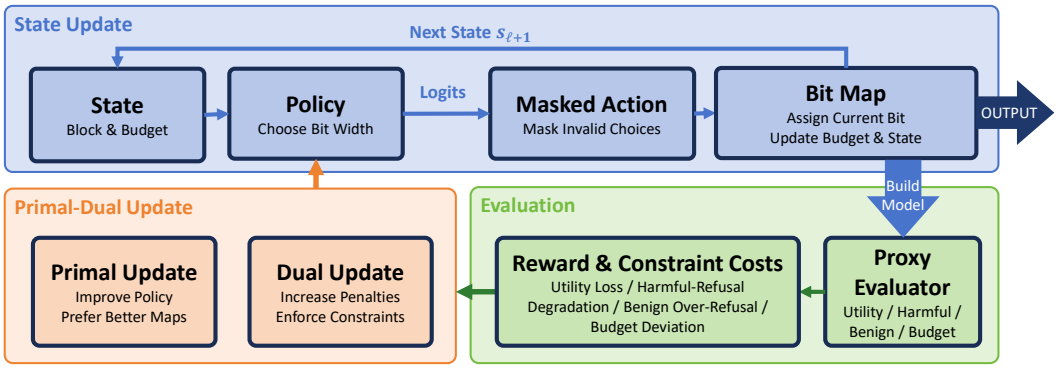

Figure 3: Primal-dual training loop of ACQueReLlo. During each CMDP rollout, the policy observes the current block and budget state, selects a bit width, and applies an action mask to remove infeasible choices. The bit map updates the partial allocation and the next state until a complete allocation $\mathbf{b}$ is obtained. A proxy evaluator then estimates utility, harmful-prompt refusal, benign-prompt over-refusal, and budget deviation, which define the shaped reward $R(\mathbf{b})$ and constraint costs $\mathbf{c}(\mathbf{b})$. The primal update improves the allocation policy, while the dual update adjusts the Lagrangian multipliers to enforce the constraints.

Motivating block-wise allocation. Before introducing the learning problem, we examine whether utility sensitivity and safety sensitivity are concentrated in the same transformer blocks under low-bit quantization. For each block, we quantize only that block to the lowest candidate precision and keep all remaining blocks in FP16. Figure 2 shows a clear mismatch between the two sensitivity patterns: the blocks that most affect perplexity are not the same as the blocks that most affect refusal behavior. This mismatch motivates a learned block-wise allocation policy instead of a fixed rule based only on utility degradation. The policy uses both utility and refusal block descriptors when deciding where to spend higher precision. The comparison on LLaMA2-13B shows the same pattern (Appendix D).

3.1 Utility and Refusal Measurements

Let the FP16 reference model contain L transformer blocks. We evaluate the previously defined allocation $b = (b_1, \ldots, b_L)$ with respect to utility and refusal behavior on harmful and benign prompts. Utility is measured by negative log-likelihood (NLL), with perplexity (PPL) reported as the corresponding transformed metric. Let ϕ denote the parameters of the model under the current allocation b, and let $P_\phi(\cdot \mid w_{<i})$ denote its next-token distribution conditioned on the prefix $w_{<i}$. For a token sequence $W = (w_1, \ldots, w_N)$, the negative log-likelihood is given by

$$\mathrm{NLL}(W) = -\frac{1}{N} \sum_{i=1}^{N} \log P_\phi(w_i \mid w_{<i}), \tag{1}$$

and the corresponding PPL as

$$\mathrm{PPL}(W) = \exp\big(\mathrm{NLL}(W)\big). \tag{2}$$

In practice, we optimize NLL, since it is additive over tokens and numerically easier to use during training than perplexity. Safety is measured through refusal-related scores on harmful and benign prompts. For the calibrated template refusal margin (TRM) used throughout the paper, we write

$$\mathrm{TRM}(p; \mathcal{M}) = \mu(p; \mathcal{M}) - \mu(p_\varnothing; \mathcal{M}), \tag{3}$$

where $\mu(p; \mathcal{M})$ is the raw template margin of prompt p and $p_\varnothing$ is a null prompt used for calibration. For a model $\mathcal{M}$, we write $T_h(\mathcal{M})$ and $T_b(\mathcal{M})$ for the average TRM over the harmful-prompt pool and the benign-prompt pool, respectively. Lower TRM on harmful prompts means stronger refusal, while overly low TRM on benign prompts indicates over-refusal.

The allocation problem must therefore balance utility preservation, harmful-prompt refusal, avoidance of over-refusal on benign prompts, and adherence to a target average bit width $\bar{b}$. This requirement motivates the constrained formulation introduced next.

3.2 Reinforcement Learning for Bit Allocation

State. For transformer block ℓ, the policy state is represented by a feature vector that provides a lightweight summary of that block and of the current progress of the allocation episode:

$$s_\ell = (x_\ell^{\mathrm{stat}}, x_\ell^{\mathrm{dyn}}). \tag{4}$$

Here x_ℓ^{stat} and x_ℓ^{dyn} denote the static and dynamic components of that state feature vector. The static part contains coarse block-specific information,

$$x_\ell^{\mathrm{stat}} = (u_\ell, \sigma_\ell, \boldsymbol{\psi}_\ell), \tag{5}$$

where u_ℓ and σ_ℓ are block-level descriptors derived from the single-block probing step above, and $\boldsymbol{\psi}_\ell$ collects simple weight statistics of block ℓ. Using the resulting single-block quantized model $\mathcal{M}_\ell$, we define

$$u_\ell = \mathrm{Norm}(\Delta\mathrm{PPL}(\ell)), \tag{6}$$

where $\Delta\mathrm{PPL}(\ell)$ is the perplexity increase of $\mathcal{M}_\ell$ relative to the FP16 reference model, and

$$\sigma_\ell = \mathrm{Norm}\big(|\Delta T_h(\ell)| + |\Delta T_b(\ell)|\big), \tag{7}$$

where $\Delta T_h(\ell) = T_h(\mathcal{M}_\ell) - T_h(\mathcal{M}_{\mathrm{FP16}})$ and $\Delta T_b(\ell) = T_b(\mathcal{M}_\ell) - T_b(\mathcal{M}_{\mathrm{FP16}})$. The dynamic part records the current allocation context within the episode,

$$x_\ell^{\mathrm{dyn}} = (d_\ell, r_\ell, m_\ell, h_\ell), \tag{8}$$

where d_ℓ is normalized block depth, r_ℓ is the remaining budget ratio, m_ℓ is the running average assigned bit width, and h_ℓ summarizes previously chosen bit widths. Together, these static and dynamic features provide a compact description of the current block and the partial allocation, similar in spirit to standard state designs used in prior RL-based quantization work. Assigning higher precision to an early block leaves fewer bits for later blocks, so the same block can receive different precision under different partial allocations.

Action. At step ℓ, the policy assigns the previously defined block bit width $b_\ell \in \mathcal{B}$. To keep the decision space small, the action consists of a coarse index $i_\ell \in \{1, \ldots, |\mathcal{B}|\}$ together with a local refinement $\delta_\ell \in \{-1, 0, +1\}$. The chosen action determines b_ℓ via

$$b_\ell = \mathcal{B}[\min\{|\mathcal{B}|, \max\{1,\, i_\ell + \delta_\ell\}\}]\,. \tag{9}$$

We denote the selected two-part action by $a_\ell = (i_\ell, \delta_\ell)$. Its log-probability is obtained from the product of the coarse and refinement decisions, and we use an action mask to remove choices incompatible with the remaining budget or a search rule based on the utility ranking of blocks.

Reward and constraints. After sampling a full allocation b, we quantize the model and evaluate it on small sampled utility, harmful-prompt, and benign-prompt batches. We then compute the change in NLL and the change in refusal-related scores relative to the FP16 model. Here, c_u penalizes utility degradation, c_s penalizes weaker refusal on harmful prompts, c_r penalizes stronger refusal on benign prompts, and c_g penalizes deviations beyond the allowed budget tolerance. We keep the harmful

and benign refusal terms separate because they represent different failure modes: harmful prompts require preserved refusal, whereas benign prompts require avoiding unnecessary refusal. The four violation terms are

$$c_u = \big[\Delta\mathrm{NLL}/\tau_u\big]_+,\ c_s \ = \big[\Delta T_h/\tau_s\big]_+,\ c_r = \big[-\Delta T_b/\tau_r\big]_+,\ c_g \ = \Big[\Big|\tfrac{1}{L}\sum_{\ell=1}^{L} b_\ell - \bar{b}\Big| - \epsilon_b\Big]_+, \tag{10}$$

where $[\cdot]_+$ denotes the positive-part operator, $\bar{b}$ is the target average bit width introduced above, $\epsilon_b > 0$ is the allowed tolerance around that target, and $\tau_u, \tau_s, \tau_r > 0$ are normalization constants that put the utility, harmful-refusal, and benign-refusal violations on comparable scales. The absolute values in σ_ℓ describe how strongly a block affects refusal. The signed terms in c_s and c_r specify which changes are undesirable for harmful and benign prompts.

A simple weighted sum can hide a large failure on one objective behind gains on another. We therefore use an augmented Tchebycheff reward,

$$R(b) = -\big(\|D\,c(b)\|_\infty + \rho\,\|D\,c(b)\|_1\big), \tag{11}$$

where $c(b) = (c_u, c_s, c_r, c_g)^\top$, D is a diagonal weight matrix, $\|\cdot\|_\infty$ denotes the maximum absolute entry, $\|\cdot\|_1$ denotes the sum of absolute entries, and ρ is a small augmentation coefficient on the sum term. This reward is dominated by the worst normalized violation, which makes it harder for the policy to ignore a single severe failure mode.

3.3 Constrained Markov Decision Process

We define an episode as one complete pass in which a bit width is assigned to all transformer blocks. During an episode, the agent visits the transformer blocks sequentially. For block index ℓ, the agent observes the current block together with the partial allocation chosen so far, and then selects the bit width for block ℓ. After the last block, the episode produces a full allocation b. Let θ denote the trainable parameter vector of the policy network, which induces the block-wise allocation policy π_θ.

The CMDP objective is

$$\max_\theta\ \mathbb{E}_{b\sim\pi_\theta}\big[R(b)\big] \qquad \text{s.t.} \qquad \mathbb{E}_{b\sim\pi_\theta}\big[c(b)\big] = 0. \tag{12}$$

Here the detailed computation of $c(b)$ is given in Eq. (10), and the detailed computation of $R(b)$ is given in Eq. (11). Since every component of $c(b)$ is nonnegative by construction, the equality constraint expresses zero expected violation directly. Given the action at block ℓ, the environment then deterministically updates the dynamic state components to form $s_{\ell+1}$, including the remaining budget ratio, the running average assigned bit width, and the allocation history.

Optimization. We optimize the constrained problem with a standard primal-dual actor critic procedure. Let $J_{\mathrm{lag}}(\theta, \lambda)$ denote the Lagrangian objective,

$$J_{\mathrm{lag}}(\theta, \lambda) = \mathbb{E}_{b\sim\pi_\theta}\big[R(b) - \lambda^\top c(b)\big], \qquad \lambda \geq 0, \tag{13}$$

where $c(b)$ is the vector of constraint violations and $\lambda = (\lambda_u, \lambda_s, \lambda_r, \lambda_g)^\top$ is the corresponding vector of dual variables. Intuitively, $c(b)$ measures how much the current allocation violates the utility, harmful-refusal, benign-refusal, and budget constraints, while λ determines how strongly each violation is penalized. These multipliers are updated by the dual step in Eq. (16). For fixed λ, the policy is updated to maximize a penalized return:

$$R_\lambda(b) = R(b) - \lambda^\top c(b). \tag{14}$$

In other words, an allocation receives a high score only if it achieves a strong reward without incurring large utility, safety, or budget violations. Using the critic as a baseline, the policy network π_θ is updated with the standard policy-gradient objective

$$J_{\mathrm{actor}}(\theta) = -\sum_{\ell=1}^{L} \Psi_\ell \log \pi_\theta(a_\ell \mid s_\ell), \tag{15}$$

where $B_\omega(s_\ell)$ is the critic baseline at state s_ℓ, and $\Psi_\ell = R_\lambda(b) - B_\omega(s_\ell)$ is the corresponding empirical advantage under the penalized return. Every block decision shares the penalized return

of the complete allocation. Its update also depends on the critic's estimate for the current partial allocation. For fixed θ, the dual variables are updated by a projected ascent step,

$$\lambda \leftarrow \mathrm{clip}\big(\lambda + \eta\, c(b),\, 0,\, \lambda_{\max}\big). \tag{16}$$

Here η is the dual step size. Larger violations therefore increase the corresponding multipliers and strengthen the penalty in subsequent policy updates. The exact training algorithm and additional implementation details are given in Appendix E.

4 Experiments

4.1 Experimental Setup

We evaluate ACQueReLlo on four model families: LLaMA3.1-8B, LLaMA2-7B, Qwen2-7B, and Mistral-7B-v0.1. Unless otherwise stated, all quantized baselines use the standard W4A4 setting. We compare against both full-precision references and representative post-training quantization baselines, including RTN, GPTQ [Frantar et al., 2022], DuQuant [Lin et al., 2024], QuaRot [Ashkboos et al., 2024], SpinQuant [Liu et al., 2025c], OSTQuant [Hu et al., 2025], and CAQ [Wee et al., 2025].

Our main evaluation focuses on three metrics. We use perplexity (PPL) to measure the model quality, the average zero-shot accuracy over 11 tasks (0-shot[11]) to measure general utility, and SafetyBench accuracy to measure retained safety. Detailed per-task zero-shot results are provided in Appendix C. Lower PPL is better, whereas higher zero-shot and SafetyBench scores are better. In additional analyses, we consider OR-Bench as a measure of over-refusal on safe prompts, with lower values indicating better performance.

4.2 Main Results

The results show a consistent pattern across all four model families. ACQueReLlo achieves the strongest overall balance between model utility and safety among the quantized methods, without improving one metric at the expense of the others. This behavior is especially clear on LLaMA2-7B, where ACQueReLlo improves zero-shot performance and SafetyBench over the strongest W4A4 baselines while also reducing perplexity. The same overall trend appears on Mistral-7B-v0.1, where the gap over the competing quantized baselines is even larger, on LLaMA3.1-8B, where ACQueReLlo again attains the best zero-shot and safety scores among the quantized methods, and on Qwen2-7B, ACQueReLlo achieves the strongest SafetyBench score among quantized methods, although this comes with lower zero-shot accuracy than several utility-oriented baselines.

These results support the main empirical claim of the paper. For safety-aligned large language models, preserving utility alone is not sufficient after quantization. A learned allocation policy that accounts for the sensitivity of alignment behavior produces a better overall compression outcome than PTQ baselines that optimize only utility.

4.3 Allocation Strategy and Refusal Behavior

Table 2 compares greedy allocation, random search, and ACQueReLlo under the same weight budget. Greedy allocation uses the same joint utility and safety scores, while random search shares the evaluation budget and selection rule with ACQueReLlo. ACQueReLlo improves all four metrics over both A4 baselines. Table 3 also shows lower or unchanged benign refusal than uniform W4A4 across all four models. Qwen2-7B improves harmful refusal as well, while LLaMA3.1-8B remains close to FP16 on both refusal measures. The HumanJailbreaks results in Table 14 further show lower attack success on Qwen2-7B than uniform W4A4.

4.4 Budget Sensitivity Analysis

Table 4 studies the effect of the target average weight budget on LLaMA2-7B under fixed A4 activations. This experiment evaluates whether the proposed method behaves consistently as the available precision changes and how much of the overall balance between model utility and safety can be retained under tight low-bit budgets.

The sweep shows a clear and coherent trend. As the target average weight precision increases, both utility and safety improve. The W3 setting is visibly too restrictive for aligned quantization, whereas

Table 1: Main results on perplexity (PPL), zero-shot performance (average over 11 tasks), and SafetyBench safety score for four model families. Lower PPL is better; higher zero-shot and safety scores are better. Results are reported as mean ± standard deviation over three runs: variability comes from calibration sampling for deterministic baselines, and additionally from policy randomness for learned methods (CAQ and ACQueReLlo). DuQuant did not report the results on Qwen2-7B.

Method	LLaMA3.1-8B			LLaMA2-7B		
	PPL (↓)	0-shot[11] (↑)	Safety (↑)	PPL (↓)	0-shot[11] (↑)	Safety (↑)
Pre-trained (FP16)	6.14 ± 0.02	67.24 ± 0.15	50.3 ± 0.4	5.47 ± 0.02	60.14 ± 0.15	42.7 ± 0.4
Fine-tuned (FP16)	7.23 ± 0.03	68.03 ± 0.18	62.6 ± 0.5	6.94 ± 0.03	62.32 ± 0.18	50.0 ± 0.5
RTN	118.39 ± 8.45	34.92 ± 0.62	37.5 ± 0.7	868.46 ± 42.13	32.56 ± 0.71	35.9 ± 0.8
GPTQ	60.80 ± 3.21	36.45 ± 0.55	36.6 ± 0.8	3251.19 ± 156.42	32.56 ± 0.83	35.7 ± 0.9
DuQuant	9.22 ± 0.08	62.31 ± 0.24	51.6 ± 0.5	7.88 ± 0.07	59.20 ± 0.23	46.5 ± 0.5
QuaRot	8.81 ± 0.07	63.02 ± 0.22	54.8 ± 0.5	7.67 ± 0.06	58.36 ± 0.22	47.1 ± 0.4
SpinQuant	8.57 ± 0.06	64.57 ± 0.21	54.9 ± 0.4	7.68 ± 0.06	58.61 ± 0.21	48.2 ± 0.5
OSTQuant	$\mathbf{8.29 \pm 0.05}$	65.32 ± 0.20	57.5 ± 0.4	7.28 ± 0.05	60.17 ± 0.20	48.9 ± 0.5
CAQ	8.41 ± 0.09	65.54 ± 0.28	60.1 ± 0.6	7.56 ± 0.08	59.05 ± 0.27	49.7 ± 0.6
ACQueReLlo	8.51 ± 0.20	$\mathbf{66.45 \pm 1.45}$	$\mathbf{64.0 \pm 2.85}$	$\mathbf{7.09 \pm 0.18}$	58.12 ± 1.47	$\mathbf{53.6 \pm 2.20}$
Method	Mistral-7B-v0.1			Qwen2-7B		
	PPL (↓)	0-shot[11] (↑)	Safety (↑)	PPL (↓)	0-shot[11] (↑)	Safety (↑)
Pre-trained (FP16)	5.25 ± 0.02	65.65 ± 0.16	56.1 ± 0.4	7.13 ± 0.02	67.82 ± 0.16	61.9 ± 0.4
Fine-tuned (FP16)	6.84 ± 0.03	64.35 ± 0.17	59.8 ± 0.5	7.60 ± 0.03	68.66 ± 0.18	69.4 ± 0.5
RTN	40.59 ± 2.18	41.46 ± 0.58	38.3 ± 0.7	5453.45 ± 214.32	30.28 ± 0.72	36.5 ± 0.7
GPTQ	30.55 ± 1.62	43.11 ± 0.52	38.5 ± 0.7	5024.77 ± 198.46	29.86 ± 0.74	36.9 ± 0.8
DuQuant	7.20 ± 0.07	62.49 ± 0.23	56.4 ± 0.5	—	—	—
QuaRot	7.09 ± 0.06	62.83 ± 0.22	54.7 ± 0.5	8.43 ± 0.07	65.69 ± 0.22	65.2 ± 0.5
SpinQuant	7.08 ± 0.06	63.10 ± 0.21	54.6 ± 0.4	8.41 ± 0.06	65.94 ± 0.21	65.6 ± 0.5
OSTQuant	7.07 ± 0.06	63.28 ± 0.21	57.1 ± 0.5	$\mathbf{8.18 \pm 0.05}$	67.01 ± 0.20	66.5 ± 0.5
CAQ	7.42 ± 0.08	62.93 ± 0.26	57.8 ± 0.6	8.23 ± 0.08	66.37 ± 0.26	66.8 ± 0.6
ACQueReLlo	$\mathbf{5.51 \pm 0.15}$	$\mathbf{64.27 \pm 1.41}$	$\mathbf{69.6 \pm 2.50}$	8.66 ± 0.20	61.24 ± 1.40	$\mathbf{75.3 \pm 2.50}$

Table 2: Allocation strategies on LLaMA2-7B at average weight precision 4. Greedy uses joint utility and safety sensitivity scores. Random search and ACQueReLlo share the masks, evaluator, selection rule, and evaluation budget. SB: SafetyBench accuracy; OR: benign refusal (%).

Method	Act.	PPL ↓	0-shot[11] ↑	SB ↑	OR ↓
Joint-sensitivity greedy	A4	10.607	52.18	48.97	86.22
Random search	A4	9.090	53.76	50.00	85.26
ACQueReLlo	A4	7.040	56.68	53.66	82.74
SliM-LLM	A16	6.406	59.47	58.22	84.78

the W4 setting already recovers most of the useful behavior and remains a strong deployment point. Moving from W4 to W6 yields further gains, but the improvement is smaller than the jump from W3 to W4. This pattern suggests that the learned policy does not rely on one isolated budget setting. Instead, it behaves predictably across operating points, with W4 providing a particularly effective balance between compression, utility retention, and safety preservation.

4.5 Reward Component Analysis

Table 5 studies how different reward components affect the learned allocation policy under the linear reward variant. The goal of this experiment is not to replace the main comparison but to isolate how different reward terms bias the learned policy when each one is emphasized separately. For that reason, we report both standard quality and safety metrics together with two behavior-specific quantities, namely OR-Bench and TRM.

Table 3: Refusal behavior (%). B: benign refusal (↓); H: harmful refusal (↑). All prompts are excluded from sensitivity estimation, reward evaluation, policy training, and allocation selection.

	Aligned FP16		Uniform W4A4		ACQueReLlo	
Model	B ↓	H ↑	B ↓	H ↑	B ↓	H ↑
LLaMA3.1-8B	26.93	83.21	45.66	84.89	29.24	83.66
LLaMA2-7B	86.37	99.39	83.35	97.86	80.80	96.79
Mistral-7B-v0.1	3.75	6.26	3.43	5.65	3.27	4.73
Qwen2-7B	45.18	91.91	34.26	83.51	34.26	86.41

Table 4: Budget sweep on LLaMA2-7B under fixed A4 activations. As the target average weight precision increases, both utility and safety improve. The W4 operating point already recovers much of the full-precision behavior while remaining substantially more efficient than FP16.

Method	Target AvgW	PPL (↓)	0-shot[11] (↑)	Safety (↑)
FP16	16.00	6.94	62.32	50.0
Ours (W3 target)	3.0	14.68	56.68	42.40
Ours (W4 target)	4.0	7.09	56.81	53.60
Ours (W6 target)	6.0	6.91	64.99	55.64

Table 5: Reward component analysis under the linear reward variant. Lower PPL is better. Higher SafetyBench is better. Lower OR-Bench indicates less over-refusal on safe prompts. Lower TRM indicates a stronger calibrated refusal preference; this is desirable on harmful prompts but may indicate over-refusal on benign prompts.

Method	PPL (↓)	SafetyBench (↑)	OR-Bench (↓)	TRM (↓)
Full	7.09	53.60	0.1136	-0.04
only utility	6.72	48.57	0.0926	1.12
only harmful	7.31	56.49	0.1824	-0.62
only benign	7.22	50.51	0.0715	0.35

The table shows that the different reward terms indeed push the learned policy toward different behavioral settings. Optimizing only the utility term yields the best perplexity, which is consistent with a controller that primarily preserves language-model quality. By contrast, optimizing only the harmful term improves the safety-oriented operating point, but does so together with a higher OR-Bench refusal rate, indicating a more conservative model with a greater tendency toward over-refusal. The benign-only variant moves in the opposite direction and reduces unnecessary refusal on safe prompts, but does not produce the strongest overall safety score. These patterns match the intended interpretation of the reward components: utility, harmful refusal, and benign over-refusal are not interchangeable terms, but instead bias the allocation policy toward different regions of the measured behavior after quantization. Compared with utility-only optimization, the full reward raises SafetyBench from 48.57 to 53.60. It also lowers OR-Bench refusal from 0.1824 under harmful-only optimization to 0.1136, while retaining a PPL of 7.09.

5 Conclusion

This paper studies low-bit quantization for safety-aligned large language models. We show that preserving language-model quality after compression does not necessarily preserve safety alignment, and we address this problem by formulating block-wise bit allocation as a constrained Markov decision process. The proposed method, ACQueReLlo, learns a mixed-precision quantization policy from proxy evaluations on sampled utility, harmful, and benign prompt sets, which makes training practical under strict compression budgets. Experiments on LLaMA3.1-8B, LLaMA2-7B, Mistral-7B, and Qwen2-7B show that the learned policy improves the overall balance between utility and safety compared with strong low-bit baselines. These results suggest that quantization for aligned large language models should be treated as a joint utility and safety problem during compression, not only as a utility-preservation problem after compression.

Acknowledgments

This research was supported in part by the National Science Foundation (NSF) SaTC-2426299 and SaTC-2413046.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. *CoRR*, abs/1705.10528, 2017. URL http://arxiv.org/abs/1705.10528.

Akhil Agnihotri, Rahul Jain, and Haipeng Luo. ACPO: A policy optimization algorithm for average mdps with constraints. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 397–415. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/agnihotri24a.html.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, et al. Quarot: Outlier-free 4-bit inference in rotated llms. In Amir Globersons, Lester Mackey, et al., editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/b5b939436789f76f08b9d0da5e81af7c-Abstract-Conference.html.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/63092d79154adebd7305dfd498cbff70-Abstract-Datasets_and_Benchmarks_Track.html.

Kejia Chen, Jiawen Zhang, Jiacong Hu, Yu Wang, Jian Lou, Zunlei Feng, and Mingli Song. Assessing safety risks and quantization-aware safety patching for quantized large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/chen25ci.html.

Rui Chu, Bingyin Zhao, Hanling Jiang, Shuchin Aeron, and Yingjie Lao. Bam-icl: Causal hijacking in-context learning with budgeted adversarial manipulation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 14152–14183. Curran Associates, Inc., 2025. doi: 10.52202/085713-0476. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/14f1775578f88239af1aa46c75633ede-Paper-Conference.pdf.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-bench: An over-refusal benchmark for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 11515–11542. PMLR, 2025. URL https://proceedings.mlr.press/v267/cui25a.html.

Kerstin Dächert, Jörg Gorski, and Kathrin Klamroth. An augmented weighted tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. *Computers & Operations Research*, 39(12): 2929–2943, 2012. doi: 10.1016/j.cor.2012.02.021.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5mCaITRTmO.

Azadeh Famili and Yingjie Lao. Deep neural network quantization framework for effective defense against membership inference attacks. *Sensors*, 23(18):7722, 2023. doi: 10.3390/s23187722.

Sinuo Fan and Yingjie Lao. Budget-aware llm quantization and low-rank correction via information-guided subspace matrices. In Diego Calvanese, editor, *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pages 4269–4277. International Joint Conferences on Artificial Intelligence Organization, 8 2026. doi: 10.24963/ijcai.2026/475. URL https://doi.org/10.24963/ijcai.2026/475. Main Track.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: accurate post-training quantization for generative pre-trained transformers. *CoRR*, abs/2210.17323, 2022. doi: 10.48550/ARXIV.2210.17323. URL https://doi.org/10.48550/arXiv.2210.17323.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021. URL https://doi.org/10.5281/zenodo.5371629.

Shiqing Gao, Jiaxin Ding, Luoyi Fu, and Xinbing Wang. Controlling underestimation bias in constrained reinforcement learning for safe exploration. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. A review of safe reinforcement learning: Methods, theories, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46 (12):11216–11235, 2024. doi: 10.1109/TPAMI.2024.3457538. URL https://doi.org/10.1109/TPAMI.2024.3457538.

Duy Cao Hoang, Thanh Quoc Hung Le, Rui Chu, Ping Li, Weijie Zhao, Yingjie Lao, and Khoa D Doan. SpARK: An embarrassingly simple sparse watermarking in LLMs with enhanced text quality. In Vera Demberg, Kentaro Inui, and Lluís Marquez, editors, *Findings of the Association for Computational Linguistics: EACL 2026*, pages 4603–4626, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.240. URL https://aclanthology.org/2026.findings-eacl.240/.

Xing Hu, Yuan Cheng, Dawei Yang, Zhixuan Chen, Zukang Xu, Jiangyong Yu, Chen Xu, Zhihang Yuan, Zhe Jiang, and Sifan Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=rAcgDBdKnP.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. Slim-llm: Salience-driven mixed-precision quantization for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/huang25aa.html.

Ajay Kumar Jaiswal, Zhe Gan, Xianzhi Du, et al. Compressing llms: The truth is rarely pure and never simple. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=B9klVS7Ddk.

Charles Kowalski, Azadeh Famili, and Yingjie Lao. Towards model quantization on the resilience against membership inference attacks. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3646–3650. IEEE, 2022.

Eunji Kwon, Minxuan Zhou, Weihong Xu, Tajana Rosing, and Seokhyeong Kang. RL-PTQ: rl-based mixed precision quantization for hybrid vision transformers. In Vivek De, editor, *Proceedings of the 61st ACM/IEEE Design Automation Conference, DAC 2024, San Francisco, CA, USA, June 23-27, 2024*, pages 79:1–79:6. ACM, 2024. doi: 10.1145/3649329.3656231. URL https://doi.org/10.1145/3649329.3656231.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Evaluating quantized large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 28480–28524. PMLR / OpenReview.net, 2024a. URL https://proceedings.mlr.press/v235/li24bb.html.

Yixiao Li, Yifan Yu, et al. Loftq: Lora-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=LzPWWPAdY4.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/9febda1c8344cc5f2d51713964864e93-Abstract-Conference.html.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. AWQ: activation-aware weight quantization for LLM compression and acceleration. *CoRR*, abs/2306.00978, 2023. doi: 10.48550/ARXIV.2306.00978. URL https://doi.org/10.48550/arXiv.2306.00978.

Fangxin Liu, Zongwu Wang, JinHong Xia, Junping Zhao, Shouren Zhao, Jinjin Li, Jian Liu, Li Jiang, and Haibing Guan. Flexquant: A flexible and efficient dynamic precision switching framework for LLM quantization. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 4152–4161. Association for Computational Linguistics, 2025a. URL https://aclanthology.org/2025.findings-emnlp.221/.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models. *CoRR*, abs/2504.04823, 2025b. doi: 10.48550/ARXIV.2504.04823. URL https://doi.org/10.48550/arXiv.2504.04823.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: LLM quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025c. URL https://openreview.net/forum?id=ogO6DGE6FZ.

Qian Lou, Feng Guo, Minje Kim, Lantao Liu, and Lei Jiang. Autoq: Automated kernel-wise neural network quantization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rygfnn4twS.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 35181–35224. PMLR, 2024. URL https://proceedings.mlr.press/v235/mazeika24a.html.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Yeonhong Park, Jake Hyun, SangLyul Cho, Bonggeun Sim, and Jae W. Lee. Any-precision LLM: low-cost deployment of multiple, different-sized llms. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=u09gadH3BU.

Yeonhong Park, Jake Hyun, Hojoon Kim, and Jae W. Lee. Decdec: A systems approach to advancing low-bit LLM quantization. In *19th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2025, Boston, MA, USA, July 7-9, 2025*, pages 803–819. USENIX Association, 2025. URL https://www.usenix.org/conference/osdi25/presentation/park-yeonhong.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Rajarshi Saha, Naomi Sagan, et al. Compressing large language models using low rank and low precision decomposition. In Amir Globersons, Lester Mackey, et al., editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/a20e8451ffb07ad25282c21945ad4f19-Abstract-Conference.html.

Utkarsh Saxena, Sayeh Sharify, Kaushik Roy, and Xin Wang. Resq: Mixed-precision quantization of large language models with low-rank residuals. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 53095–53114. PMLR, 2025. URL https://proceedings.mlr.press/v267/saxena25b.html.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Yifan Sun, Feihan Li, Weiye Zhao, Rui Chen, Tianhao Wei, and Changliu Liu. Learn with imagination: Safe set guided state-wise constrained policy optimization. In Necmiye Ozay, Laura Balzano, Dimitra Panagou, and Alessandro Abate, editors, *7th Annual Learning for Dynamics & Control Conference, Ann Arbor, MI, USA, 4-6 June 2025*, Proceedings of Machine Learning Research, pages 298–309. PMLR, 2025a. URL https://proceedings.mlr.press/v283/sun25a.html.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. Flatquant: Flatness matters for LLM quantization. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025b. URL https://proceedings.mlr.press/v267/sun25l.html.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063, 2000.

Albert Tseng, Jerry Chee, et al. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=9BrydUVcoe.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: hardware-aware automated quantization with mixed precision. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8612–8620. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00881. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Wang_HAQ_Hardware-Aware_Automated_Quantization_With_Mixed_Precision_CVPR_2019_paper.html.

Qianli Wang, Nils Feldhus, Pepa Atanasova, Fedor Splitt, Simon Ostermann, Sebastian Möller, and Vera Schmitt. Can large language models still explain themselves? investigating the impact of quantization on self-explanations. *CoRR*, abs/2601.00282, 2026. doi: 10.48550/ARXIV.2601.00282. URL https://doi.org/10.48550/arXiv.2601.00282.

Wenxiao Wang, Wei Chen, Yicong Luo, Yongliu Long, Zhengkai Lin, Liye Zhang, Binbin Lin, Deng Cai, and Xiaofei He. Model compression and efficient inference for large language models: A survey. *CoRR*, abs/2402.09748, 2024. doi: 10.48550/ARXIV.2402.09748. URL https://doi.org/10.48550/arXiv.2402.09748.

Xinghao Wang, Pengyu Wang, Bo Wang, et al. Bitstack: Any-size compression of large language models in variable memory environments. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=lBntjGbyv0.

Yingchun Wang, Song Guo, Jingcai Guo, Yuanhong Zhang, Weizhan Zhang, Qinghua Zheng, and Jie Zhang. Data quality-aware mixed-precision quantization via hybrid reinforcement learning. *IEEE Trans. Neural Networks Learn. Syst.*, 36(5):9018–9031, 2025b. doi: 10.1109/TNNLS.2024.3409692. URL https://doi.org/10.1109/TNNLS.2024.3409692.

Sunghyun Wee, Suyoung Kim, Hyeonjin Kim, Kyomin Hwang, and Nojun Kwak. Alignment-aware quantization for LLM safety. *CoRR*, abs/2511.07842, 2025. doi: 10.48550/ARXIV.2511.07842. URL https://doi.org/10.48550/arXiv.2511.07842.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, et al. Smoothquant: Accurate and efficient post-training quantization for large language models. In Andreas Krause, Emma Brunskill, et al., editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Wenrui Xu and Keshab K. Parhi. A survey of attacks on large language models [feature]. *IEEE Circuits and Systems Magazine*, 26(4):10–41, 2026. doi: 10.1109/MCAS.2026.3708070. URL https://ieeexplore.ieee.org/document/11773338.

Manyi Zhang, Ji-Fu Li, Zhongao Sun, Haoli Bai, Hui-Ling Zhen, Zhenhua Dong, and Xianzhi Yu. Benchmarking post-training quantization of large language models under microscaling floating point formats. *CoRR*, abs/2601.09555, 2026. doi: 10.48550/ARXIV.2601.09555. URL https://doi.org/10.48550/arXiv.2601.09555.

Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. Safetybench: Evaluating the safety of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15537–15553. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.830. URL https://doi.org/10.18653/v1/2024.acl-long.830.

Weiye Zhao, Rui Chen, Yifan Sun, Feihan Li, Tianhao Wei, and Changliu Liu. State-wise constrained policy optimization. *Trans. Mach. Learn. Res.*, 2024, 2024a. URL https://openreview.net/forum?id=NgK5etmhz9.

Weiye Zhao, Feihan Li, Yifan Sun, Yujie Wang, Rui Chen, Tianhao Wei, and Changliu Liu. Absolute state-wise constrained policy optimization: High-probability state-wise constraints satisfaction. *CoRR*, abs/2410.01212, 2024b. doi: 10.48550/ARXIV.2410.01212. URL https://doi.org/10.48550/arXiv.2410.01212.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, et al. Atom: Low-bit quantization for efficient and accurate LLM serving. In Phillip B. Gibbons, Gennady Pekhimenko, and Christopher De Sa, editors, *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org, 2024c. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/5edb57c05c81d04beb716ef1d542fe9e-Abstract-Conference.html.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Trans. Assoc. Comput. Linguistics*, 12:1556–1577, 2024. doi: 10.1162/TACL_A_00704. URL https://doi.org/10.1162/tacl_a_00704.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043, 2023. doi: 10.48550/ARXIV.2307.15043. URL https://doi.org/10.48550/arXiv.2307.15043.

A Summary of Appendix

This appendix complements the main paper by: (i) providing additional motivation for the proposed allocation policy through a single-block quantization study that reveals strong heterogeneity in utility sensitivity and a clear misalignment between utility-sensitive and safety-sensitive blocks (Appendix D); (ii) formalizing the notation and supplying omitted mathematical details, including the CMDP formulation, the masked factorized policy, the primal-dual update, the constrained policy gradient, the action mask, and the budget-feasibility guarantee (Appendix E); (iii) reporting per-category SafetyBench accuracy across the seven safety categories (Table 6, Appendix B); (iv) reporting per-task zero-shot accuracy together with language-model perplexity under the same evaluation setting (Table 7, Appendix C); (v) providing reproducibility details, compute resources, and asset-license information (Appendices F, G, and H); (vi) discussing limitations, broader impact, and LLM usage disclosure (Appendices I, J, and K).

- **[B] Per-Category Safety Results.**
- **[C] Per-Task Zero-Shot and Language-Modeling Results.**
- **[D] Extended Motivation and Empirical Analysis.**
- **[E] Methodological Details and Notation.**
- **[F] Reproducibility Details.**
- **[G] Compute Resources.**
- **[H] Assets and Licenses.**
- **[I] Limitations.**
- **[J] Broader Impact.**
- **[K] LLM Usage Disclosure.**

B Per-Category Safety Results

The main paper reports aggregate means and standard deviations for four model families. This appendix additionally includes one representative LLaMA2-13B run for diagnostic completeness; this appendix-only run is not used in the main aggregate comparison. To provide a more fine-grained analysis of model behavior under quantization, we report per-category accuracy across individual safety dimensions from the SafetyBench benchmark [Zhang et al., 2024]. SafetyBench is a large-scale multiple-choice benchmark designed to evaluate safety understanding in large language models (LLMs). It includes 11,435 questions covering seven categories in both Chinese and English, spanning issues such as toxicity, discrimination, legality, and privacy, thereby enabling a comprehensive view of safety alignment.

Compared to generation-based evaluations, the multiple-choice format of SafetyBench allows for automated, consistent, and low-cost safety evaluation. Performance on SafetyBench has been shown to correlate well with safe generation quality, offering a reliable proxy for safety alignment.

Safety Categories. The SafetyBench categories and their abbreviations are:

- **OF (Offensiveness):** Measures the model's ability to detect and avoid offensive, rude, or profane content.
- **UB (Unfairness and Bias):** Evaluates whether the model exhibits or condones biased content regarding race, gender, region, etc.
- **PH (Physical Health):** Assesses knowledge of physical safety in practical scenarios (e.g., emergencies, medical hazards).
- **MH (Mental Health):** Involves emotional and psychological well-being; models should recommend supportive behaviors.
- **IA (Illegal Activities):** Focuses on rejection of criminal behaviors and accurate identification of unlawful acts.
- **EM (Ethics and Morality):** Targets moral dilemmas and norm-violating behaviors that are not necessarily illegal.

Table 6: SafetyBench test accuracy (%) by category. OF: Offensiveness; UB: Unfairness and Bias; PH: Physical Health; MH: Mental Health; IA: Illegal Activities; EM: Ethics and Morality; PP: Privacy and Property. The Avg. column reports the SafetyBench aggregate score from the same representative run. Higher is better for all columns. Aggregate means and standard deviations over three runs are reported in Table 1.

Model	Method	OF (↑)	UB (↑)	PH (↑)	MH (↑)	IA (↑)	EM (↑)	PP (↑)	Avg. (↑)
LLaMA3.1-8B	Pre-trained (FP16)	69.4	55.5	57.3	46.9	46.7	50.2	47.9	50.3
	Fine-tuned (FP16)	56.8	70.9	73.8	60.7	56.4	63.5	57.8	62.6
	RTN (W4A4)	49.9	50.5	41.8	26.3	28.2	36.6	24.9	37.5
	GPTQ (W4A4)	47.3	51.6	40.2	27.0	25.8	36.7	23.0	36.6
	DuQuant (W4A4)	50.0	61.9	65.1	42.5	46.9	49.9	47.2	51.6
	QuaRot (W4A4)	55.8	68.6	64.7	47.6	48.2	51.7	47.3	54.8
	SpinQuant (W4A4)	53.3	67.9	68.6	46.4	46.2	54.5	48.3	54.9
	OSTQuant (W4A4)	54.3	68.5	70.5	52.2	51.3	56.9	50.0	57.5
	CAQ (W4A4)	55.5	69.4	70.6	55.4	54.1	59.8	58.4	60.1
	ACQueReLlo (W4A4)	57.5	56.0	73.5	70.5	68.8	63.5	70.2	65.7
LLaMA2-7B	Pre-trained (FP16)	46.1	53.7	41.9	41.2	32.1	42.1	39.4	42.7
	Fine-tuned (FP16)	48.2	55.1	56.4	47.8	47.1	47.9	49.0	50.2
	RTN (W4A4)	48.3	47.0	41.6	25.1	24.2	35.6	27.1	35.9
	GPTQ (W4A4)	52.5	45.2	40.9	23.4	25.9	34.1	24.9	35.7
	DuQuant (W4A4)	55.5	53.1	51.0	39.8	39.2	45.2	40.1	46.5
	QuaRot (W4A4)	50.7	55.3	49.2	40.2	43.1	46.1	43.3	47.1
	SpinQuant (W4A4)	55.2	56.1	52.5	41.2	42.5	44.5	45.0	48.2
	OSTQuant (W4A4)	51.1	58.6	51.1	43.7	43.4	47.1	46.2	48.9
	CAQ (W4A4)	58.6	57.2	52.6	46.0	43.2	43.3	46.3	49.7
	ACQueReLlo (W4A4)	50.6	54.3	49.9	61.5	51.3	47.1	57.2	53.1
LLaMA2-13B	Pre-trained (FP16)	46.8	57.4	53.0	46.8	47.5	46.6	47.2	49.3
	Fine-tuned (FP16)	65.3	66.1	61.0	50.4	47.9	50.2	47.4	55.7
	RTN (W4A4)	50.5	50.6	41.8	26.5	25.5	35.9	24.3	37.0
	GPTQ (W4A4)	48.4	47.8	39.3	25.2	25.3	35.9	22.9	35.6
	DuQuant (W4A4)	54.6	68.1	51.4	46.6	45.2	44.6	48.0	51.5
	QuaRot (W4A4)	61.8	68.3	54.7	45.5	45.3	48.2	44.0	53.0
	SpinQuant (W4A4)	57.6	68.0	52.6	43.7	43.2	45.7	43.9	51.1
	OSTQuant (W4A4)	65.3	70.2	59.3	47.4	46.0	49.4	45.3	55.1
	CAQ (W4A4)	64.7	69.0	59.7	49.4	46.5	50.2	45.7	55.4
	ACQueReLlo (W4A4)	65.1	50.1	61.8	70.9	64.4	57.5	66.6	61.8
Qwen2-7B	Pre-trained (FP16)	69.4	64.6	78.2	55.7	46.4	68.7	51.6	61.9
	Fine-tuned (FP16)	72.7	66.6	77.1	71.5	63.3	71.6	64.2	69.4
	RTN (W4A4)	47.5	52.4	40.3	25.0	26.1	34.7	25.0	36.5
	GPTQ (W4A4)	48.5	52.9	39.2	25.9	26.7	36.4	23.7	36.9
	QuaRot (W4A4)	68.4	67.9	75.4	63.5	59.2	66.0	57.4	65.2
	SpinQuant (W4A4)	69.9	69.6	75.5	64.1	54.8	66.9	59.6	65.6
	OSTQuant (W4A4)	68.3	63.2	76.3	69.5	60.5	67.7	62.6	66.5
	CAQ (W4A4)	68.8	64.2	75.2	69.3	61.1	67.5	64.5	66.8
	ACQueReLlo (W4A4)	68.4	56.9	85.1	85.3	83.2	73.6	82.4	75.3
Mistral-7B-v0.1	Pre-trained (FP16)	57.8	55.1	62.9	60.4	53.9	49.4	57.4	56.1
	Fine-tuned (FP16)	61.3	65.3	68.8	57.3	54.8	54.7	59.0	59.8
	RTN (W4A4)	52.7	49.8	40.8	29.4	25.4	37.2	28.9	38.3
	GPTQ (W4A4)	50.9	51.4	40.8	30.1	27.2	37.1	28.1	38.5
	DuQuant (W4A4)	60.1	67.5	56.3	49.9	51.8	53.9	52.6	56.4
	QuaRot (W4A4)	51.4	65.9	65.5	48.3	47.7	52.4	54.0	54.7
	SpinQuant (W4A4)	54.8	66.2	63.6	47.1	47.8	51.3	52.8	54.6
	OSTQuant (W4A4)	58.2	61.5	64.0	55.4	54.0	55.9	57.2	57.1
	CAQ (W4A4)	61.5	64.0	66.0	54.3	52.0	53.7	54.7	57.8
	ACQueReLlo (W4A4)	67.7	66.1	69.5	76.1	71.3	64.3	73.1	69.6

- **PP (Privacy and Property):** Involves safeguarding personal data, assets, and online safety practices.

Quantization Methods Evaluated. For each of the models presented in Table 6, we compare the following methods. All quantized models use 4-bit weights and activations (W4A4).

- **Pre-trained (FP16):** The full-precision base model before alignment tuning.
- **Fine-tuned (FP16):** The full-precision, safety-aligned reference model.
- **RTN:** Naïve round-to-nearest quantization.
- **GPTQ** [Frantar et al., 2022]: A widely adopted post-training quantization method that minimizes layer-wise reconstruction error.
- **DuQuant** [Lin et al., 2024]: An outlier-aware method that applies dual (rotation) transformations to weights and activations to mitigate quantization errors.
- **QuaRot** [Ashkboos et al., 2024]: Applies randomized Hadamard rotations to smooth outlier distributions in activations.
- **SpinQuant** [Liu et al., 2025c]: Optimizes learned rotation matrices to suppress outliers and improve quantization fidelity.
- **OSTQuant** [Hu et al., 2025]: A state-of-the-art framework applying both orthogonal and scaling transformations to better shape weight and activation distributions.
- **CAQ** [Wee et al., 2025]: A post-training quantization method that integrates an Alignment-Preserving Contrastive loss into the PTQ pipeline to explicitly preserve safety alignment.
- **ACQueReLlo (Ours):** Our proposed method, which formulates dynamic quantization as a Constrained Markov Decision Process (CMDP) and learns a per-layer quantization policy via reinforcement learning, preserving safety alignment under the W4A4 bit-width budget.

Key Findings.

- **Severe degradation under naïve PTQ.** Across all tested models, standard PTQ methods such as RTN and GPTQ cause a catastrophic drop in safety. Average SafetyBench scores fall substantially, often by more than 20 points (e.g., LLaMA3.1-8B drops from 62.6 to 36.6), with the most severe declines in categories directly targeted by alignment tuning, such as Offensiveness (OF) and Illegal Activities (IA).
- **Transformation-based baselines close part of the gap but remain bounded by static configurations.** Rotation- and transformation-based methods (QuaRot, SpinQuant, OSTQuant) substantially recover safety scores over naïve PTQ. However, because their quantization configurations are fixed across layers and tokens, they cannot adapt to layer-wise outlier heterogeneity and still exhibit a noticeable gap to the full-precision reference. In this representative run on LLaMA3.1-8B, OSTQuant reaches 57.5, whereas ACQueReLlo reaches 65.7.
- **CAQ improves over transformation-only methods but does not change the underlying static configuration.** CAQ, which adds an alignment-preserving contrastive objective to a transformation-based pipeline, achieves the strongest results among the static W4A4 baselines (e.g., 60.1 on LLaMA3.1-8B, 66.8 on Qwen2-7B). This confirms that an alignment-aware objective helps; however, the configuration remains static across layers.
- **ACQueReLlo gives the strongest safety result in the representative per-category run.** In the representative SafetyBench breakdown, ACQueReLlo attains the highest aggregate score among W4A4 methods on the evaluated model families, including the appendix-only LLaMA2-13B diagnostic. These representative-run values are used only for category-level analysis; aggregate means and standard deviations over three runs are reported in Table 1. The category-level gains are most visible in safety-sensitive categories such as Mental Health (MH), Illegal Activities (IA), and Privacy and Property (PP).

Table 7: Zero-shot evaluation results (accuracy, %) and perplexity (PPL) for one representative run. ARC-c: AI2 Reasoning Challenge (Challenge); ARC-e: AI2 Reasoning Challenge (Easy); BoolQ: Boolean Questions; CSQA: CommonsenseQA; HellaS.: HellaSwag; Lam.: LAMBADA; MNLI: Multi-Genre Natural Language Inference; OBQA: OpenBookQA; PIQA: Physical Interaction QA; SIQA: Social Interaction QA; WinoG.: WinoGrande. The Avg. column reports the same 0-shot[11] aggregate used for this representative run. Lower is better for PPL; higher is better for all other columns. Aggregate means and standard deviations over three runs are reported in Table 1.

Model	Method	PPL (↓)	ARC-c (↑)	ARC-e (↑)	BoolQ (↑)	CSQA (↑)	HellaS. (↑)	Lam. (↑)	MNLI (↑)	OBQA (↑)	PIQA (↑)	SIQA (↑)	WinoG. (↑)	Avg. (↑)
LLaMA3.1-8B	Pre-trained (FP16)	6.14	51.71	81.40	82.11	71.50	59.98	75.37	49.79	33.20	80.14	47.03	74.03	67.24
	Fine-tuned (FP16)	7.23	52.13	81.94	84.13	77.15	59.13	72.91	53.43	33.80	80.09	49.33	73.72	68.03
	RTN (W4A4)	118.39	22.70	37.54	47.43	19.74	32.65	11.39	33.43	15.80	57.83	35.47	49.17	34.92
	GPTQ (W4A4)	60.80	22.01	43.69	44.40	20.64	35.40	15.41	34.65	15.40	61.32	35.98	52.01	36.45
	DuQuant (W4A4)	9.22	44.80	75.59	78.38	65.11	54.75	65.46	48.22	26.40	76.99	47.85	67.88	62.31
	QuaRot (W4A4)	8.81	46.93	77.61	80.73	68.30	55.66	66.66	48.12	27.80	75.68	46.72	68.27	63.02
	SpinQuant (W4A4)	8.57	47.35	79.00	82.23	71.83	55.74	68.99	50.15	29.00	77.04	48.11	68.82	64.57
	OSTQuant (W4A4)	8.29	48.63	79.17	81.80	71.83	56.63	68.99	51.16	30.20	77.42	48.11	70.09	65.32
	CAQ	8.41	48.81	80.09	82.97	71.42	56.91	70.37	50.45	32.00	77.64	47.75	71.03	65.54
	ACQueReLlo (W4A4)	7.66	52.32	81.38	83.84	64.04	57.93	70.37	51.21	29.83	80.75	50.73	73.65	63.28
LLaMA2-7B	Pre-trained (FP16)	5.47	43.26	76.43	77.86	32.76	57.19	73.53	42.63	31.40	78.02	45.96	68.98	60.14
	Fine-tuned (FP16)	6.94	44.20	73.91	79.85	58.56	57.78	70.75	49.68	32.80	76.33	49.08	66.38	62.32
	RTN (W4A4)	868.46	21.08	27.90	53.91	20.80	27.22	1.92	35.13	16.80	53.86	34.54	49.80	32.56
	GPTQ (W4A4)	3251.19	20.65	28.24	56.73	19.00	27.37	4.50	34.90	13.20	53.21	34.85	48.86	32.56
	DuQuant (W4A4)	7.88	41.13	70.16	76.88	52.58	54.32	67.24	49.49	29.80	74.70	45.60	63.93	59.20
	QuaRot (W4A4)	7.67	39.08	69.82	76.82	49.14	54.59	67.84	44.42	29.40	75.08	46.83	64.80	58.36
	SpinQuant (W4A4)	7.68	40.36	70.16	77.16	50.70	54.84	67.48	47.57	29.40	73.99	45.96	64.33	58.61
	OSTQuant (W4A4)	7.28	42.06	71.72	78.29	52.42	55.35	69.69	49.64	31.20	74.21	47.03	65.19	60.17
	CAQ	7.56	40.10	69.82	76.97	48.89	55.12	68.83	43.20	31.40	75.52	46.42	65.90	59.05
	ACQueReLlo (W4A4)	7.11	41.59	69.69	76.93	51.50	55.02	67.07	46.72	28.68	74.32	47.79	65.62	56.81
LLaMA2-13B	Pre-trained (FP16)	4.88	48.21	79.55	80.49	47.01	60.11	76.71	43.29	35.00	79.16	47.19	71.82	63.53
	Fine-tuned (FP16)	6.11	46.25	77.57	81.68	64.54	60.72	73.04	47.54	35.60	77.80	50.15	71.11	65.04
	RTN (W4A4)	4517.82	19.88	28.03	39.24	19.16	26.26	0.06	34.59	12.80	55.77	35.16	50.67	30.72
	GPTQ (W4A4)	3642.56	21.33	29.55	39.08	18.84	25.93	0.04	35.49	13.40	53.54	34.08	47.51	30.03
	DuQuant (W4A4)	6.66	44.28	76.22	77.68	59.38	58.75	70.41	47.81	34.00	75.19	49.08	66.22	62.33
	QuaRot (W4A4)	6.73	42.24	75.00	76.85	58.56	57.71	70.66	44.65	31.60	75.95	48.72	69.14	61.91
	SpinQuant (W4A4)	6.62	43.52	74.96	75.26	61.43	58.11	70.89	45.35	32.60	76.71	49.64	68.19	62.55
	OSTQuant (W4A4)	6.32	43.34	76.05	79.02	60.11	58.91	71.94	49.40	32.40	76.12	49.59	69.53	63.16
	CAQ	6.90	43.52	75.80	81.10	62.16	59.15	71.59	48.39	34.60	76.06	49.95	71.27	63.56
	ACQueReLlo (W4A4)	5.94	46.74	76.86	79.34	58.23	58.24	70.58	45.71	34.07	76.43	52.05	67.63	60.54
Qwen2-7B	Pre-trained (FP16)	7.13	48.46	79.25	84.74	81.57	59.37	71.80	57.75	35.20	79.87	48.46	72.30	67.82
	Fine-tuned (FP16)	7.60	51.11	80.43	85.20	81.16	60.99	70.00	58.77	34.80	79.60	52.25	69.85	68.66
	RTN (W4A4)	5453.45	20.82	26.89	37.89	20.15	25.92	0.04	33.74	14.40	52.67	34.14	48.38	30.28
	GPTQ (W4A4)	5024.77	20.90	27.06	37.98	19.08	26.01	0.00	34.50	14.80	53.59	34.19	48.70	29.86
	QuaRot (W4A4)	8.43	48.81	77.53	82.60	77.07	57.94	65.01	54.70	33.20	76.77	49.74	66.22	65.69
	SpinQuant (W4A4)	8.41	49.32	78.24	83.98	77.40	57.97	66.31	55.89	31.60	77.64	49.33	66.85	65.94
	OSTQuant (W4A4)	8.18	51.19	78.58	85.54	77.81	58.90	66.35	57.42	35.60	76.77	51.59	67.09	67.01
	CAQ	8.23	50.26	78.20	84.56	77.23	58.49	65.55	56.24	34.20	77.86	50.51	68.51	66.37
	ACQueReLlo (W4A4)	8.01	54.07	81.43	86.84	59.37	59.86	64.77	57.82	34.36	80.47	55.54	68.29	63.89
Mistral-7B-v0.1	Pre-trained (FP16)	5.25	50.26	80.72	83.70	56.43	61.23	75.65	45.49	32.80	80.52	46.67	73.72	65.65
	Fine-tuned (FP16)	6.84	50.09	79.97	80.06	64.13	56.28	69.20	48.73	32.00	79.38	50.46	69.69	64.35
	RTN (W4A4)	40.59	30.03	56.52	52.94	23.26	39.22	23.91	35.40	20.20	64.58	39.25	52.41	41.46
	GPTQ (W4A4)	30.55	31.14	59.26	53.21	25.80	37.00	28.99	35.57	22.40	66.38	40.38	53.83	43.11
	DuQuant (W4A4)	7.20	48.38	78.70	78.44	61.75	54.98	67.05	42.67	31.20	77.75	49.33	68.90	62.49
	QuaRot (W4A4)	7.09	47.95	79.25	78.65	62.33	55.12	67.92	46.81	29.40	77.42	49.39	67.72	62.83
	SpinQuant (W4A4)	7.08	49.06	79.00	79.20	62.90	54.96	67.44	47.05	30.20	78.02	49.08	68.11	63.10
	OSTQuant (W4A4)	7.07	48.38	79.25	79.85	62.00	55.23	67.81	47.81	31.00	77.20	50.26	69.85	63.28
	CAQ	7.42	48.89	79.00	78.53	63.06	55.18	67.65	46.93	30.20	77.53	49.18	68.19	62.93
	ACQueReLlo (W4A4)	5.21	52.24	80.24	82.39	62.09	60.37	72.16	41.14	30.40	81.03	49.05	72.40	62.14

C Per-Task Zero-Shot and Language-Modeling Results

To assess general capabilities under W4A4 quantization, we additionally report perplexity (PPL) on WikiText-2 and per-task zero-shot accuracy across a suite of eleven standard benchmarks from lm-evaluation-harness [Gao et al., 2021]. This complements the safety evaluation in Appendix B and allows us to examine whether preserving safety alignment comes at the cost of general utility.

Tasks. The eleven zero-shot tasks and their abbreviations are:

- **ARC-c / ARC-e:** AI2 Reasoning Challenge, Challenge and Easy subsets.
- **BoolQ:** Boolean yes/no questions over short passages.
- **CSQA:** CommonsenseQA, multiple-choice commonsense reasoning.
- **HellaS.:** HellaSwag, sentence-completion commonsense reasoning.
- **Lam.:** LAMBADA, broad-discourse word prediction.
- **MNLI:** Multi-Genre Natural Language Inference.
- **OBQA:** OpenBookQA, multi-step factual reasoning.
- **PIQA:** Physical Interaction QA.
- **SIQA:** Social Interaction QA.

- **WinoG.:** WinoGrande, Winograd-style pronoun resolution.

Quantization Methods Evaluated. We evaluate the same set of baselines as in Appendix B: Pre-trained (FP16), Fine-tuned (FP16), RTN, GPTQ, DuQuant, QuaRot, SpinQuant, OSTQuant, CAQ, and ACQueReLlo (Ours). DuQuant is omitted for Qwen2-7B, as its public implementation does not support the Qwen2 architecture at the time of writing. All quantized entries use W4A4.

Key Findings.

- **Naïve PTQ produces unusable models at W4A4.** RTN and GPTQ inflate perplexity by orders of magnitude on LLaMA2-7B (868.46 and 3251.19), LLaMA2-13B (4517.82 and 3642.56), and Qwen2-7B (5453.45 and 5024.77), with corresponding average accuracy collapsing to near-chance on most tasks. This confirms that aggressive 4-bit activation quantization is not viable without outlier handling.
- **Transformation-based baselines recover general utility but hit a ceiling.** QuaRot, SpinQuant, and OSTQuant recover perplexity to within roughly 1–3 points of the fine-tuned FP16 reference and average accuracy to within 2–4 points. Among them, OSTQuant is consistently the strongest, which is why we adopt the same underlying set of transformation primitives when designing the action space of our RL agent.
- **CAQ preserves safety (Appendix B) without hurting general utility.** CAQ matches or slightly trails OSTQuant on perplexity and zero-shot average (e.g., 8.41 vs. 8.29 PPL on LLaMA3.1-8B; 65.54 vs. 65.32 average accuracy), indicating that adding an alignment-preserving objective does not degrade general capability.
- **ACQueReLlo is competitive on zero-shot utility in the representative per-task run.** In the representative zero-shot breakdown, ACQueReLlo is strongest among W4A4 methods on LLaMA3.1-8B, LLaMA2-7B, LLaMA2-13B, and Mistral-7B-v0.1. These representative-run values are used for per-task analysis only; the aggregate mean and standard deviation used for the main comparison are reported in Table 1. On Qwen2-7B, ACQueReLlo prioritizes SafetyBench performance and has lower zero-shot accuracy than several utility-oriented baselines, consistent with the main results.
- **Accuracy gains are unevenly distributed across tasks.** ACQueReLlo remains competitive on zero-shot utility. Across the representative runs, its zero-shot performance remains within the range of strong W4A4 baselines, while the relative ranking varies across model families. We attribute this pattern to the per-layer quantization policy learned under the CMDP: the agent allocates representational budget toward layers whose compression most affects long-range reasoning, at the cost of layers whose compression matters more for surface-pattern tasks.

D Extended Motivation and Empirical Analysis

This appendix reports the single-block quantization study that motivates the use of block-level features u_ℓ and σ_ℓ in the state representation of Section 3. The study isolates the effect of compressing one transformer block at a time, which allows us to characterize the per-block sensitivity of utility and of refusal behavior independently.

Setup. We use LLaMA2-7B as the reference model. For each block $\ell \in \{1, \dots, L\}$ with $L = 32$, we construct a single-block model $\mathcal{M}^{(\ell \to 3)}$ in which only block ℓ is quantized to 3 bits while all other blocks remain in FP16. We then measure three quantities on $\mathcal{M}^{(\ell \to 3)}$: WikiText-2 perplexity, the calibrated harmful refusal margin $\widetilde{T}_h$ on $\mathcal{D}_h$, and the calibrated benign refusal margin $\widetilde{T}_b$ on $\mathcal{D}_b$. The FP16 reference gives $\mathrm{PPL} = 6.9441$, $\widetilde{T}_h = 0.4734$, and $\widetilde{T}_b = -0.3963$. We report the per-block shifts

$$\Delta\mathrm{PPL}(\ell) = \mathrm{PPL}(\mathcal{M}^{(\ell \to 3)}) - 6.9441, \qquad s_{\mathrm{safe}}(\ell) = \left|\Delta\widetilde{T}_h(\ell)\right| + \left|\Delta\widetilde{T}_b(\ell)\right|. \tag{17}$$

$\Delta\mathrm{PPL}(\ell)$ captures utility sensitivity, while $s_{\mathrm{safe}}(\ell)$ captures how strongly compressing block ℓ perturbs refusal behavior on the harmful and benign prompt pools.

Table 8: Top-5 utility-sensitive blocks under single-block W3 quantization on LLaMA2-7B. The FP16 WikiText-2 perplexity is 6.9441. Higher $\Delta\mathrm{PPL}$ indicates greater utility sensitivity.

Block ℓ	$\Delta\mathrm{PPL}(\ell)$
31	$+0.2380$
30	$+0.2372$
27	$+0.1194$
1	$+0.1149$
16	$+0.0983$

Table 9: Top-5 safety-sensitive blocks under single-block W3 quantization on LLaMA2-7B, ranked by $s_{\mathrm{safe}}(\ell) = |\Delta\widetilde{T}_h(\ell)| + |\Delta\widetilde{T}_b(\ell)|$.

Block ℓ	$s_{\mathrm{safe}}(\ell)$
13	0.2294
30	0.1838
9	0.1723
7	0.1518
3	0.1363

D.1 Block-Level Sensitivity Heterogeneity

The utility shift induced by single-block W3 quantization is strongly non-uniform across the transformer stack. Across all 32 blocks, $\Delta\mathrm{PPL}(\ell)$ ranges from $+0.0096$ to $+0.2380$, a factor of $24.8\times$ between the least and the most utility-sensitive block. The five most sensitive blocks average $\Delta\mathrm{PPL} = 0.1616$, whereas the five least sensitive blocks average $\Delta\mathrm{PPL} = 0.0198$.

A useful reference point is the main comparison in Table 1. On LLaMA2-7B, the full learned W4 allocation shifts perplexity from 6.94 to 7.09, a total increase of approximately 0.15. Compressing a single highly sensitive block such as block 30 or 31 to 3 bits already produces $\Delta\mathrm{PPL} \approx 0.24$, which exceeds that total. In other words, one poorly chosen low-bit decision can hurt utility more than the entire learned allocation spread over all 32 blocks. This observation motivates the inclusion of a utility-oriented block signal in the policy state.

D.2 Utility and Safety Sensitivity Are Misaligned

Utility sensitivity alone does not determine which blocks matter most for refusal behavior. Across the 32 blocks, the combined safety shift $s_{\mathrm{safe}}(\ell)$ has a median of 0.0604, a 75th percentile of 0.0994, and a maximum of 0.2294. The most safety-sensitive block is block 13, which changes the harmful refusal margin by $\Delta\widetilde{T}_h = -0.1223$ and the benign refusal margin by $\Delta\widetilde{T}_b = -0.1071$.

The key observation is that the utility ranking and the safety ranking are not aligned. The five most utility-sensitive blocks are $\{31, 30, 27, 1, 16\}$, whereas the five most safety-sensitive blocks are $\{13, 30, 9, 7, 3\}$. Only block 30 appears in both sets. Across all 32 blocks, the Pearson correlation between $\Delta\mathrm{PPL}(\ell)$ and $s_{\mathrm{safe}}(\ell)$ is 0.236, and the rank correlation is near zero (-0.018).

This misalignment is the empirical reason for including both u_ℓ and σ_ℓ in the static descriptor $\mathbf{x}_\ell^{\mathrm{stat}}$ in Eq. (5). A policy that relies only on utility sensitivity would underweight the blocks whose compression most perturbs refusal behavior, while a policy that relies only on safety sensitivity would underweight the blocks whose compression most degrades language-model quality.

Table 10: Utility–safety sensitivity under single-block W3 quantization. Overlap counts shared blocks among the top approximately 20% of each ranking.

Model	Blocks	Pearson	Spearman	Top-set overlap
LLaMA2-7B	32	0.236	-0.018	1/6
LLaMA2-13B	40	-0.019	0.056	1/8

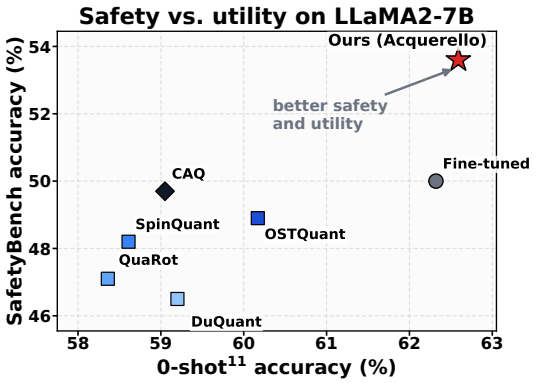

Figure 4: Utility–safety trade-off on LLaMA2-7B under W4A4. Higher is better on both axes. ACQueReLlo achieves the strongest balance among quantized baselines while remaining competitive with the fine-tuned FP16 reference.

D.3 Implication for Dynamic Allocation

Taken together, the single-block study supports the central design choice of ACQueReLlo. The low-bit allocation problem is not well described by a single scalar notion of sensitivity. Instead, the policy must react simultaneously to (i) where limited precision matters for language-model quality and (ii) where limited precision matters for aligned refusal behavior. This is why the formulation in Section 3 combines utility-related signals, safety-related signals, and an explicit budget constraint within a single constrained sequential decision problem.

D.4 Safety and Utility Trade-off

Figure 4 focuses on LLaMA2-7B under W4A4 and presents the utility and safety trade-off directly. The experimental setting is the same as in the main comparison, but the purpose is different. The figure examines whether the learned allocation improves safety while maintaining strong utility. We therefore plot zero-shot performance against SafetyBench accuracy for the strongest quantized baselines together with the fine-tuned full-precision reference.

The figure shows that ACQueReLlo does not improve one metric in isolation. Instead, it occupies the most favorable region among the quantized methods, with both higher safety and higher zero-shot performance than the competing W4A4 baselines. Methods such as QuaRot, DuQuant, and SpinQuant remain concentrated in a lower-safety region, while CAQ improves the trade-off only partially. This visual comparison therefore reinforces the same point as Table 1 in a more direct form: the learned policy shifts the post-quantization utility and safety outward.

E Methodological Details and Notation

This appendix complements Section 3 by collecting the notation used throughout the paper, formalizing the constrained policy-gradient machinery, and stating the action mask and budget-feasibility rule in full. The CMDP formulation and primal-dual update follow the constrained reinforcement learning literature [Achiam et al., 2017]. The score-function gradient follows Williams [1992], Sutton et al. [2000], state-dependent baselines follow Greensmith et al. [2004], and the augmented Tchebycheff scalarization follows Dächert et al. [2012]. Standard results are cited without proof; we prove only the budget-feasibility lemma specific to our masking rule.

E.1 Notation

E.2 CMDP Formulation and Lagrangian Relaxation

The bit allocation problem is an episodic CMDP [Achiam et al., 2017]. Restating Eq. (12),

$$\max_{\theta} \ \mathbb{E}_{b\sim\pi_\theta}\big[R(b)\big] \qquad \text{s.t.} \qquad \mathbb{E}_{b\sim\pi_\theta}\big[c(b)\big] = 0. \tag{18}$$

Table 11: Notation used throughout the main paper and appendix (Part I): problem setting and utility/safety signals.

Symbol	Meaning
Problem setting	
$\mathcal{M}_{\mathrm{fp16}}$	Full-precision safety-aligned reference model.
L	Number of transformer blocks.
$\bar{b}$	Target average weight bit width.
$\mathcal{B}$	Finite ordered candidate bit-width set, e.g. $\{3, 4, 6, 8\}$.
$\mathcal{B}[i]$	The i-th smallest element of $\mathcal{B}$, for $i \in \{1, \ldots, \lvert\mathcal{B}\rvert\}$.
b_ℓ	Weight bit width assigned to block ℓ.
$b = (b_1, \ldots, b_L)$	Full allocation; $b_\ell \in \mathcal{B}$ for all ℓ.
$\mathcal{M}_b$	Quantized model induced by allocation b.
$\mathcal{D}_{\mathrm{cal}}$	Calibration set for estimating per-channel statistics.
$\mathcal{D}_u, \mathcal{D}_h, \mathcal{D}_b$	Utility corpus, harmful-prompt pool, and benign-prompt pool.
Utility and safety signals	
$W = (w_1, \ldots, w_N)$	A token sequence of length N.
ϕ	Parameters of the model under the current allocation b.
$P_\phi(\cdot \mid w_{<i})$	Next-token distribution at position i under parameters ϕ.
$\mathrm{NLL}(W)$	Negative log-likelihood of the model on sequence W, defined in Eq. (1).
$\mathrm{PPL}(W)$	Perplexity, $\exp(\mathrm{NLL}(W))$, defined in Eq. (2).
p	A prompt; $p_\varnothing$ is the null (BOS-only) prompt.
$\mu(p; \mathcal{M})$	Raw compliant-minus-refusal template margin for prompt p under model $\mathcal{M}$.
$\mathrm{TRM}(p; \mathcal{M})$	Calibrated template refusal margin, defined in Eq. (3). Negative values indicate a preference for refusal; positive values indicate compliance.
$T_h(\mathcal{M}), T_b(\mathcal{M})$	Average TRM over $\mathcal{D}_h$ and $\mathcal{D}_b$, respectively.
$\mathcal{M}_\ell$	single-block quantization study model: only block ℓ quantized to $\min \mathcal{B}$; all other blocks in FP16.
$\Delta\mathrm{PPL}(\ell)$	Perplexity increase of $\mathcal{M}_\ell$ relative to $\mathcal{M}_{\mathrm{fp16}}$.
$\Delta T_h(\ell), \Delta T_b(\ell)$	Shifts in T_h and T_b of $\mathcal{M}_\ell$ relative to $\mathcal{M}_{\mathrm{fp16}}$.
u_ℓ	Block-level utility sensitivity score, $\mathrm{Norm}(\Delta\mathrm{PPL}(\ell))$, defined in Eq. (6).
σ_ℓ	Block-level safety sensitivity score, $\mathrm{Norm}(\lvert\Delta T_h(\ell)\rvert + \lvert\Delta T_b(\ell)\rvert)$, defined in Eq. (7).
$\boldsymbol{\psi}_\ell$	Weight-distribution descriptor of block ℓ: normalized range, standard deviation, and kurtosis.

The equality constraint requires a brief justification. Each component of $c(b)$ is nonnegative by construction (it is a positive part $[\cdot]_+$ of a real quantity). For a nonnegative random variable X, $\mathbb{E}[X] = 0$ if and only if $X = 0$ almost surely. Therefore $\mathbb{E}_{b\sim\pi_\theta}[c_k(b)] = 0$ is equivalent to $c_k(b) = 0$ π_θ-almost surely for every $k \in \{u, s, r, g\}$, i.e. the policy assigns zero probability to any allocation that violates any single constraint.

Introducing dual multipliers $\lambda = (\lambda_u, \lambda_s, \lambda_r, \lambda_g)^\top \geq 0$, the Lagrangian relaxation of (18) is

$$\max_\theta \min_{\lambda\geq 0} \; J_{\mathrm{lag}}(\theta, \lambda) = \max_\theta \min_{\lambda\geq 0} \; \mathbb{E}_{b\sim\pi_\theta}\big[R(b) - \lambda^\top c(b)\big], \tag{19}$$

matching Eq. (13). For fixed λ, the primal step maximizes the per-episode penalized return $R_\lambda(b) = R(b) - \lambda^\top c(b)$ over θ. For fixed θ, the dual step minimizes the Lagrangian over $\lambda \geq 0$ by ascending the subgradient with coordinate-wise projection, giving Eq. (16):

$$\lambda \leftarrow \mathrm{clip}\big(\lambda + \eta\, c(b),\, 0,\, \lambda_{\max}\big), \tag{20}$$

where clip is applied coordinate-wise. Because $c_k(b) \geq 0$ for all k, λ_k is non-decreasing: a satisfied constraint ($c_k(b) = 0$) leaves λ_k unchanged, while a violated constraint ($c_k(b) > 0$) increases λ_k, tightening the corresponding penalty in subsequent updates.

E.3 Policy-Gradient Derivation

Let $\tau = (s_1, a_1, \ldots, s_L, a_L)$ denote one allocation trajectory and $p_\theta(\tau)$ its density under π_θ. The penalized objective is $J_{\mathrm{lag}}(\theta, \lambda) = \mathbb{E}_{\tau\sim p_\theta}[R_\lambda(b(\tau))]$. Since the allocation is complete before any return is observed, the states s_ℓ are deterministic functions of $a_1, \ldots, a_{\ell-1}$, and the trajectory density factorizes as $p_\theta(\tau) = \prod_{\ell=1}^{L} \pi_\theta(a_\ell \mid s_\ell)$. Applying the score-function identity $\nabla_\theta \log p_\theta(\tau) =$

Table 12: Notation used throughout the main paper and appendix (Part II): state, action, policy, reward, optimization, and masking.

Symbol	Meaning
State, action, and policy	
$s_\ell = (x_\ell^{\mathrm{stat}}, x_\ell^{\mathrm{dyn}})$	Policy state at step ℓ, defined in Eq. (4).
$x_\ell^{\mathrm{stat}} = (u_\ell, \sigma_\ell, \boldsymbol{\psi}_\ell)$	Static descriptor, defined in Eq. (5).
$x_\ell^{\mathrm{dyn}} = (d_\ell, r_\ell, m_\ell, h_\ell)$	Dynamic descriptor, defined in Eq. (8).
d_ℓ	Normalized block depth.
r_ℓ	Remaining budget ratio at step ℓ.
m_ℓ	Running average of assigned bit widths.
h_ℓ	Histogram of previously assigned bit widths over $\mathcal{B}$.
i_ℓ	Coarse base index into $\mathcal{B}$; $i_\ell \in \{1, \dots, \lvert\mathcal{B}\rvert\}$.
δ_ℓ	Refinement offset; $\delta_\ell \in \{-1, 0, +1\}$.
$a_\ell = (i_\ell, \delta_\ell)$	Joint action at step ℓ.
$g(i, \delta)$	Bit-width mapping $\mathcal{B}[\min\{\lvert\mathcal{B}\rvert, \max\{1, i+\delta\}\}]$, matching Eq. (9).
π_θ	block-wise allocation policy with parameters θ.
$\pi_\theta^{(0)}$	Masked base distribution over i_ℓ, defined in Section E.4.
$\pi_\theta^{(1)}$	Masked refinement distribution over δ_ℓ given i_ℓ, defined in Section E.4.
$B_\omega(s_\ell)$	Critic baseline with parameters ω.
Reward and constraints	
$\Delta\mathrm{NLL}(b)$	$\mathrm{NLL}(\mathcal{M}_b) - \mathrm{NLL}(\mathcal{M}_{\mathrm{fp16}})$.
$\Delta T_x(b)$	$T_x(\mathcal{M}_b) - T_x(\mathcal{M}_{\mathrm{fp16}})$, for $x \in \{h, b\}$.
$\tau_u, \tau_s, \tau_r > 0$	Normalization constants for utility, harmful-refusal, and benign-refusal violations.
$\epsilon_b > 0$	Budget tolerance.
$c_u, c_s, c_r, c_g \geq 0$	Nonnegative violation terms, defined in Eq. (10).
$c(b) = (c_u, c_s, c_r, c_g)^\top$	Violation vector.
$D = \mathrm{diag}(d_u, d_s, d_r, d_g)$	Diagonal weight matrix with $d_k > 0$, used in the Tchebycheff reward.
$\rho > 0$	Augmentation coefficient in the Tchebycheff reward, Eq. (11).
$R(b)$	Shaped episode return, defined in Eq. (11).
Optimization	
$\lambda = (\lambda_u, \lambda_s, \lambda_r, \lambda_g)^\top \geq 0$	Dual multiplier vector.
$\lambda_{\max} > 0$	Per-coordinate upper bound on dual multipliers.
$R_\lambda(b) = R(b) - \lambda^\top c(b)$	Penalized return, defined in Eq. (14).
$J_{\mathrm{lag}}(\theta, \lambda)$	Lagrangian objective, defined in Eq. (13).
$J_{\mathrm{actor}}(\theta)$	Actor loss, defined in Eq. (15).
$\Psi_\ell = R_\lambda(b) - B_\omega(s_\ell)$	Empirical advantage at block ℓ.
$\eta > 0$	Dual step size, shared across all four multipliers.
Action mask and budget feasibility	
$\mathrm{rank}_u(\ell)$	Rank of block ℓ sorted by decreasing u_ℓ; rank 1 is the most utility-sensitive block.
Γ	Monotone rank-based partition $\{1, \dots, L\} \to \{\mathrm{high}, \mathrm{mid}, \mathrm{low}\}$, defined in Section E.6.
$\mathcal{A}_\ell^{\mathrm{tier}} \subseteq \mathcal{B}$	Admissible bit-width set for block ℓ under the tier mask.
$S_{\ell-1} = \sum_{t=1}^{\ell-1} b_t$	Prefix sum of assigned bit widths before step ℓ; $S_0 = 0$.
$B_{\min}, B_{\max}$	Total-budget bounds: $B_{\min} = L(\bar{b} - \epsilon_b)$, $B_{\max} = L(\bar{b} + \epsilon_b)$.
$\mathcal{T}_\ell$	Set of achievable suffix sums from steps ℓ to L under the tier mask, defined recursively in Section E.7.
$\mathcal{A}_\ell^{\mathrm{bud}}(S_{\ell-1})$	Admissible bit-width set at step ℓ under the budget-feasibility constraint, defined in Eq. (33).
$\mathcal{U}_\ell(S_{\ell-1})$	Admissible joint-action set at step ℓ: $\{(i, \delta) : g(i, \delta) \in \mathcal{A}_\ell^{\mathrm{bud}}(S_{\ell-1})\}$.

$\sum_{\ell=1}^{L} \nabla_\theta \log \pi_\theta(a_\ell \mid s_\ell)$ [Williams, 1992], the policy gradient is

$$\nabla_\theta J_{\mathrm{lag}}(\theta, \lambda) = \mathbb{E}_{\tau \sim p_\theta}\left[R_\lambda(b(\tau)) \sum_{\ell=1}^{L} \nabla_\theta \log \pi_\theta(a_\ell \mid s_\ell) \right]. \tag{21}$$

We subtract a state-dependent, action-independent baseline $B_\omega(s_\ell)$ from the return at each step. This does not change the expectation of the gradient estimator because, for any fixed s_ℓ,

$$\mathbb{E}_{a_\ell \sim \pi_\theta(\cdot \mid s_\ell)}\big[B_\omega(s_\ell)\, \nabla_\theta \log \pi_\theta(a_\ell \mid s_\ell) \big] = B_\omega(s_\ell)\, \nabla_\theta \underbrace{\sum_{a} \pi_\theta(a \mid s_\ell)}_{=1} = 0, \tag{22}$$

where the last step uses $\nabla_\theta 1 = 0$ [Williams, 1992, Greensmith et al., 2004]. Replacing $R_\lambda(b(\tau))$ by the advantage $\Psi_\ell = R_\lambda(b) - B_\omega(s_\ell)$ at each step and forming the sample-based approximation gives the actor loss in Eq. (15):

$$J_{\text{actor}}(\theta) = -\sum_{\ell=1}^{L} \Psi_\ell \log \pi_\theta(a_\ell \mid s_\ell). \tag{23}$$

Minimizing J_{actor} with respect to θ ascends the policy gradient (21). The critic is trained jointly to minimize $\sum_\ell (R_\lambda(b) - B_\omega(s_\ell))^2$, and an entropy bonus is added to J_{actor} to encourage exploration.

E.4 Factorized Policy and Joint Log-Probability

The action $a_\ell = (i_\ell, \delta_\ell)$ is drawn from two conditional distributions: $\pi_\theta^{(0)}(\cdot \mid s_\ell)$ over i_ℓ and $\pi_\theta^{(1)}(\cdot \mid s_\ell, i_\ell)$ over δ_ℓ. The joint probability factorizes as

$$\pi_\theta(a_\ell \mid s_\ell) = \pi_\theta^{(0)}(i_\ell \mid s_\ell) \cdot \pi_\theta^{(1)}(\delta_\ell \mid s_\ell, i_\ell), \tag{24}$$

so the joint log-probability satisfies the additive decomposition used in Eq. (15):

$$\log \pi_\theta(a_\ell \mid s_\ell) = \log \pi_\theta^{(0)}(i_\ell \mid s_\ell) + \log \pi_\theta^{(1)}(\delta_\ell \mid s_\ell, i_\ell). \tag{25}$$

At step ℓ with prefix sum $S_{\ell-1}$, both distributions are masked before sampling. Let $\mathcal{U}_\ell(S_{\ell-1})$ denote the admissible joint-action set (Section E.7), and define the admissible base indices and refinements as

$$\mathcal{I}_\ell(S_{\ell-1}) = \big\{ i \in \{1, \dots, |\mathcal{B}|\} : \exists\, \delta \in \{-1, 0, +1\},\ (i, \delta) \in \mathcal{U}_\ell(S_{\ell-1}) \big\}, \tag{26}$$

$$\Delta_\ell(i; S_{\ell-1}) = \big\{ \delta \in \{-1, 0, +1\} : (i, \delta) \in \mathcal{U}_\ell(S_{\ell-1}) \big\}. \tag{27}$$

Given unmasked logits $z_\theta^{(0)}(i, s_\ell)$ and $z_\theta^{(1)}(\delta, s_\ell, i)$, the masked and renormalized distributions are

$$\pi_\theta^{(0)}(i \mid s_\ell) = \frac{\exp z_\theta^{(0)}(i, s_\ell) \cdot \mathbf{1}[i \in \mathcal{I}_\ell(S_{\ell-1})]}{\sum_{j \in \mathcal{I}_\ell(S_{\ell-1})} \exp z_\theta^{(0)}(j, s_\ell)}, \tag{28}$$

$$\pi_\theta^{(1)}(\delta \mid s_\ell, i) = \frac{\exp z_\theta^{(1)}(\delta, s_\ell, i) \cdot \mathbf{1}[\delta \in \Delta_\ell(i; S_{\ell-1})]}{\sum_{\delta' \in \Delta_\ell(i; S_{\ell-1})} \exp z_\theta^{(1)}(\delta', s_\ell, i)}. \tag{29}$$

Since both masked distributions are proper probability distributions over their support, the score-function identity and the baseline subtraction argument in Section E.3 apply without modification.

E.5 Augmented Tchebycheff Scalarization

A plain weighted sum $\sum_k d_k c_k(b)$ can hide a catastrophic failure in one constraint behind gains in others. The augmented weighted Tchebycheff criterion [Dächert et al., 2012] addresses this by penalizing the worst-case weighted violation first:

$$R(b) = -\Big(\|Dc(b)\|_\infty + \rho \, \|Dc(b)\|_1 \Big) = -\Big(\max_k d_k c_k(b) + \rho \sum_k d_k c_k(b) \Big), \tag{30}$$

matching Eq. (11) with $D = \operatorname{diag}(d_u, d_s, d_r, d_g)$. The ℓ_∞ term dominates when any single violation is large; the ℓ_1 term with small $\rho > 0$ breaks ties among allocations sharing the same worst-case violation and avoids the weak-efficiency issue of a pure ℓ_∞ criterion [Dächert et al., 2012]. When all constraints are satisfied, $c(b) = 0$ and $R(b) = 0$.

E.6 Rank-Based Tier Mask

Let $\operatorname{rank}_u(\ell) \in \{1, \dots, L\}$ be the rank of block ℓ when blocks are sorted by decreasing u_ℓ (rank 1 is the most utility-sensitive block). A partition $\Gamma : \{1, \dots, L\} \to \{\text{high}, \text{mid}, \text{low}\}$ is *monotone* if

Table 13: Action-mask ablation on LLaMA2-7B under the W4 target. Lower PPL is better; higher zero-shot and SafetyBench scores are better.

Mask design	PPL ↓	0-shot ↑	SafetyBench ↑	AvgW
Overlapping tier	7.81	62.14	54.30	4.19
Adjacent tier	7.80	62.48	51.78	4.19
Lower-bound mask	8.03	60.76	50.07	4.19

$\mathrm{rank}_u(\ell_1) < \mathrm{rank}_u(\ell_2)$ implies that $\Gamma(\ell_1)$ is not in a lower tier than $\Gamma(\ell_2)$, i.e. higher-ranked blocks are never placed in a lower tier than lower-ranked blocks. We use the overlapping tier mask:

$$\mathcal{A}_\ell^{\mathrm{tier}} = \begin{cases} \{6,8\}, & \Gamma(\ell) = \mathrm{high}, \\ \{4,6,8\}, & \Gamma(\ell) = \mathrm{mid}, \\ \{3,4,6\}, & \Gamma(\ell) = \mathrm{low}. \end{cases} \tag{31}$$

The overlap between adjacent tiers is intentional: because utility and safety rankings are largely disjoint (Section 3), a strict partition would prevent safety-sensitive but utility-low-ranked blocks from receiving sufficient precision. Safety awareness enters through σ_ℓ in the state and through c_s, c_r in the reward; Γ is derived solely from u_ℓ and encodes a conservative utility-informed search prior.

Table 13 reports an ablation over three mask designs. The adjacent tier mask restricts each tier to neighboring candidates only ($\{6,8\}$, $\{4,6\}$, $\{3,4\}$); the lower-bound mask removes tiers and applies only a minimum-precision floor for high-ranked blocks. The overlapping mask achieves the best SafetyBench score while maintaining comparable utility and average bit width, and is used in all main experiments.

E.7 Budget-Feasibility Mask

Let $B_{\min} = L(\bar{b} - \epsilon_b)$ and $B_{\max} = L(\bar{b} + \epsilon_b)$. Define the sets of achievable suffix sums recursively:

$$\mathcal{T}_{L+1} = \{0\}, \qquad \mathcal{T}_\ell = \left\{ b' + t : b' \in \mathcal{A}_\ell^{\mathrm{tier}},\ t \in \mathcal{T}_{\ell+1} \right\}, \quad \ell = L, L-1, \ldots, 1. \tag{32}$$

Here $\mathcal{T}_\ell$ is the set of total bit widths that can be contributed by blocks $\ell, \ldots, L$ while respecting the tier mask at every step. At step ℓ with prefix sum $S_{\ell-1}$, the budget-admissible set is

$$\mathcal{A}_\ell^{\mathrm{bud}}(S_{\ell-1}) = \left\{ b' \in \mathcal{A}_\ell^{\mathrm{tier}} : \exists\, t \in \mathcal{T}_{\ell+1},\ B_{\min} \le S_{\ell-1} + b' + t \le B_{\max} \right\}, \tag{33}$$

and the admissible joint-action set is $\mathcal{U}_\ell(S_{\ell-1}) = \{(i,\delta) : g(i,\delta) \in \mathcal{A}_\ell^{\mathrm{bud}}(S_{\ell-1})\}$.

Lemma 1 (Budget feasibility). *Suppose $\mathcal{A}_1^{\mathrm{bud}}(0) \neq \varnothing$. If $b_\ell \in \mathcal{A}_\ell^{\mathrm{bud}}(S_{\ell-1})$ for every $\ell \in \{1, \ldots, L\}$, then*

$$B_{\min} \ \le\ \sum_{\ell=1}^{L} b_\ell \ \le\ B_{\max}, \qquad \textit{equivalently,} \qquad \left| \tfrac{1}{L} \sum_{\ell=1}^{L} b_\ell - \bar{b} \right| \le \epsilon_b.$$

Proof. Instantiate the hypothesis at $\ell = L$. Since $\mathcal{T}_{L+1} = \{0\}$, the unique element is $t = 0$, so

$$b_L \in \mathcal{A}_L^{\mathrm{bud}}(S_{L-1}) \implies \exists\, t \in \{0\} :\ B_{\min} \le S_{L-1} + b_L + t \le B_{\max} \implies B_{\min} \le S_{L-1} + b_L \le B_{\max}.$$

By definition of the prefix sum, $S_{L-1} + b_L = \sum_{\ell=1}^{L} b_\ell$. Therefore

$$B_{\min} \le \sum_{\ell=1}^{L} b_\ell \le B_{\max} \iff L(\bar{b} - \epsilon_b) \le \sum_{\ell=1}^{L} b_\ell \le L(\bar{b} + \epsilon_b) \iff \left| \tfrac{1}{L} \sum_{\ell=1}^{L} b_\ell - \bar{b} \right| \le \epsilon_b. \quad \square$$

Remark (no dead ends). The recursive construction of $\mathcal{T}_\ell$ guarantees the following stronger property: any $b' \in \mathcal{A}_\ell^{\mathrm{bud}}(S_{\ell-1})$ has at least one tier-feasible continuation $(b_{\ell+1}, \ldots, b_L)$ with $b_j \in \mathcal{A}_j^{\mathrm{tier}}$ for all $j > \ell$ that keeps the running total within $[B_{\min}, B_{\max}]$. This follows directly from the definition of $\mathcal{A}_\ell^{\mathrm{bud}}$: the existence of $t \in \mathcal{T}_{\ell+1}$ in (33) witnesses such a continuation. Therefore greedy step-by-step sampling from the masked distributions never reaches a state where no admissible action exists, provided $\mathcal{A}_1^{\mathrm{bud}}(0) \neq \varnothing$. In practice a cheaper interval-based approximation may be used in place of the exact recursive sets; any residual budget deviation is then captured and penalized by $c_g(b)$ in Eq. (10).

Algorithm 1 ACQueReLlo: primal-dual training for safety-aware mixed-precision quantization

Input: $\mathcal{M}_{\mathrm{fp16}},\ \mathcal{D}_{\mathrm{cal}},\ \mathcal{D}_u,\ \mathcal{D}_h,\ \mathcal{D}_b,\ \bar{b}$
Compute $u_\ell = \mathrm{Norm}(\Delta\mathrm{PPL}(\ell))$ and $\sigma_\ell = \mathrm{Norm}(|\Delta T_h(\ell)| + |\Delta T_b(\ell)|)$ for all ℓ via single-block quantization studies on $\mathcal{D}_{\mathrm{cal}}$
Construct monotone partition Γ; compute $\{\mathcal{A}_\ell^{\mathrm{tier}}\}$ and $\{\mathcal{T}_\ell\}$ (Sections E.6–E.7)
Initialize policy parameters θ, critic parameters ω, and dual variables $\lambda \in [0, \lambda_{\max}]^4$
Initialize best-feasible record $(b^*, R^*) \leftarrow (\varnothing, -\infty)$ and best-penalized record $(\hat{b}, \hat{R}) \leftarrow (\varnothing, -\infty)$
for each episode **do**
    Reset $S_0 \leftarrow 0$ and running statistics $d_\ell, r_\ell, m_\ell, h_\ell$
    for $\ell = 1$ to L **do**
        Build $s_\ell = (x_\ell^{\mathrm{stat}}, x_\ell^{\mathrm{dyn}})$ using Eqs. (4)–(8)
        Compute $\mathcal{A}_\ell^{\mathrm{bud}}(S_{\ell-1})$ via Eq. (33) and $\mathcal{U}_\ell(S_{\ell-1})$
        Sample $i_\ell \sim \pi_\theta^{(0)}(\cdot \mid s_\ell)$ and $\delta_\ell \sim \pi_\theta^{(1)}(\cdot \mid s_\ell, i_\ell)$ under the masks (Section E.4)
        Set $b_\ell = g(i_\ell, \delta_\ell)$; update $S_\ell \leftarrow S_{\ell-1} + b_\ell$
    end for
    Quantize $\mathcal{M}_b$; sample proxy batches from $\mathcal{D}_u, \mathcal{D}_h, \mathcal{D}_b$
    Compute $c(b)$ via Eq. (10), $R(b)$ via Eq. (11), $R_\lambda(b) = R(b) - \lambda^\top c(b)$
    if $c(b) = 0$ **and** $R(b) > R^*$ **then** $(b^*, R^*) \leftarrow (b, R(b))$
    end if
    if $R_\lambda(b) > \hat{R}$ **then** $(\hat{b}, \hat{R}) \leftarrow (b, R_\lambda(b))$
    end if
    Compute advantage $\Psi_\ell = R_\lambda(b) - B_\omega(s_\ell)$ for all ℓ
    Update θ by minimizing $J_{\mathrm{actor}}(\theta) = -\sum_\ell \Psi_\ell \log \pi_\theta(a_\ell \mid s_\ell)$ (Eq. (15))
    Update ω by minimizing $\sum_\ell (R_\lambda(b) - B_\omega(s_\ell))^2$; apply entropy bonus to θ update
    Update $\lambda \leftarrow \mathrm{clip}(\lambda + \eta\, c(b),\ 0,\ \lambda_{\max})$ (Eq. (16))
end for
return b^* if $b^* \neq \varnothing$, else $\hat{b}$

E.8 TRM Sign Convention

The raw margin $\mu(p; \mathcal{M}) = \max_{v \in \mathcal{C}} \log p(v \mid p; \mathcal{M}) - \max_{r \in \mathcal{R}} \log p(r \mid p; \mathcal{M})$ is negative when the model assigns higher likelihood to refusal templates and positive otherwise. The calibrated margin in Eq. (3) subtracts the input-independent bias. Under this convention, the violation terms in Eq. (10) have the following interpretations:

- $c_s = [\Delta T_h(b)/\tau_s]_+$ is positive when T_h *increases* after quantization, meaning the model becomes more likely to comply with harmful prompts (weaker refusal). This is the undesired direction.
- $c_r = [-\Delta T_b(b)/\tau_r]_+$ is positive when T_b *decreases* after quantization, meaning the model becomes more inclined to refuse benign prompts (over-refusal). This is also undesired.

Both terms are zero when quantization leaves refusal behavior unchanged or moves it in the desired direction.

E.9 Training Algorithm

Algorithm 1 gives the complete training loop. At the end of training, the best feasible allocation b^* (all constraints satisfied) is returned if one was found; otherwise the allocation $\hat{b}$ with the highest penalized return is returned.

Algorithm 1 presents the exact conceptual version of the masking rule in Eq. (33). A cheaper interval-based screening rule may be used in implementation, with any remaining deviation from the target average bit width captured by $c_g(\mathbf{b})$.

F Reproducibility Details

This section summarizes the experimental details needed to reproduce the reported results. The full method definition, notation, action mask, primal-dual update, and budget-feasibility rule are provided in Appendix E.

Models and quantization setting. The main experiments evaluate LLaMA3.1-8B, LLaMA2-7B, Qwen2-7B, and Mistral-7B-v0.1. Unless otherwise specified, all quantized baselines use the W4A4 setting. For ACQUERELLO, the candidate weight bit set is $\mathcal{B} = \{3, 4, 6, 8\}$ and the default target average weight precision is $\bar{b} = 4.0$. Activation precision is fixed to A4 in the reported low-bit experiments.

Evaluation protocol. We report perplexity (PPL) for language-modeling quality, the average zero-shot accuracy over 11 tasks for general utility, SafetyBench accuracy for safety knowledge, OR-Bench for over-refusal on benign prompts, and TRM for calibrated refusal preference. Higher zero-shot accuracy and SafetyBench accuracy are better, while lower PPL and lower OR-Bench are better. For TRM, lower values indicate a stronger calibrated refusal preference; this is desirable on harmful prompts but can indicate over-refusal on benign prompts.

Runs and variability. The main results are reported as mean $\pm$ standard deviation over three runs. For deterministic PTQ baselines, variability comes from calibration sampling. For ACQUERELLO, variability comes from both calibration sampling and policy randomness.

Baselines. We compare against RTN, GPTQ, DuQuant, QuaRot, SpinQuant, OSTQuant, and CAQ under the same W4A4 setting unless otherwise specified. All baselines use the same evaluation protocol as ACQUERELLO. When a baseline has method-specific calibration or transformation steps, we follow the corresponding public implementation or the setting described in the original paper.

Implementation details. The policy and value function share an MLP encoder and use the factorized action space described in Section 3. The detailed notation, masking rule, constrained policy-gradient formulation, and budget-feasibility guarantee are provided in Appendix E. The policy is trained with proxy batches sampled from the utility, harmful-prompt, and benign-prompt pools instead of with full benchmark evaluation at every episode. In the reported experiments, we instantiate the ordered candidate bit set as $\mathcal{B} = \{3, 4, 6, 8\}$ and use the default target average weight precision $\bar{b} = 4.0$.

Code release. We include an anonymized code package and configuration files in the supplementary material. The paper provides the algorithmic description, evaluation protocol, and implementation details needed to describe the reproduction path, and the supplementary package provides the commands and settings used to reproduce the main experiments.

F.1 Adversarial Evaluation

Table 14: HumanJailbreaks/HarmBench attack success rate (ASR, %) using the official classifier. Behavior-level ASR counts behaviors for which at least one attack succeeds. Lower is better. These prompts are excluded from sensitivity estimation, reward evaluation, policy training, and allocation selection.

Model	Method	Attack-instance ASR	Worst-case behavior ASR
LLaMA2-7B	Fine-tuned FP16	1.76	8.18
	Uniform W4A4	2.01	8.81
	ACQueReLlo	2.14	8.18
Qwen2-7B	Fine-tuned FP16	23.65	58.49
	Uniform W4A4	26.79	68.55
	ACQueReLlo	24.91	67.92

G Compute Resources

All experiments were conducted on a GPU cluster. Each quantization or policy-search run used a single NVIDIA L40S GPU with 48 GB of memory. For LLaMA2-7B, one end-to-end ACQUERELLO policy-search run required approximately 4 GPU hours, including single-block sensitivity estimation, proxy evaluation, and policy optimization. Baseline PTQ runs were less expensive and required approximately 0.5-1 GPU hours per method on a 7B model under the same evaluation setting. Larger

models and additional random seeds required proportionally more compute. Preliminary runs, failed runs, and pilot experiments are not included in these estimates.

Table 15: LLaMA2-7B runtime (GPU-hours) on one NVIDIA L40S (48 GB). Preparation includes data processing and block sensitivity estimation.

Preparation	Search	Final quantization and evaluation	Total	Uniform W4A4
0.104	1.568	0.953	2.625	0.961

H Assets and Licenses

Table 16: External assets used in this work. We cite the original sources and follow the corresponding licenses and terms of use.

Asset	Use in this paper	License / terms
LLaMA2	Evaluated model family	LLaMA2 Community License
LLaMA3.1	Evaluated model family	LLaMA3.1 Community License
Mistral-7B-v0.1	Evaluated model family	Apache License 2.0
Qwen2-7B	Evaluated model family	Apache License 2.0
WikiText-2	Language-modeling evaluation	CC BY-SA
SafetyBench	Safety evaluation	Official benchmark terms / repository license
OR-Bench	Over-refusal evaluation	Apache License 2.0; Huggingface for data
lm-evaluation-harness	Zero-shot evaluation	MIT License
PyTorch	Model execution and training	BSD-style license
Hugging Face Transformers	Model loading and inference	Apache License 2.0
GPTQ implementation	Quantization baseline	Apache License 2.0
QuaRot implementation	Quantization baseline	Apache License 2.0

I Limitations

Our experiments focus on four safety-aligned model families and several commonly used post-training quantization baselines. The results therefore do not cover all model families, safety-tuning recipes, or deployment settings. The safety evaluation is also based on benchmark accuracy, refusal-related proxy margins, and OR-Bench, which are useful but not complete measures of real-world safety. In addition, ACQueReLlo uses sampled proxy evaluations during policy training, so the learned allocation can depend on the choice of calibration data, harmful-prompt pool, benign-prompt pool, and budget target. Finally, the current implementation searches over block-level weight bit widths with fixed activation quantization, and finer-grained layer-wise or hardware-specific allocation may change the trade-off.

J Broader Impact

This work aims to make compression of safety-aligned language models more aware of safety-related behavior. A positive impact is that efficient deployment can be evaluated and optimized with both utility and safety in mind, instead of using utility metrics alone. A possible negative impact is that better compression methods can also lower the cost of deploying powerful language models, including in settings where monitoring or safety evaluation is incomplete. The method should therefore be used together with standard model release policies, safety evaluations, and deployment safeguards.

K LLM Usage Disclosure

LLMs were used as the subject models evaluated in this study. They were not used as an original or non-standard component of the proposed algorithm. Any language-model assistance, if used, was limited to writing, proofreading, or formatting and did not affect the scientific claims, methodology, or experimental results.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by the method formulation in Section 3 and the empirical results in Section 4. The claims are limited to the evaluated model families, benchmarks, and low-bit post-training quantization settings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the Limitations section, including the evaluated model scope, the reliance on benchmark and proxy safety measures, the dependence on sampled proxy evaluations, and the current focus on block-level weight bit allocation with fixed activation quantization.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The appendix provides the assumptions and proof for the budget-feasibility result associated with the action mask. Other mathematical expressions define the CMDP formulation, state, constraints, reward, and optimization procedure rather than formal theoretical guarantees.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the model families, quantization setting, policy state, action space, masking rule, reward, primal–dual update, baselines, evaluation metrics, and run-to-run variability in Section 3, Section 4, Appendix E, and Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will include an anonymized code package and configuration files with the NeurIPS supplementary material to reproduce the main experiments. The package will contain instructions for running the proposed method and evaluation scripts, while third-party models and datasets should be obtained from their original providers under the corresponding licenses and terms.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is described in Section 4, including the model families, baselines, detailed quantization setting, and evaluation metrics. Additional evaluation protocol, variability, baseline, and implementation details are provided in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Table 1 reports mean and standard deviation over three runs. For deterministic baselines, variability comes from calibration sampling; for learned methods, it comes from both calibration sampling and policy randomness.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix G reports the GPU type, GPU memory, approximate runtime for one ACQUERELLO policy-search run, approximate runtime for baseline PTQ runs, and notes that preliminary runs, failed runs, and pilot experiments are excluded from the estimates.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics, and the research conforms to it. The work studies safety-aware compression of existing LLMs and does not involve deception, human-subject experiments, or release of unsafe new datasets.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses broader impact in the Broader Impact section, including the positive impact of safety-aware efficient deployment and the possible negative impact that improved compression could lower the cost of deploying powerful language models without adequate monitoring.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained model, harmful prompt dataset, or scraped dataset. If quantized checkpoints are released, they will follow the original model licenses and usage restrictions.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Appendix H summarizes the external models, datasets, benchmarks, and software packages used in this work and lists their licenses or terms where available. We cite the original sources and do not redistribute third-party model weights, benchmark datasets, or external code.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new dataset, benchmark, pretrained model, or standalone released asset. The proposed contribution is an algorithmic framework and empirical evaluation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject studies.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the subject models evaluated in this work, but they are not used as an original or non-standard component of the proposed algorithm. Any language-model assistance, if used, was limited to writing, proofreading, or formatting and did not affect the scientific claims, methodology, or experimental results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.