CALIBRATE THE DECISIONS THAT CHANGE THE FUTURE: ON-POLICY POST-TRAINING QUANTIZATION FOR MULTIMODAL LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training quantization (PTQ) lowers deployment cost for multimodal large language models, but calibration typically reconstructs fixed sequences with local objectives. This overlooks autoregressive feedback: a quantization-induced token change redirects the prefix and changes future states. Yet on-policy coverage alone is insufficient because many decision mismatches barely affect future generation. We propose ONPTQ, an on-policy framework that calibrates on trajectories visited by the current quantized policy. On shared prefixes, ONPTQ identifies quantization-eroded boundaries, evaluates competing tokens through short counterfactual rollouts, and combines current discrepancy with branch consequence into a Decision–Consequence risk. The risk prioritizes critical states, while context anchoring and trajectory refresh preserve multimodal behavior and keep calibration aligned with the updated policy. We further derive a Decision–Consequence bound linking behavioral deviation to current policy discrepancy and action-conditioned future-value span. Across vision–language and omni-modal Qwen models under multiple low-bit settings, ONPTQ improves downstream performance and yields fewer correctness flips against the corresponding Dense/FP16 references, without changing the deployed inference graph.

1 INTRODUCTION

Large vision–language models (VLMs) have advanced multimodal understanding and generation (Bai et al., 2025b;a; Xu et al., 2025), but remain costly to deploy. Post-training quantization (PTQ) reduces this cost without full retraining through smoothing, learned transformations, rotations, and task-aware calibration (Xiao et al., 2023; Shao et al., 2024; Liu et al., 2025; Sun et al., 2025; Kim et al., 2025). VLM-specific methods further model modality imbalance, token and channel sensitivity, and precision allocation (Li et al., 2025; Xiang et al., 2026; Jia et al., 2026; Zhang et al., 2026; Hu et al., 2026). Despite this progress, calibration is still commonly performed on largely fixed image–text contexts selected independently of the prefixes produced by the quantized model. Existing objectives refine *what* to preserve at a given input, but not *which autoregressive states* should receive calibration effort.

Autoregressive generation makes this omission consequential. At step t, the state $s_t = (I, x, y_{<t})$ contains the model-generated prefix; a quantization-induced token change therefore alters the next state and all subsequent predictions. This creates a closed-loop calibration–deployment gap related to the train–generation mismatch studied in sequential learning (Ross et al., 2011; Bengio et al., 2015; Ranzato et al., 2016; Goyal et al., 2016). Yet collecting on-policy states alone is insufficient: many numerical discrepancies never change the trajectory. We call a state *decision-critical* when quantization erodes a competitive next-token boundary and the competing actions lead to meaningfully different futures. Our central question is therefore: how can PTQ follow the quantized policy while focusing calibration on the decisions that shape future generation?

Our diagnostics expose two coupled factors: state coverage and decision consequence. Fig. 1 shows that dense–quantized discrepancy amplifies after a generation fork, and that calibration on quantized-policy states better covers deployment. However, uniformly reconstructing these states does not by itself preserve downstream behavior, while local divergence only weakly reflects counterfactual

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

branch separation. Calibration must therefore combine on-policy coverage with the future consequence of each decision.

We propose OnPTQ (**On**-Policy **P**ost-**T**raining **Q**uantization), a framework that turns calibration into a *Follow–Compare–Branch–Calibrate* cycle, as illustrated in Fig. 2. OnPTQ first follows the current quantized policy to collect the states it actually visits. At each shared quantized prefix, it compares dense and quantized next-token decisions to identify mismatches and near-miss boundaries, then branches from the competing tokens through short counterfactual rollouts to estimate how strongly the alternatives separate in the future. Boundary erosion, same-state distribution divergence, and branch consequence together form a Decision–Consequence risk that prioritizes critical-state calibration, while a static context anchor preserves broader multimodal behavior. After each update, OnPTQ refreshes the trajectories under the new quantized policy, keeping the calibration distribution aligned with the model being deployed. We further derive a Decision–Consequence bound that connects behavioral deviation to current policy discrepancy and the span of action-conditioned future values. All additional computation is confined to offline calibration, leaving the deployed inference graph unchanged. Across multiple models and quantization settings, OnPTQ improves downstream performance and produces fewer correctness flips relative to the same Dense/FP16 reference. Our contributions are threefold:

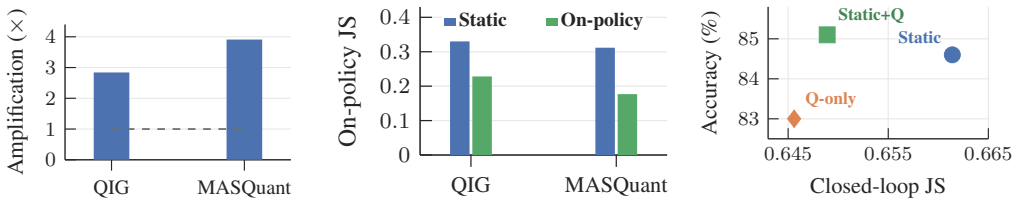
- We formulate autoregressive PTQ as an on-policy calibration problem and identify decision-critical states through controlled diagnostics of state coverage and future consequence under the quantized model's own generation distribution.
- We introduce OnPTQ, which combines shared-prefix decision comparison, short counterfactual branches, risk-weighted calibration, and iterative policy refresh, supported by a Decision–Consequence bound that separates current decision discrepancy from future consequence at each state along the generated trajectory.
- Extensive experiments across multiple VLM backbones and quantization settings show that OnPTQ consistently improves downstream performance while better preserving the behavior of the corresponding Dense/FP16 models.

2 Related Work

Post-training quantization for large language models. LLM PTQ improves dense-to-low-precision fidelity through activation-outlier handling and layer reconstruction, as in LLM.int8() and ZeroQuant, or through second-order and activation-aware weight quantization, as in OPTQ/GPTQ and AWQ (Dettmers et al., 2022; Yao et al., 2022; Frantar et al., 2023; Lin et al., 2024). SmoothQuant and OmniQuant learn scaling or equivalent transformations from calibration data (Xiao et al., 2023; Shao et al., 2024), while QuaRot, SpinQuant, LRQ, FlatQuant, and GuidedQuant introduce rotations, low-rank scaling, affine flattening, or end-loss guidance (Ashkboos et al., 2024; Liu et al., 2025; Lee et al., 2025; Sun et al., 2025; Kim et al., 2025). These methods primarily optimize numerical fidelity on fixed calibration distributions. OnPTQ instead targets autoregressive calibration by prioritizing states visited by the current quantized policy according to their behavioral risk.

Post-training quantization for multimodal large language models. Multimodal PTQ additionally models heterogeneous token statistics and cross-modal structure. Q-VLM captures intra- and inter-block dependencies, MBQ reweights reconstruction by modality sensitivity, and VLM-PTQ combines asymmetric correction with channel importance (Wang et al., 2024; Li et al., 2025; Deng & Huang, 2026). QIG, Quant Experts, MABA, and MASQuant further introduce token-level sensitivity, adaptive compensation, hardware-aware precision allocation, and modality-specific smoothing (Xiang et al., 2026; Jia et al., 2026; Zhang et al., 2026; Hu et al., 2026). This line refines how multimodal quantization error is represented; OnPTQ instead addresses an orthogonal autoregressive problem by drawing states from the current quantized policy and prioritizing them by decision-boundary erosion and branch consequence.

Autoregressive distribution shift and on-policy learning. Autoregressive models face distribution shift because predictions determine future inputs. DAgger, scheduled sampling, and sequence-level training expose models to policy-induced states or their own predictions (Ross et al., 2011; Bengio et al., 2015; Ranzato et al., 2016), while Professor Forcing aligns guided and free-running dynam-



(a) Post-fork amplification **(b)** On-policy coverage **(c)** Anchor trade-off

Figure 1: Motivating diagnostics for ONPTQ. (a) Quantization discrepancy amplifies after a decision fork. (b) On-policy calibration better covers deployment states. (c) A static anchor preserves downstream accuracy.

ics (Goyal et al., 2016). ImitKD and GKD apply related ideas to distillation on student-generated prefixes (Lin et al., 2020; Agarwal et al., 2024). These methods address training or distillation; ONPTQ transfers the on-policy principle to PTQ and further prioritizes visited states by counterfactual consequence rather than weighting the rollout uniformly.

3 Problem Setup and Motivating Observations

Autoregressive PTQ as state-distribution matching. Let D be a frozen dense VLM and Q_θ its quantized counterpart, where θ denotes the learnable quantization calibration parameters. Given an image–instruction pair (I, x), the quantized model induces

$$s_t = (I, x, y_{<t}), \qquad y_t \sim \pi_{Q_\theta}(\cdot \mid s_t), \qquad s_{t+1} = (s_t, y_t). \tag{1}$$

Here π_{Q_θ} is the next-token policy and d_{Q_θ} is its induced state distribution. We denote the unlabeled calibration set of image–instruction pairs by $\mathcal{D}_{\text{cal}}$, and abbreviate Q_θ, π_{Q_θ}, and d_{Q_θ} as Q, π_Q, and d_Q when θ is clear. Conventional reconstruction-based calibration instead optimizes

$$\min_\theta \ \mathbb{E}_{s\sim d_{\text{static}}}\big[\mathcal{L}_{\text{rec}}(D(s), Q_\theta(s))\big], \tag{2}$$

where $\mathcal{L}_{\text{rec}}$ is a static reconstruction objective and d_{static} is the state distribution fixed by the calibration corpus. Deployment occurs under d_{Q_θ}, which changes with θ and therefore turns calibration into a closed loop rather than fixed supervised regression.

All dense–quantized comparisons use a *shared quantized prefix*: Q_θ determines $y_{<t}$, and both models are queried at the resulting s_t. This isolates how quantization alters the next decision at a state the deployed model actually visits.

Observation I: local errors are amplified after a fork. We compare same-state JS divergence with the divergence accumulated after dense and quantized greedy generations first choose different tokens. Under QIG and MASQuant W4A8 checkpoints, post-fork divergence is respectively $2.84\times$ and $3.91\times$ the same-state divergence (Fig. 1(a)), showing across two quantizers that local numerical agreement understates the discrepancy induced by a changed prefix.

Observation II: the quantized policy visits a distinct calibration distribution. Fig. 1(b) evaluates divergence on quantized-policy states after calibration on either static or quantized-policy states. The latter reduces on-policy divergence by 30.8% for QIG and 43.3% for MASQuant, revealing that the deployed prefixes—not only the calibration examples—determine the relevant state distribution.

Observation III: on-policy coverage is necessary but not sufficient. A naive baseline that uniformly reconstructs quantized-policy trajectories delays the first dense–quantized fork by 3.54 tokens, but barely changes post-fork divergence and lowers accuracy by approximately 2.5 points. A static context anchor mitigates this trade-off, reducing closed-loop JS while maintaining accuracy comparable to static calibration (Fig. 1(c)). Thus, on-policy signals should complement, not replace, the multimodal context preserved by static calibration. Local dense–quantized JS also correlates only weakly with downstream branch effect (Pearson $r = 0.232$), so local-divergence ranking can miss subtle yet consequential decision boundaries.

Together, these findings motivate three requirements: use states from the current quantized policy, retain a static context anchor, and prioritize states by current decision damage and future consequence. ONPTQ implements these principles below.

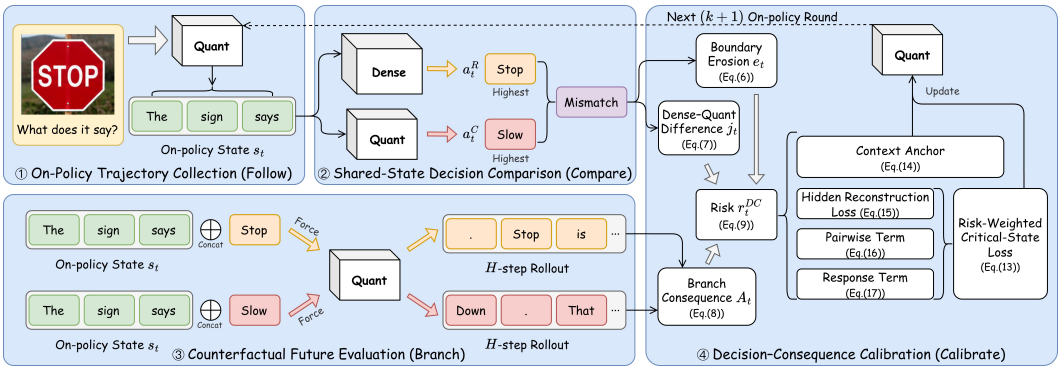


Figure 2: Overview of ONPTQ for a representative dense–quantized mismatch. **Follow** collects on-policy states, **Compare** identifies competing actions, **Branch** estimates future consequence, and **Calibrate** updates the quantizer before trajectory refresh.

4 ON-POLICY POST-TRAINING QUANTIZATION

4.1 OVERVIEW

Fig. 2 illustrates the ONPTQ cycle through a representative mismatch. The current quantized model supplies calibration states (**Follow**); the dense and quantized models are compared at each shared state to identify a reference–competitor fork (**Compare**), whose short counterfactual branches estimate future consequence (**Branch**). Boundary erosion, same-state distribution divergence, and branch consequence form a Decision–Consequence risk that weights critical-state calibration, while a context anchor preserves broader multimodal behavior (**Calibrate**). Updating θ starts a new on-policy round. Secs. 4.2 and 4.3 detail the calibration cycle and policy refresh, while Sec. 4.4 provides the Decision–Consequence analysis.

4.2 THE ONPTQ CALIBRATION CYCLE

Follow: on-policy trajectory collection. We begin by asking *which states should define calibration for the current quantized model*. In round k, $Q_{\theta^{(k)}}$ generates a response $y_{1:T}$ of length T for each $(I, x) \in \mathcal{D}_{\text{cal}}$, and every prefix induces an on-policy state $s_t = (I, x, y_{<t})$. We retain the exact generated token IDs so that subsequent dense–quantized comparisons reuse the same prefixes without retokenization drift. These states form the on-policy trajectory bank

$$\mathcal{B}^{(k)} = \left\{ s_t = (I, x, y_{<t}) \;\middle|\; (I, x) \in \mathcal{D}_{\text{cal}},\ y_{1:T} \sim \pi_{Q_{\theta^{(k)}}},\ 1 \le t \le T \right\},$$

which samples $d_{Q_{\theta^{(k)}}}$ and provides the shared states used by **Compare**.

Compare: shared-prefix decision comparison. We next ask *which next-token decisions have changed ordering, or are close to doing so, at states visited by the current quantized model*. Let $\mathcal{A}$ denote the vocabulary. At each $s_t \in \mathcal{B}^{(k)}$, we feed the same quantized on-policy state to both models and obtain the dense and quantized next-token logits z_D^t and z_Q^t over $\mathcal{A}$. We then define

$$a_t^R = \arg\max_a z_D^t(a), \qquad \hat{a}_t = \arg\max_a z_Q^t(a). \tag{3}$$

We define the competing action as

$$a_t^C = \arg\max_{a \neq a_t^R} z_Q^t(a). \tag{4}$$

The dense and quantized pairwise margins are

$$m_D^t = z_D^t(a_t^R) - z_D^t(a_t^C), \qquad m_Q^t = z_Q^t(a_t^R) - z_Q^t(a_t^C), \tag{5}$$

with boundary erosion

$$e_t = [m_D^t - m_Q^t]_+. \tag{6}$$

If $\hat{a}_t \neq a_t^R$, s_t is a *mismatch* and $a_t^C = \hat{a}_t$; if $\hat{a}_t = a_t^R$ but $e_t > 0$, it is a *near miss*: the top action is preserved while its margin over the strongest quantized runner-up has been eroded. We additionally compute

$$j_t = \mathrm{JS}(\pi_D(\cdot \mid s_t), \pi_Q(\cdot \mid s_t)) . \tag{7}$$

where JS is the Jensen–Shannon divergence. Mismatches and near misses form the candidate set $\mathcal{C}_{\theta^{(k)}} \subseteq \mathcal{B}^{(k)}$. Here, R denotes the token preferred by the dense reference model, whereas C denotes its strongest competing token under the current quantized model. These superscripts describe the tokens' relative roles in the decision comparison; they neither require task labels nor imply that either token is task-correct. We omit the round index when it is clear from context.

Branch: counterfactual consequence and risk. Among the candidate forks, we next ask *which quantization-affected next-token decisions lead to substantially different futures*. At each candidate state, we force a_t^R and a_t^C, then let the current quantized model greedily continue both branches over a small fixed horizon H. We use s_{t+h}^R and s_{t+h}^C to denote the resulting branch states at offset h. The dense model evaluates their predictive separation:

$$A_t = \frac{1}{H_t} \sum_{h=1}^{H_t} \mathrm{JS}\big(\pi_D(\cdot \mid s_{t+h}^R), \pi_D(\cdot \mid s_{t+h}^C)\big) , \tag{8}$$

where $H_t \leq H$ is the common valid horizon under early termination. Thus, Q generates both futures, while D measures whether they are behaviorally distinct.

We define the Decision–Consequence risk of state s_t as

$$r_t^{\mathrm{DC}} = e_t\big(1 + \lambda_b A_t\big) + \eta j_t. \tag{9}$$

Here λ_b and η balance the branch and distribution terms: A_t amplifies an eroded decision boundary, while j_t provides a full-vocabulary backstop. Given a critical-state budget K_c, OnPTQ selects the K_c states with the highest risk:

$$\mathcal{S}_{\mathrm{critical}} = \mathrm{TopK}_{s_t \in \mathcal{C}_\theta}\big(r_t^{\mathrm{DC}}\big) . \tag{10}$$

The selected risks are normalized as

$$\widetilde{w}_t = \frac{r_t^{\mathrm{DC}}}{\sum_{s_u \in \mathcal{S}_{\mathrm{critical}}} r_u^{\mathrm{DC}}} . \tag{11}$$

Calibrate: branch-aware calibration objective. Having identified the critical decision states, we ask *how to repair their quantization-eroded boundaries without disrupting the model's broader multimodal behavior*. OnPTQ separates these two roles: a context anchor preserves representations over the original multimodal context, while a risk-weighted loss concentrates calibration on selected on-policy decision states:

$$\mathcal{L}_{\mathrm{OnPTQ}} = \mathcal{L}_{\mathrm{anchor}} + \gamma \sum_{s_t \in \mathcal{S}_{\mathrm{critical}}} \widetilde{w}_t \, \mathcal{L}_{\mathrm{critical}}(s_t), \tag{12}$$

$$\mathcal{L}_{\mathrm{critical}}(s_t) = \mathcal{L}_{\mathrm{hidden}}(s_t) + \mathcal{L}_{\mathrm{pair}}(s_t) + \mathcal{L}_{\mathrm{response}}(s_t). \tag{13}$$

Here γ controls the overall contribution of critical-state calibration. For $M \in \{D, Q\}$, let $\mathcal{I}_{\mathrm{blk}}$ be the set of calibrated blocks and let $h_{M,\ell}^u \in \mathbb{R}^{d_\ell}$ be the hidden vector at token u after block $\ell \in \mathcal{I}_{\mathrm{blk}}$, where d_ℓ is the corresponding hidden width. Let $\mathcal{U}$ contain the image, instruction, and prompt-prefix anchor positions. With $\mathrm{MSE}(p, q) = \|p - q\|_2^2 / d_\ell$,

$$\mathcal{L}_{\mathrm{anchor}} = \frac{1}{|\mathcal{U}| \, |\mathcal{I}_{\mathrm{blk}}|} \sum_{u \in \mathcal{U}} \sum_{\ell \in \mathcal{I}_{\mathrm{blk}}} \mathrm{MSE}\big(h_{D,\ell}^u, h_{Q,\ell}^u\big) . \tag{14}$$

At critical state s_t, let $h_{M,\ell}(s_t)$ be the final-prefix hidden vector. Its reconstruction loss is

$$\mathcal{L}_{\mathrm{hidden}}(s_t) = \frac{1}{|\mathcal{I}_{\mathrm{blk}}|} \sum_{\ell \in \mathcal{I}_{\mathrm{blk}}} \mathrm{MSE}(h_{D,\ell}(s_t), h_{Q,\ell}(s_t)) . \tag{15}$$

For $\Delta m_t = m_Q^t - m_D^t$, the pairwise term uses the unit-threshold Huber loss (Huber, 1964):

$$\mathcal{L}_{\mathrm{pair}}(s_t) = \mathrm{Huber}(\Delta m_t) = \begin{cases} \frac{1}{2}(\Delta m_t)^2, & |\Delta m_t| \leq 1, \\ |\Delta m_t| - \frac{1}{2}, & |\Delta m_t| > 1. \end{cases} \tag{16}$$

Table 1: Accuracy (%, ↑) across precision settings. OCR: OCRBench; Viz: VizWiz; S-QA: ScienceQA; T-VQA: TextVQA. Avg: unweighted mean over the five benchmarks.

			Qwen2.5-VL-7B						Qwen3-VL-8B-Instruct					
Method	**Venue**	**Bits**	**MMMU**	**OCR**	**Viz**	**S-QA**	**T-VQA**	**Avg**	**MMMU**	**OCR**	**Viz**	**S-QA**	**T-VQA**	**Avg**
Dense	–	FP16	46.7	83.8	70.8	88.4	82.9	74.5	51.6	86.2	69.3	94.6	81.6	76.6
RTN	–	W4A16	43.3	83.7	67.8	81.3	82.1	71.6	51.6	75.8	70.5	92.5	80.3	74.1
SmoothQuant	ICML'23	W4A16	40.7	79.4	67.3	81.9	81.7	70.2	49.0	71.5	70.0	93.1	79.9	72.7
MBQ	CVPR'25	W4A16	44.4	82.8	70.6	87.8	**82.9**	73.7	50.6	73.2	70.0	93.8	80.8	73.7
QIG	CVPR'26	W4A16	48.2	82.6	70.5	**88.0**	82.5	**74.4**	51.8	74.0	69.7	93.5	80.1	73.8
MASQuant	CVPR'26	W4A16	44.4	**84.6**	**71.5**	87.8	82.5	74.2	49.4	**85.2**	69.2	92.9	**81.3**	75.6
ONPTQ	Ours	W4A16	**48.3**	83.2	70.1	87.1	81.9	74.1	**52.0**	85.1	70.7	94.3	81.1	**76.6**
RTN	–	W8A8	45.6	83.8	70.5	88.1	82.5	74.1	50.8	75.1	67.8	94.1	80.1	73.6
SmoothQuant	ICML'23	W8A8	43.3	83.8	70.0	88.2	82.6	73.6	51.4	75.8	67.9	**94.7**	80.2	74.0
MBQ	CVPR'25	W8A8	46.7	83.5	70.6	88.5	**82.9**	74.4	50.3	75.7	68.6	94.2	80.3	73.8
QIG	CVPR'26	W8A8	48.8	83.3	**71.1**	87.6	82.6	74.7	51.0	75.1	68.8	94.3	80.5	73.9
MASQuant	CVPR'26	W8A8	46.2	**84.2**	70.6	**88.6**	82.6	74.4	**52.0**	**86.0**	69.0	94.2	**81.3**	**76.5**
ONPTQ	Ours	W8A8	**50.3**	83.2	70.3	87.8	82.8	**74.9**	51.7	85.6	**69.9**	93.9	80.6	76.3
RTN	–	W4A8	43.3	68.3	63.2	85.2	76.9	67.4	50.6	76.3	70.0	92.2	79.6	73.8
SmoothQuant	ICML'23	W4A8	37.8	70.2	61.5	83.3	71.1	64.8	49.1	70.7	69.6	92.8	79.6	72.4
MBQ	CVPR'25	W4A8	43.3	74.1	64.3	**86.0**	74.8	68.5	**51.8**	74.5	69.0	92.5	80.6	73.7
QIG	CVPR'26	W4A8	40.4	64.9	58.6	83.2	73.5	64.1	50.6	75.2	70.6	**93.0**	80.1	73.9
MASQuant	CVPR'26	W4A8	43.3	72.8	**66.4**	85.7	77.0	69.0	51.2	**85.0**	69.5	92.1	80.8	75.7
ONPTQ	Ours	W4A8	**44.8**	**76.7**	64.8	85.8	**77.3**	**69.9**	49.1	84.8	**71.0**	**93.0**	**81.4**	**75.9**
RTN	–	W4A6	23.7	52.9	54.7	27.2	66.9	45.1	31.0	60.9	61.5	34.2	69.6	51.5
SmoothQuant	ICML'23	W4A6	28.9	67.7	60.5	78.6	69.3	61.0	46.2	66.6	**67.8**	88.0	77.1	69.1
MBQ	CVPR'25	W4A6	30.0	71.7	59.8	80.1	72.9	62.9	45.7	74.4	67.6	88.9	**78.9**	71.1
QIG	CVPR'26	W4A6	35.4	62.7	55.4	68.4	68.4	58.1	47.0	72.5	65.6	88.1	78.3	70.3
MASQuant	CVPR'26	W4A6	29.7	70.3	**62.6**	79.7	72.9	63.0	44.1	82.7	66.8	87.2	77.3	71.6
ONPTQ	Ours	W4A6	**37.7**	**72.8**	61.8	**81.9**	**74.9**	**65.8**	**48.8**	**83.0**	67.2	**89.7**	78.2	**73.4**

The response term preserves the full next-token distribution:

$$\mathcal{L}_{\text{response}}(s_t) = \mathrm{KL}(\pi_D(\cdot \mid s_t) \,\|\, \pi_Q(\cdot \mid s_t)) \,. \tag{17}$$

The anchor prevents broad context drift; the hidden and response terms stabilize each critical state; and the pairwise term restores the reference–competitor boundary without reconstructing every logit equally. Calibration requires neither task labels nor a reward model.

4.3 Iterative policy refresh

Because calibration changes both the quantized policy and its induced state distribution, $\mathcal{B}^{(k)}$ becomes off-policy once $\theta^{(k)}$ is updated to $\theta^{(k+1)}$. ONPTQ therefore refreshes the trajectory bank in every round:

$$Q_{\theta^{(k)}} \xrightarrow{\text{Follow}} \mathcal{B}^{(k)} \xrightarrow{\text{Compare, Branch, Calibrate}} Q_{\theta^{(k+1)}} \,. \tag{18}$$

The next round then recollects $\mathcal{B}^{(k+1)}$ under $Q_{\theta^{(k+1)}}$. Thus, the current quantized model always determines the visited prefixes and branch continuations, while the dense model remains a frozen reference.

4.4 Decision–Consequence Analysis

We formalize why behavioral risk depends jointly on current decision discrepancy and future consequence. For a generated trajectory τ, fixed continuation policy μ, and bounded future utility $G(\tau)$, define

$$V_\mu(s,a) = \mathbb{E}[G(\tau) \mid s, a, \tau_{>t} \sim \mu] \,, \tag{19}$$

$$J_{M\to\mu}(s) = \mathbb{E}_{a\sim\pi_M(\cdot|s)}[V_\mu(s,a)], \qquad M \in \{D, Q\}. \tag{20}$$

Theorem 1 (Decision–Consequence bound). At any shared state s,

$$|J_{Q\to\mu}(s) - J_{D\to\mu}(s)| \le \underbrace{\mathrm{TV}(\pi_Q(\cdot \mid s), \pi_D(\cdot \mid s))}_{\text{Decision}} \underbrace{\mathrm{span}_a \, V_\mu(s,\cdot)}_{\text{Consequence}}, \tag{21}$$

Table 2: Results on Qwen2.5-Omni under W4A8 and W4A6. Libri/Wen: WER (%, ↓) on LibriSpeech/WenetSpeech; MMMU/Omni: accuracy (%, ↑) on MMMU/OmniBench.

			Qwen2.5-Omni-3B				Qwen2.5-Omni-7B			
			Audio–Text		Vision–Text	Omni-modal	Audio–Text		Vision–Text	Omni-modal
Method	**Venue**	**Bits**	**Libri↓**	**Wen↓**	**MMMU↑**	**Omni↑**	**Libri↓**	**Wen↓**	**MMMU↑**	**Omni↑**
Dense	–	FP16	3.9	7.5	43.3	43.8	2.9	7.1	50.0	45.3
RTN	–	W4A8	109.7	105.6	28.9	29.7	9.0	8.7	42.2	35.2
SmoothQuant	ICML'23	W4A8	77.4	94.2	30.0	27.3	8.6	8.3	42.2	36.7
MBQ	CVPR'25	W4A8	9.5	8.5	27.8	36.7	3.8	8.2	47.8	40.6
QIG	CVPR'26	W4A8	8.3	8.5	28.8	37.0	3.6	8.2	41.9	41.1
MASQuant	CVPR'26	W4A8	3.6	8.7	**36.7**	41.4	2.9	8.0	**48.8**	43.8
ONPTQ	Ours	W4A8	**3.5**	**8.4**	36.4	**42.0**	**2.8**	**7.8**	43.3	**44.0**
RTN	–	W4A6	87.8	99.6	28.9	23.4	11.8	15.7	28.9	40.6
SmoothQuant	ICML'23	W4A6	87.8	99.5	28.9	18.8	5.8	19.8	37.8	35.9
MBQ	CVPR'25	W4A6	10.8	10.4	31.4	**46.9**	6.4	14.8	33.3	42.1
QIG	CVPR'26	W4A6	9.4	10.1	34.8	44.1	6.1	13.6	36.1	38.0
MASQuant	CVPR'26	W4A6	3.7	8.9	32.2	42.2	4.7	8.7	36.8	42.2
ONPTQ	Ours	W4A6	**3.5**	**8.5**	**35.4**	44.5	**4.5**	**8.3**	**38.1**	**43.4**

where $\mathrm{TV}(p,q) = \frac{1}{2}\sum_a |p(a) - q(a)|$ and $\mathrm{span}_a\, V_\mu = \max_a V_\mu(s,a) - \min_a V_\mu(s,a)$. The bound shows that either factor vanishes when the current policies agree or all actions have the same future value. It motivates, rather than certifies, Equation 9: e_t and j_t proxy decision discrepancy, while A_t proxies future consequence. The proof and two-action specialization appear in Appendix B.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUPS

Models and quantization. We evaluate ONPTQ on Qwen2.5-VL-7B, Qwen3-VL-8B-Instruct, and the 3B/7B Qwen2.5-Omni models (Bai et al., 2025b;a; Xu et al., 2025). Following prior multimodal PTQ protocols (Xiang et al., 2026; Hu et al., 2026), we quantize the language-model component while retaining the modality encoders at their original precision. We consider W4A16, W8A8, W4A8, and W4A6 for vision–language models, and the two settings, W4A8 and W4A6, for omni-modal models. Here, Wb_wAb_a denotes b_w-bit weights and b_a-bit activations.

Calibration datasets. Following QIG (Xiang et al., 2026), all matched vision–language methods share a fixed manifest of 128 image–caption pairs randomly drawn from the ShareGPT4V-improved COCO Caption dataset. ONPTQ uses these common prompts as initial contexts and collects calibration states by rolling out the current quantized model.

Benchmarks and metrics. We compare against RTN, AWQ (Lin et al., 2024), SmoothQuant (Xiao et al., 2023), MBQ (Li et al., 2025), QIG (Xiang et al., 2026), and MASQuant (Hu et al., 2026). Vision–language performance is evaluated on MMMU, OCRBench, VizWiz, ScienceQA, and TextVQA using task accuracy (Yue et al., 2024; Liu et al., 2024; Gurari et al., 2018; Lu et al., 2022; Singh et al., 2019). For Qwen2.5-Omni, we additionally report word error rate on LibriSpeech and WenetSpeech and accuracy on OmniBench (Panayotov et al., 2015; Zhang et al., 2022; Li et al., 2024). The vision–language average is the unweighted mean over its five benchmarks. We also report sample-wise Flip Rate against the same Dense/FP16 reference for the closed-choice MMMU and ScienceQA-IMG evaluations. Full calibration, decoding, provenance, and evaluation details are provided in Appendix C.1; hyperparameter settings and sensitivity are reported in Appendix C.2.

5.2 MAIN RESULTS

Vision-Language MLLMs. Tab. 1 compares Dense/FP16 and four quantization settings on Qwen2.5-VL-7B and Qwen3-VL-8B-Instruct (Bai et al., 2025b;a). ONPTQ achieves the highest average accuracy in six settings and leads on both backbones under W4A6. Its advantage is most consistent at lower activation precision, while remaining competitive under W4A16 and W8A8.

Vision-Audio-Language MLLMs. Tab. 2 evaluates Qwen2.5-Omni-3B and Qwen2.5-Omni-7B (Xu et al., 2025) under W4A8 and W4A6. Across all four settings, ONPTQ gives the lowest

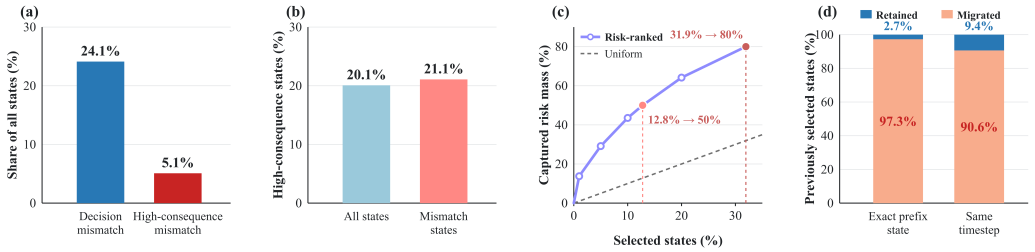


Figure 3: Properties of Decision–Consequence risk. (a) High-consequence mismatches are sparse. (b) Mismatch weakly enriches future consequence. (c) Risk mass concentrates in few states. (d) Selected states migrate after one OnPTQ update.

WER on both speech benchmarks. Under W4A6, it also achieves the highest MMMU accuracy at both model scales, while leading OmniBench in three of four combinations.

5.3 Analysis

Decision mismatch is overinclusive. Fig. 3(a) shows that top-1 mismatches cover 24.1% of on-policy states, whereas states that are both mismatches and in the top consequence quintile account for only 5.1%. This gap motivates filtering mismatches by future consequence.

Mismatch weakly predicts future consequence. In Fig. 3(b), high-consequence states constitute 20.1% of all states and 21.1% of mismatch states, indicating that current token disagreement alone provides little information about future branch divergence.

Risk is concentrated in few states. Fig. 3(c) shows that 12.8% and 31.9% of states capture 50% and 80% of the total risk, respectively, supporting selective risk weighting over uniform calibration.

Selected states migrate after model updates. After one OnPTQ update, Fig. 3(d) shows retention rates of only 2.7% under exact-prefix matching and 9.4% under prompt–timestep matching, motivating iterative trajectory refresh.

Together, these results support consequence-aware filtering, selective risk weighting, and iterative on-policy refresh. Detailed protocols are provided in Appendices D.2, D.3, and D.4.

Does counterfactual consequence improve state selection? We compare fixed-budget selectors from the same checkpoint while holding the calibration data and optimization fixed. Mismatch-only uses top-1 disagreements; Mismatch + Near Miss ranks both candidate types by $e_t + \eta j_t$; and OnPTQ further incorporates counterfactual consequence A_t through Eq. 9. The complete protocol is given in Appendix D.5.

Table 3: Fixed-budget selection on held-out joint-critical states.

Selection	Mean$(e_t)\downarrow$	$\Pr(m_Q^t < 0)\downarrow$
Dense–InitQ	2.322	69.23
Mismatch-only	2.446	61.54
Mismatch + Near Miss	2.144	58.97
OnPTQ	**1.609**	**53.85**

Tab. 3 reports the mean boundary erosion $\mathrm{Mean}(e_t)$ and the fraction of reference–competitor pairs whose quantized margin flips sign, $\Pr(m_Q^t < 0)$. Lower is better for both. OnPTQ achieves the lowest values, showing that counterfactual consequence improves state allocation beyond local decision signals.

Behavior preservation beyond aggregate accuracy. Accuracy can obscure sample-level correctness changes when opposing transitions cancel. We therefore report *Flip Rate* in Fig. 4, measuring the fraction of samples whose correctness differs from the same Dense/FP16 reference. Across all three evaluated settings, OnPTQ produces fewer correctness flips than the compared PTQ methods, more closely preserving the Dense/FP16 model's sample-wise correctness behavior.

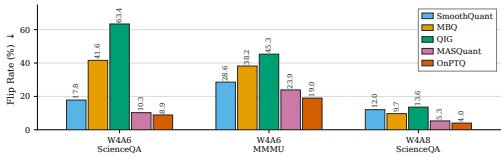


Figure 4: Sample-wise Flip Rate relative to the same Dense/FP16 reference on Qwen3-VL-8B-Instruct (lower is better).

5.4 Ablation Studies

Tab. 4 changes one design choice at a time under the same W4A6 protocol.

Table 4: Core ablations on Qwen3-VL-8B-Instruct (W4A6); accuracy (↑).

Setting	MMMU	ScienceQA
State Selection		
All on-policy states	45.4	86.8
Mismatch-only	47.1	88.3
Risk Composition		
Boundary only: e_t	44.9	86.5
w/o consequence A_t	45.7	87.3
w/o distribution term j_t	47.5	88.6
KL instead of JS	48.0	89.0
Loss Composition		
w/o $\mathcal{L}_{\text{anchor}}$	44.3	85.9
w/o $\mathcal{L}_{\text{hidden}}$	46.6	87.8
w/o $\mathcal{L}_{\text{pair}}$	45.5	87.1
w/o $\mathcal{L}_{\text{response}}$	47.0	88.2
Policy Refresh		
Only one round	45.9	87.4
2 rounds, no refresh	46.8	88.3
5 rounds, refresh	44.2	86.1
ONPTQ	**48.8**	**89.7**

State selection and risk composition. Using all on-policy states without candidate filtering performs worst among the selection variants, while mismatch-only selection recovers part of the gap. The full method further improves accuracy to 48.8 on MMMU and 89.7 on ScienceQA, supporting the inclusion of near misses. Boundary erosion alone is insufficient: removing A_t costs 3.1 and 2.4 points, whereas removing j_t costs 1.3 and 1.1 points. Thus, future consequence provides the larger gain, while the same-state distribution term remains complementary. Replacing JS with KL is competitive but remains below the full method.

Loss composition and policy refresh. Removing any loss term lowers accuracy on both datasets. The context anchor has the largest effect, followed by the pairwise boundary term, while hidden and response reconstruction provide additional gains. One round reaches 45.9 and 87.4. A second round without trajectory refresh improves these values to 46.8 and 88.3, but remains below the full method with refreshed trajectories (48.8 and 89.7), isolating the benefit of recollecting states under the updated quantized policy. Increasing the procedure to five refreshed rounds instead reduces accuracy to 44.2 and 86.1, showing that more rounds are not monotonically beneficial.

5.5 Inference Efficiency

Following the prefill protocol of MASQuant (Hu et al., 2026), we compare FP16 and W4A8 execution on a single NVIDIA B200 using the same model, sequence length, and runtime configuration. As shown in Tab. 5, W4A8 yields $1.79\times$ and $1.41\times$ prefill speedups at batch sizes 1 and 8, while reducing peak memory by 37.4% and 28.7%, respectively. All additional computation introduced by ONPTQ— on-policy trajectory collection, shared-state comparison, counterfactual rollout, and risk scoring—is confined to offline calibration. At inference time, only the calibrated W4A8 model is retained; neither the dense reference nor any branching or risk-estimation module is executed. Consequently, ONPTQ leaves the W4A8 inference graph and runtime cost unchanged while retaining its efficiency gains over FP16.

Table 5: End-to-end prefill performance of Qwen2.5-VL-7B on a single NVIDIA B200 (sequence length $= 2048$). BS: batch size.

Metric	BS	FP16	W4A8
Peak Memory (GB)	1	16.22	10.15
Prefill (ms)	1	36.73	20.48
Peak Memory (GB)	8	21.13	15.06
Prefill (ms)	8	271.39	192.76

6 Conclusion

This paper reframes post-training quantization for autoregressive MLLMs as an on-policy calibration problem. ONPTQ uses shared-prefix comparison and short counterfactual branches to prioritize decision-critical states, while context anchoring and trajectory refresh maintain calibration coverage. Its Decision–Consequence analysis links current policy discrepancy to future consequence. Across vision–language and omni-modal Qwen models, ONPTQ improves low-bit performance and reduces correctness flips without changing the inference graph. Overall, autoregressive PTQ should prioritize errors that alter future generation over numerical discrepancy alone.

AI Assistance Disclosure

Generative AI tools were used to draft selected passages, improve writing clarity,formulating mathematical claims and retrieve and identify relevant literature. All AI-assisted text and suggested references were reviewed, revised, and verified by the authors against the original sources. The authors take full responsibility for the final content of this paper.

References

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. GKD: Generalized knowledge distillation for auto-regressive sequence models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3180.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1171–1179, 2015.

Juncan Deng and Kejie Huang. VLM-PTQ: Efficient post-training quantization for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24696–24705, 2026.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2198.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4601–4609, 2016.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.

Lulu Hu, Wenhu Xiao, Xin Chen, Xinhua Xu, Bowen Xu, Kun Li, and Yongliang Tao. MASQuant: Modality-aware smoothing quantization for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8707–8716, 2026.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732.

Chenwei Jia, Baoting Li, Xuchong Zhang, Mingzhuo Wei, Bochen Lin, and Hongbin Sun. Quant experts: Token-aware adaptive error reconstruction with mixture of experts for large vision-language models quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24716–24726, 2026.

Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens J. S. Schaefer, Deokjae Lee, Yeonhong Park, Jae W. Lee, and Hyun Oh Song. GuidedQuant: Large language model quantization via exploiting end loss guidance. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 30011–30037, 2025.

Jung Hyun Lee, Jeonghoon Kim, June Yong Yang, Se Jung Kwon, Eunho Yang, Kang Min Yoo, and Dongsoo Lee. LRQ: Optimizing post-training quantization for large language models by learning low-rank weight-scaling matrices. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7708–7743. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.393.

Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. MBQ: Modality-balanced quantization for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4167–4177, 2025.

Yizhi Li, Ge Zhang, Yinghao Ma, Ruibin Yuan, Kang Zhu, Hangyu Guo, Yiming Liang, Jiaheng Liu, Zekun Wang, Jian Yang, et al. OmniBench: Towards the future of universal omni-language models, 2024.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6121–6133. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.494.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

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*, 2025. URL https://openreview.net/forum?id=ogO6DGE6FZ.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2507–2521, 2022.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5206–5210, 2015.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06732.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635, 2011.

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 *International Conference on Learning Representations*, 2024.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 57587–57613, 2025.

Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-VLM: Post-training quantization for large vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ziwei Xiang, Fanhu Zeng, Hongjian Fang, Rui-Qi Wang, Renxing Chen, Yanan Zhu, Yi Chen, Peipei Yang, and Xu-Yao Zhang. Fine-grained post-training quantization for large vision language models with quantization-aware integrated gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3500–3510, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2.5-Omni technical report, 2025. URL `https://arxiv.org/abs/2503.20215`.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1970.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, et al. WenetSpeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6182–6186, 2022.

Xi Zhang, Hanwei Zhu, Jiamang Wang, Xiaolin Wu, and Weisi Lin. Modality-aware bit allocation for mixed-precision quantization of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 9305–9315, 2026.

Appendix organization. Appendix A presents the complete OnPTQ calibration algorithm, and Appendix B provides the theoretical derivations. Appendix C details the implementation, evaluation protocol, and hyperparameters, while Appendix D reports supplementary analyses that support the motivating observations and state-selection design. Appendix E discusses the limitations.

Reproducibility Statement

The paper provides the complete calibration algorithm in Appendix A, the proof and surrogate correspondence in Appendix B, and the calibration manifest, hyperparameters, decoding protocol, and metric definitions in Appendix C. Local comparisons use fixed prompts, decoding settings, sample identities, and Dense/FP16 references.

A OnPTQ Algorithm

Algorithm 1 summarizes the complete offline calibration procedure. The current quantized checkpoint generates every response prefix and every counterfactual continuation, while the frozen dense model supplies the reference action and evaluates branch separation. The routine updates the quantization calibration parameters θ.

In round k, **Follow** first constructs the on-policy trajectory bank $\mathcal{B}^{(k)}$ defined in Sec. 4.2. **Compare** then extracts

$$\mathcal{C}_{\theta^{(k)}} = \left\{ s_t \in \mathcal{B}^{(k)} \;\middle|\; s_t \text{ is a mismatch or near miss} \right\}.$$

For each $s_t \in \mathcal{C}_{\theta^{(k)}}$, **Branch** stores the reference–competitor token pair and the resulting Decision–Consequence risk. The TopK step ranks these states by r_t^{DC} to obtain $\mathcal{S}^{(k)}_{\mathrm{critical}}$ for calibration.

Algorithm 1 On-Policy Post-Training Quantization

Require: Frozen dense model D; initial quantized model $Q_{\theta^{(0)}}$; unlabeled calibration set $\mathcal{D}_{\mathrm{cal}}$; rounds R; branch horizon H; critical-state budget K_c
Ensure: Calibrated quantized model $Q_{\theta^{(R)}}$

for $k = 0, \ldots, R-1$ **do**
  $\mathcal{B}^{(k)} \leftarrow \varnothing$ ▷ all on-policy states
  $\mathcal{C}_{\theta^{(k)}} \leftarrow \varnothing$ ▷ candidate states
  for each $(I, x) \in \mathcal{D}_{\mathrm{cal}}$ **do**
    $y_{1:T} \leftarrow \textsc{GreedyRollout}(Q_{\theta^{(k)}}, I, x)$
    Store the exact generated token IDs $y_{1:T}$
    for $t = 1, \ldots, T$ **do**
      $s_t \leftarrow (I, x, y_{<t})$
      $\mathcal{B}^{(k)} \leftarrow \mathcal{B}^{(k)} \cup \{s_t\}$
      Evaluate z_D^t and z_Q^t at the same state s_t
      $a_t^R \leftarrow \arg\max_a z_D^t(a)$; $a_t^C \leftarrow \arg\max_{a \neq a_t^R} z_Q^t(a)$
      if s_t is a mismatch or near miss **then**
        Compute e_t and j_t using Equations 6–7
        $\tau_t^R \leftarrow \textsc{BranchRollout}(Q_{\theta^{(k)}}, s_t, a_t^R, H)$
        $\tau_t^C \leftarrow \textsc{BranchRollout}(Q_{\theta^{(k)}}, s_t, a_t^C, H)$
        Compute A_t with frozen D using Equation 8
        $r_t^{\mathrm{DC}} \leftarrow e_t(1 + \lambda_b A_t) + \eta j_t$
        $\mathcal{C}_{\theta^{(k)}} \leftarrow \mathcal{C}_{\theta^{(k)}} \cup \{s_t\}$
        Store $(a_t^R, a_t^C, r_t^{\mathrm{DC}})$ for s_t
      end if
    end for
  end for
  $\mathcal{S}^{(k)}_{\mathrm{critical}} \leftarrow \mathrm{TopK}_{K_c}(\mathcal{C}_{\theta^{(k)}}, r_t^{\mathrm{DC}})$
  Normalize r_t^{DC} over $\mathcal{S}^{(k)}_{\mathrm{critical}}$ to obtain $\widetilde{w}_t$
  Form $\mathcal{L}_{\mathrm{OnPTQ}}$ using Equations 12–13
  $\theta^{(k+1)} \leftarrow \textsc{BasePTQUpdate}(\theta^{(k)}, \mathcal{L}_{\mathrm{OnPTQ}})$
end for
return $Q_{\theta^{(R)}}$

B Additional Derivations

B.1 Proof of Theorem 1

Fix a shared state s. Since $G(\tau)$ is bounded, $V_\mu(s,a)$ is finite for every $a \in \mathcal{A}$; since $\mathcal{A}$ is finite, the extrema

$$V_{\max}(s) = \max_{a\in\mathcal{A}} V_\mu(s,a), \qquad V_{\min}(s) = \min_{a\in\mathcal{A}} V_\mu(s,a) \tag{22}$$

exist. Define $\Delta\pi(a) = \pi_Q(a \mid s) - \pi_D(a \mid s)$. From the definition of the hybrid return,

$$J_{Q\to\mu}(s) - J_{D\to\mu}(s) = \sum_{a\in\mathcal{A}} \Delta\pi(a) V_\mu(s,a). \tag{23}$$

Both policies are normalized, so $\sum_{a\in\mathcal{A}} \Delta\pi(a) = 0$. Hence, for any constant c,

$$\sum_{a\in\mathcal{A}} \Delta\pi(a) V_\mu(s,a) = \sum_{a\in\mathcal{A}} \Delta\pi(a)\big(V_\mu(s,a) - c\big). \tag{24}$$

Choose the midpoint $c = (V_{\max}(s) + V_{\min}(s))/2$. Then

$$|V_\mu(s,a) - c| \le \frac{V_{\max}(s) - V_{\min}(s)}{2} = \frac{\operatorname{span}_a V_\mu(s,\cdot)}{2} \tag{25}$$

for every action. Applying the triangle inequality to Equation 24 gives

$$|J_{Q\to\mu}(s) - J_{D\to\mu}(s)| \le \sum_{a\in\mathcal{A}} |\Delta\pi(a)| \frac{\operatorname{span}_a V_\mu(s,\cdot)}{2} \tag{26}$$

$$= \mathrm{TV}(\pi_Q(\cdot \mid s), \pi_D(\cdot \mid s))\, \operatorname{span}_a V_\mu(s,\cdot), \tag{27}$$

where the final equality uses $\mathrm{TV}(\pi_Q(\cdot \mid s), \pi_D(\cdot \mid s)) = \frac{1}{2}\sum_{a\in\mathcal{A}} |\Delta\pi(a)|$. This proves Equation 21. The argument also covers the degenerate cases $\pi_Q = \pi_D$ or $\operatorname{span}_a V_\mu(s,\cdot) = 0$, for which both sides reduce appropriately. □

B.2 Pairwise Exact Decomposition

Restrict the action set at state s_t to the reference–competitor pair $\mathcal{P}_t = \{a_t^R, a_t^C\}$. For $M \in \{D, Q\}$, define the pair-normalized policy

$$\widetilde{\pi}_M(a \mid s_t) = \frac{\pi_M(a \mid s_t)}{\pi_M(a_t^R \mid s_t) + \pi_M(a_t^C \mid s_t)}, \qquad a \in \mathcal{P}_t. \tag{28}$$

Let $q_M^t = \widetilde{\pi}_M(a_t^R \mid s_t)$. Since $\widetilde{\pi}_M(a_t^C \mid s_t) = 1 - q_M^t$, the pair-restricted return is

$$J_{M\to\mu}^{\mathrm{pair}}(s_t) = q_M^t V_\mu(s_t, a_t^R) + (1 - q_M^t) V_\mu(s_t, a_t^C). \tag{29}$$

Subtracting the dense and quantized returns yields

$$\left|J_{Q\to\mu}^{\mathrm{pair}}(s_t) - J_{D\to\mu}^{\mathrm{pair}}(s_t)\right| = \left|\widetilde{\pi}_Q(a_t^R \mid s_t) - \widetilde{\pi}_D(a_t^R \mid s_t)\right| \left|V_\mu(s_t, a_t^R) - V_\mu(s_t, a_t^C)\right|. \tag{30}$$

Thus, on two actions, behavioral deviation decomposes exactly into shifted probability mass and an action-conditioned future-value difference.

For a softmax policy, the pairwise margin is also the pair-normalized log odds:

$$m_M^t = z_M^t(a_t^R) - z_M^t(a_t^C) = \log \frac{\widetilde{\pi}_M(a_t^R \mid s_t)}{\widetilde{\pi}_M(a_t^C \mid s_t)}. \tag{31}$$

Consequently, $m_Q^t < m_D^t$ implies that quantization shifts pair-normalized probability mass from the reference action toward the competitor. The boundary erosion $e_t = [m_D^t - m_Q^t]_+$ measures the magnitude of this directional log-odds shift.

B.3 From Theoretical Factors to OnPTQ Surrogates

The theorem is expressed in terms of true action distributions and action-conditioned future values. OnPTQ instantiates these factors with label-free quantities available during calibration:

Table 6: Correspondence between the Decision–Consequence analysis and the computable quantities used by OnPTQ.

Theoretical object	OnPTQ instantiation
On-policy state $s_t \sim d_Q$	Quant-policy trajectory collection
Decision discrepancy	Boundary erosion e_t and same-state JS j_t
Future consequence	Counterfactual branch amplification A_t
Policy-induced state shift	Iterative policy refresh

Boundary erosion focuses on the selected reference–competitor pair, while j_t detects changes elsewhere in the full next-token distribution. The inaccessible future-value difference in Equation 30 is represented by A_t, which measures how far the two forced-token branches separate over a short quantized-model rollout. These quantities yield the practical risk in Equation 9; the theorem supplies its decision–consequence structure, while the surrogate quality is evaluated empirically.

The statewise result becomes an on-policy objective by taking expectation over the states induced by the quantized model:

$$\mathcal{R}_{\mathrm{on}}(Q) = \mathbb{E}_{s_t \sim d_Q}\left[\mathrm{TV}(\pi_Q(\cdot \mid s_t), \pi_D(\cdot \mid s_t))\ \mathrm{span}_a\, V_\mu(s_t, \cdot)\right]. \tag{32}$$

Static PTQ changes both factors in Equation 32 only indirectly and evaluates them under a different state distribution. OnPTQ instead samples d_Q explicitly and constructs separate surrogates for the decision and consequence terms.

C Experimental Setup and Hyperparameters

C.1 Implementation and evaluation protocol

Hardware. All experiments were conducted on a server equipped with eight NVIDIA B200 GPUs. The inference-efficiency measurements in Sec. 5.5 use a single B200, as specified there.

Quantization scope. Only the language-model component is quantized; the visual and audio encoders remain at their original precision. W4A16 denotes 4-bit weights with 16-bit activations, whereas W8A8, W4A8, and W4A6 jointly quantize weights and activations. We use per-channel weight quantization and dynamic per-token activation quantization.

Result provenance. For Qwen2.5-VL-7B, the available Dense and baseline results in Tab. 1 are transcribed from the original MASQuant paper; the OnPTQ rows are evaluated separately under the settings reported here. For Qwen3-VL-8B-Instruct, we reproduce the compared methods from their released open-source implementations and evaluate them with a common local pipeline.

Calibration datasets. Following QIG (Xiang et al., 2026), all matched vision–language runs use the same frozen 128-pair manifest. We sample uniformly without replacement from 50,027 ShareGPT4V-improved COCO Caption records using seed 26260725. The COCO image ID is the stable sample identifier, whereas the ordinal denotes its zero-based position in the frozen manifest. The complete manifest is listed below; each pair is formatted with the conversational template of the target model (Tab. 7).

Table 7: COCO image IDs of the frozen 128-sample vision–language calibration manifest. Ord. denotes the zero-based manifest order.

Ord.	COCO image ID	Ord.	COCO image ID	Ord.	COCO image ID	Ord.	COCO image ID
0	000000039043	32	000000127647	64	000000222468	96	000000069410
1	000000153506	33	000000201420	65	000000044928	97	000000170040
2	000000098431	34	000000071044	66	000000117289	98	000000007232
3	000000130270	35	000000100034	67	000000235642	99	000000091912
4	000000016706	36	000000187474	68	000000211041	100	000000034785
5	000000173142	37	000000207597	69	000000066866	101	000000108531
6	000000082142	38	000000215303	70	000000230177	102	000000022051
7	000000073141	39	000000042743	71	000000245733	103	000000103331
8	000000009807	40	000000088286	72	000000211919	104	000000057801
9	000000038685	41	000000003737	73	000000167264	105	000000068120
10	000000146570	42	000000028540	74	000000239532	106	000000032523
11	000000076590	43	000000220097	75	000000140073	107	000000240033
12	000000110547	44	000000239499	76	000000120241	108	000000154354
13	000000227540	45	000000062307	77	000000056233	109	000000200483
14	000000227802	46	000000031747	78	000000220182	110	000000105688
15	000000166320	47	000000038902	79	000000164042	111	000000130287
16	000000160103	48	000000109278	80	000000048491	112	000000049819
17	000000172467	49	000000068359	81	000000189957	113	000000146504
18	000000218772	50	000000152823	82	000000116088	114	000000201632
19	000000194034	51	000000242734	83	000000215708	115	000000202387
20	000000176414	52	000000190172	84	000000041818	116	000000095595
21	000000047935	53	000000149588	85	000000158726	117	000000077750
22	000000014766	54	000000025060	86	000000043555	118	000000035299
23	000000059540	55	000000049363	87	000000028097	119	000000119979
24	000000221346	56	000000195188	88	000000170211	120	000000103722
25	000000105804	57	000000118302	89	000000000599	121	000000046936
26	000000205473	58	000000125661	90	000000063307	122	000000204805
27	000000187866	59	000000112577	91	000000081906	123	000000217376
28	000000229852	60	000000208886	92	000000052386	124	000000206469
29	000000090683	61	000000094544	93	000000186606	125	000000067814
30	000000134375	62	000000209925	94	000000105899	126	000000099030
31	000000113722	63	000000115912	95	000000103267	127	000000179793

For the matched Qwen2.5-Omni-7B runs, the frozen 128-example calibration manifest contains OmniBench IDs 0–127.

OnPTQ calibration. All matched runs start from the same initialized quantized checkpoint, and OnPTQ updates only the quantization calibration parameters θ. Unless otherwise stated, we use greedy on-policy rollouts with at most 64 response tokens, two rounds with trajectory refresh, branch horizon $H = 4$, critical-state budget $K_c = 256$, $\lambda_b = \eta = 1.0$, and $\gamma = 0.2$. The full procedure is given in Algorithm 1, and the sensitivity of these choices is reported in Appendix C.2.

Evaluation protocol. Methods within each matched comparison use identical task prompts, decoding, and evaluation samples. The local Qwen2.5-Omni-7B evaluation uses 900 MMMU validation examples. Tab. 2 reports the 1,014-example OmniBench evaluation split after excluding calibration IDs 0–127. Accuracy is reported for MMMU, OCRBench, VizWiz, ScienceQA, TextVQA, and OmniBench; lower word error rate is better for LibriSpeech and WenetSpeech. The paired Accuracy–Flip evaluation is restricted to MMMU and ScienceQA-IMG and compares each quantized prediction with the same Dense/FP16 prediction by sample identity; Appendix C.3 gives the complete definition.

C.2 Hyperparameter Analysis

Tab. 8 varies one method-specific hyperparameter at a time while fixing the others to the main configuration. Here, H is the branch horizon, K_c is the number of critical states retained in each round, λ_b and η balance the two risk terms, and γ weights critical-state calibration. The number of on-policy rounds R is studied separately in Tab. 4. All variants use the same calibration data, initialized quantized model, and optimization schedule; only the listed hyperparameter changes.

In the risk-composition ablation, *KL instead of JS* replaces the JS divergence in both the same-state term j_t and the per-step branch divergence used to compute A_t; the remaining risk and optimization settings are fixed. In the policy-refresh ablation, *2 rounds, no refresh* reuses the first round's trajectory bank for the second update, whereas *5 rounds, refresh* recollects the trajectory bank after every model update.

Table 8: One-factor-at-a-time hyperparameter analysis on Qwen3-VL-8B-Instruct under W4A6. Shaded rows denote the main settings; all values are accuracy (↑).

Setting	MMMU	ScienceQA
Branch horizon H		
2	47.9	88.9
4	48.8	89.7
8	48.3	89.2
Critical-state budget K_c		
64	47.4	88.5
128	48.1	89.1
256	48.8	89.7
Branch weight λ_b		
0.5	47.9	88.9
1.0	48.8	89.7
2.0	48.2	89.1
Distribution weight η		
0.5	48.3	89.2
1.0	48.8	89.7
2.0	48.1	89.0
Critical-loss weight γ		
0.1	47.8	88.8
0.2	48.8	89.7
0.4	48.0	89.0

The two datasets exhibit consistent trends. Increasing K_c from 64 to 256 steadily improves accuracy, showing the benefit of covering more critical states within the evaluated budget. The branch horizon peaks at $H = 4$; extending it to eight steps provides no additional gain under this protocol. Both risk coefficients perform best at $\lambda_b = \eta = 1.0$, while $\gamma = 0.2$ gives the best balance between critical-state calibration and the context anchor. Across all sweeps, non-default settings remain within 1.4 points on MMMU and 1.2 points on ScienceQA of the main configuration, indicating moderate sensitivity without performance collapse.

C.3 Accuracy–Flip evaluation

For the closed-choice ScienceQA-IMG and MMMU evaluations, every quantized prediction is paired by sample identity with the same Dense/FP16 prediction. Let $c_i^D, c_i^Q \in \{0, 1\}$ denote their correctness indicators. Flip Rate is $N^{-1} \sum_i \mathbf{1}\{c_i^D \neq c_i^Q\}$, equivalently the sum of correct-to-incorrect and incorrect-to-correct transition rates. In contrast, the accuracy change is their difference, so opposing transitions can cancel. Fig. 4 reports single-run paired estimates on ScienceQA-IMG (N=2,017) and MMMU (N=900), using the same evaluation samples and Dense/FP16 reference for every method.

D Supplementary Analyses

The mismatch–consequence, risk-concentration, and risk-migration analyses in Secs. D.2–D.4 are computed from model trajectories without downstream labels or task scores.

D.1 Motivating diagnostics

Sec. 3 uses Qwen2.5-VL-7B-Instruct with W4A8 QIG and MASQuant checkpoints; their calibration settings follow Appendix C.1. The state-coverage and fork-aligned diagnostics use a 64-example state bank and 120 non-overlapping ScienceQA-IMG examples for evaluation. Rollouts use greedy decoding with at most 64 response tokens. Counterfactual branches follow the current quantized

policy with horizon $H = 4$; if either branch terminates early, statistics are computed over the common valid horizon.

For the anchored on-policy diagnostic in Observation III, the closed-loop JS change relative to full-sequence static calibration is -0.0125 with a paired 95% interval of $[-0.0241, -0.0007]$; the corresponding accuracy change is $+0.005$ with interval $[-0.010, 0.020]$. Across candidate states, local JS and counterfactual branch amplification have Pearson correlation 0.232, Spearman correlation 0.170, and top-20% overlap 21.15%.

D.2 Mismatch–consequence analysis

Fig. 3 uses Qwen2.5-VL-7B-Instruct under W4A8 quantization and 32 source-disjoint held-out prompts, yielding 767 states with valid $H{=}16$ counterfactual branches. A state is a decision mismatch when the dense and quantized top-1 tokens differ, and is high consequence when its $H{=}16$ branch divergence lies in the top quintile. Uncertainty is estimated with 2,000 prompt-cluster bootstrap replicates.

D.3 Risk-concentration analysis

Fig. 3(c) uses the same Qwen2.5-VL-7B-Instruct W4A8 checkpoint and 767 held-out on-policy states as the mismatch–consequence analysis. We rank states by the full Decision–Consequence risk and compute the fraction of its total mass captured by each selection budget. The top 1%, 5%, 10%, and 20% of states capture 13.8%, 29.1%, 43.5%, and 64.2% of the total risk, respectively.

D.4 Risk-migration analysis

Fig. 3(d) compares the states selected on the same 128-prompt calibration manifest before and after the first OnPTQ update. Each round stores two selected states per prompt, yielding 256 states. Exact retention requires both the prompt identity and quantized prefix token sequence to match; timestep retention requires only the prompt identity and generation timestep to match. Both rounds use the same rollout seed schedule. Because the stored rollouts contain only the selected states rather than every token-level risk value, we report migration through selected-state retention rather than changes in the complete risk ranking.

D.5 State-selection value analysis

Setup and controlled variants. Tab. 3 uses Qwen2.5-VL-7B-Instruct under W4A8. All calibrated variants start from the same Dense–InitQ checkpoint and use the same 128 calibration prompts. Each prompt contributes one mismatch and one near-miss candidate, from which each rule selects one state, giving an identical budget of 128 states. State weights are fixed to one, and the optimized parameters, loss terms, optimizer, and update schedule are shared; only the selection rule changes. Dense–InitQ receives no additional update. Mismatch-only selects the mismatch; Mismatch + Near Miss selects the higher-ranked candidate under $e_t + \eta j_t$; Consequence-only uses A_t; OnPTQ uses r_t^{DC}; and Random control samples from the same two-candidate pool.

Held-out evaluation. We freeze a common evaluation bank before comparing checkpoints: 32 source-disjoint prompts yield 767 valid states, including 185 dense–quantized top-1 mismatches and 39 joint-critical states. A joint-critical state is both a top-1 mismatch and in the top quintile of independently measured $H{=}16$ branch consequence. Every checkpoint is evaluated on identical quantized prefixes and identical reference–competitor action pairs. Tab. 3 reports the joint-critical subset; Tab. 9 additionally reports all states and the full mismatch subset.

Metrics and uncertainty. For a frozen state set $\mathcal{S}$, we report

$$\bar{e}_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} [m_D^t - m_Q^t]_+, \qquad \mathrm{Flip}_{\mathcal{S}} = \frac{100}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \mathbf{1}\{m_Q^t < 0\}. \tag{33}$$

The first metric measures how much quantization erodes the dense reference–competitor margin; the second measures how often that erosion reverses the pair ordering. Lower values mean that the

quantized checkpoint more closely preserves the dense decision boundary. Pairwise uncertainty is estimated with a paired prompt-cluster bootstrap, treating prompts rather than individual tokens as independent sampling units.

Table 9: Complete fixed-state recovery results for alternative state-selection rules (lower is better).

Selection	All erosion	All flip (%)	Mismatch erosion	Mismatch flip (%)	Joint-critical erosion	Joint-critical flip (%)
Dense–InitQ	1.608	22.43	2.600	64.86	2.322	69.23
Mismatch-only	1.303	19.95	2.363	50.81	2.446	61.54
Mismatch + Near Miss	**1.032**	17.73	2.248	52.43	2.144	58.97
Consequence-only	1.073	**15.91**	2.003	45.95	1.664	**43.59**
ONPTQ	1.069	19.04	**1.642**	**44.86**	**1.609**	53.85
Random control	1.216	20.99	2.407	55.68	2.199	64.10

This controlled experiment measures fixed-state decision-boundary recovery; downstream task performance is evaluated separately in Sec. 5.

D.6 SINGLE-SIGNAL DIAGNOSTICS

Tab. 10 compares how each state-level signal relates to current mismatch and future consequence. Same-state JS and boundary erosion primarily track current mismatch, whereas branch consequence is more strongly aligned with future consequence. The full ONPTQ risk integrates these complementary signals for state selection.

Table 10: Diagnostics of individual signals and the ONPTQ risk for current mismatch and future consequence.

Signal	Mismatch AUROC ↑	Future Spearman ↑
Hidden reconstruction error	0.470	-0.050
Same-state JS divergence	**0.870**	0.111
Boundary erosion e_t	0.821	0.077
Branch consequence A_t	0.543	**0.504**
ONPTQ risk r_t^{DC}	0.822	0.127

E LIMITATIONS

ONPTQ adds offline calibration cost through dense–quantized comparisons and short greedy rollouts, whose local consequence surrogate may miss longer-horizon or stochastic effects. Evaluation is currently limited to Qwen-based multimodal models.