SPATIALLY ANCHORED QUERIES FOR EFFICIENT VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

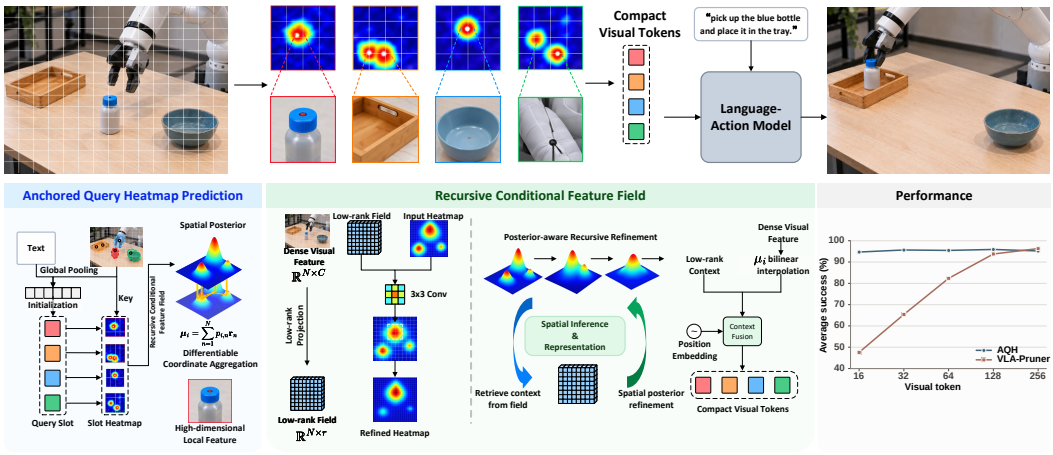


Figure 1: **Anchored Query Heatmaps for compact visual representation in VLA policies.** AQH represents each compact visual slot with an anchored, task-conditioned spatial posterior rather than a single sampling point. Recursive Conditional Feature Field further couples posterior-aware low-rank contextual reasoning with a precise high-dimensional local read, producing compact tokens that retain both spatial coverage and local visual detail. This structured representation remains effective under aggressive visual-token compression, maintaining strong performance at substantially reduced token.

ABSTRACT

Vision-language-action (VLA) models typically process hundreds of visual tokens at every control step, making visual token compression important for efficient robot control. However, aggressive compression is not only about reducing token count, but also about how limited representational capacity is allocated across the scene. Existing pruning methods retain discrete visual tokens, while continuous sampling methods represent each compact token by a single spatial coordinate, potentially discarding broader spatial evidence needed for action. We introduce **Anchored Query Heatmaps (AQH)**, a structured visual bottleneck that represents each compact token with a persistent spatial anchor and a task-conditioned spatial posterior. The posterior guides a precise continuous read from the dense visual feature map while retaining broader spatial context. To capture this context efficiently, we further introduce a **Recursive Conditional Feature Field (RCFF)** that performs posterior-aware aggregation in a shared low-rank spatial field. With only 32 of the original 512 visual tokens, AQH achieves 95.65% average success on LIBERO-2000 and outperforms existing compression methods under the same token budget. A controlled 32-token configuration yields a 1.92× end-to-end policy speedup over the dense reference. AQH also shows consistent gains on LIBERO-Plus and RoboTwin, demonstrating that spatially structured representations enable aggressive visual compression while preserving the evidence needed for action.

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.

1 INTRODUCTION

Vision-language-action (VLA) policies map visual observations and natural-language instructions to robot actions, enabling a single model to perform a broad range of manipulation tasks (Kim et al., 2024; 2025). Modern VLA architectures typically retain a dense sequence of visual patch tokens throughout the downstream language-action model. While this preserves fine-grained visual information, it also makes visual processing a major contributor to inference cost at every control step (Kim et al., 2024; Yang et al., 2025; Li et al., 2025). Reducing the visual-token sequence is therefore an important route toward efficient embodied decision-making.

Visual compression for manipulation, however, is not solely a problem of reducing token count. Robot actions often depend on localized, spatially precise evidence, such as an object boundary, a graspable region, or the spatial relationship between an object and its target. As the visual sequence becomes smaller, each remaining token must represent a larger fraction of the scene. The central question is therefore *how limited representational capacity should be allocated spatially*: a compact policy must determine not only which evidence is relevant, but also how that evidence should be organized and retained within a small number of visual tokens.

Existing compression strategies span discrete token pruning or merging (Bolya et al., 2023; Chen et al., 2024; Yang et al., 2025; Li et al., 2025), learned spatial tokenization (Ryoo et al., 2021), and continuous spatial resampling (Zhu et al., 2021; Feng et al., 2026). These mechanisms reduce downstream computation, but aggressive reduction can discard action-critical geometric evidence before it reaches the policy (Feng et al., 2026). Continuous resampling preserves flexible, potentially off-grid access to the dense feature map, but a coordinate-based read still summarizes each compact slot's high-dimensional access by a point. Conversely, learned spatial aggregation can pool evidence over broader support, but typically collapses that support directly into a token representation (Ryoo et al., 2021). This leaves open how a compact VLA token can retain an explicit, task-conditioned spatial distribution while also preserving a precise local read from the original feature map.

We introduce **Anchored Query Heatmaps (AQH)**, a structured visual bottleneck that represents each compact visual slot with a task-conditioned *spatial posterior* rather than a single sampling coordinate. Each slot is associated with a persistent spatial anchor that provides a stable spatial identity, while visual and language conditioning adapt its posterior toward task-relevant regions. AQH uses these spatial posteriors for both *routing* and *representation*: the posterior guides a precise continuous read from the original high-dimensional feature map while retaining broader spatial context. To capture this context efficiently, we introduce a **Recursive Conditional Feature Field (RCFF)**, which performs posterior-aware aggregation in a shared low-rank spatial field. This design preserves precise local evidence without collapsing the broader spatial distribution into a single sampling coordinate.

AQH requires no additional spatial supervision: no keypoint, segmentation, affordance, coordinate, or posterior annotations are used. The anchors, task-conditioned posteriors, and posterior-aware representations are learned solely through the downstream action objective. We evaluate AQH on LIBERO-2000, based on the LIBERO manipulation benchmark (Liu et al., 2023), and the LIBERO-Plus robustness benchmark (Fei et al., 2026). Compressing the original 512-token visual prefix to only 32 tokens, AQH achieves 95.65% average success on LIBERO-2000, while a controlled 32-token mean-sampling configuration reaches 95.60% with a 1.92× end-to-end policy speedup over the dense reference. On LIBERO-Plus, AQH achieves 65.95% success under the same compact token budget, showing that the compact representation remains effective under robustness perturbations.

Our contributions are:

- We formulate compact VLA representation as a problem of *spatial evidence allocation* and introduce **Anchored Query Heatmaps (AQH)**, which organizes a small set of visual tokens through persistent spatial anchors and task-conditioned spatial posteriors.
- We introduce **Recursive Conditional Feature Field (RCFF)**, a posterior-aware token construction mechanism that recursively conditions anchored spatial posteriors in a shared low-rank feature field and combines posterior-selected context with a single high-dimensional local source read.
- We evaluate AQH across visual-token budgets on LIBERO-2000 and LIBERO-Plus, showing that it supports aggressive visual compression while retaining task performance and reducing end-to-end policy inference cost.

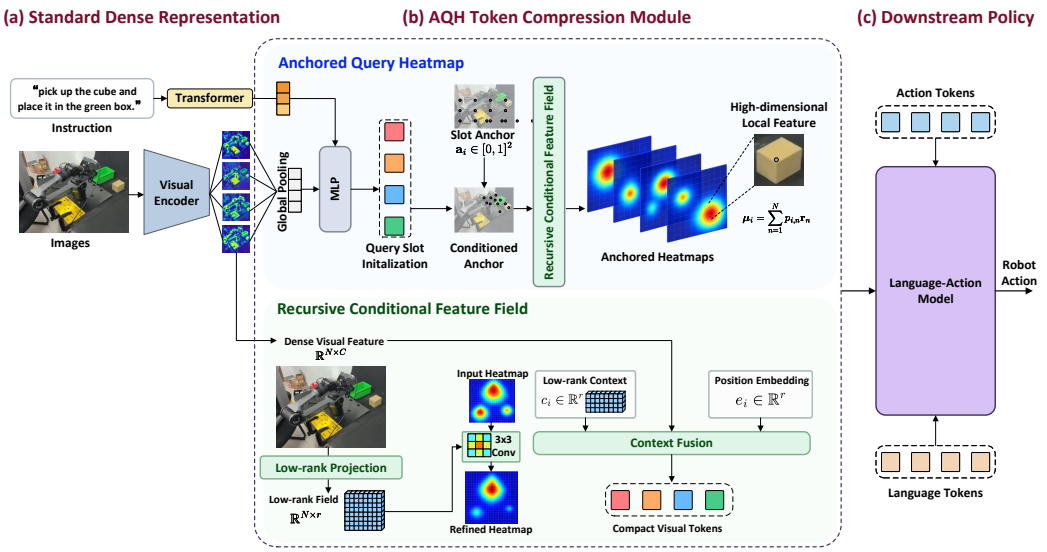


Figure 2: **Architecture of Anchored Query Heatmaps (AQH).** Given dense visual features and a language instruction, AQH initializes task-conditioned query slots and associates each slot with a spatial anchor to predict an anchored posterior over the image. A precise high-dimensional local feature is read from the posterior-selected location, while RCFF performs posterior-aware contextual reasoning in a shared low-rank feature field and recursively refines the spatial posterior. The local feature, low-rank context, and positional information are fused into compact visual tokens, which are then consumed by the downstream language-action model. This asymmetric design combines high-fidelity local evidence with broader spatial context while substantially reducing the visual-token sequence.

2 Related Work

Efficient visual representations for VLA. Vision-language-action models increasingly build on large multimodal backbones for language-conditioned robotic control (Brohan et al., 2022; Zitkovich et al., 2023; Driess et al., 2023; Kim et al., 2024; Black et al., 2024; Li et al., 2024; Bu et al., 2025). Their dense visual streams motivate learned visual bottlenecks (Jaegle et al., 2021; Ryoo et al., 2021) and token reduction methods based on attention, diversity, or calibration (Chen et al., 2024; Zhang et al., 2025; Alvar et al., 2025; Ye et al., 2025). Recent VLA-specific approaches further exploit temporal reuse, model scheduling, and action-aware pruning (Xu et al., 2025; Yang et al., 2025; Li et al., 2025; Wang et al., 2025; Liu et al., 2025). These methods primarily reduce computation by selecting, pruning, or reusing discrete visual tokens. AQH instead learns a fixed-budget, spatially grounded bottleneck that preserves task-relevant spatial information under aggressive compression.

Continuous and adaptive spatial sampling. Adaptive spatial computation has been explored through deformable convolution and attention (Dai et al., 2017; Zhu et al., 2021; Xia et al., 2022), while coordinate-based implicit representations support continuous feature querying beyond fixed image grids (Chen et al., 2021; Mildenhall et al., 2020). Most closely related, GridS (Feng et al., 2026) reformulates VLA token pruning as task-conditioned continuous resampling with differentiable bilinear interpolation. AQH preserves this differentiable continuous readout but lifts the routing representation from a single coordinate to a task-conditioned spatial posterior, using a continuous location for precise local sampling while retaining the full distribution for contextual representation.

3 ANCHORED QUERY HEATMAPS

3.1 OVERVIEW AND DESIGN RATIONALE

Dense visual encoders produce hundreds of tokens, although only a small subset of the image is typically relevant to the current manipulation task. Continuous token sampling provides an effective way to remove this redundancy: rather than selecting discrete patches, a sampler predicts task-dependent continuous coordinates and interpolates features at those locations. Because bilinear interpolation is differentiable with respect to the sampling coordinates, these locations can be optimized directly from the downstream action objective.

While continuous sampling avoids the quantization error of discrete token selection, it still represents each compact visual slot by a single two-dimensional coordinate. This creates a different information bottleneck. A point specifies *where* to read a feature, but cannot express uncertainty, multiple plausible source regions, or how visual evidence is distributed around the selected location.

We introduce **Anchored Query Heatmaps (AQH)** (Fig. 2) to retain this missing spatial structure while preserving differentiable continuous sampling. Instead of directly regressing one coordinate per slot, AQH predicts a normalized task-conditioned heatmap over the source feature grid, which we interpret as a spatial posterior. Each slot starts from an anchor that provides an initial spatial reference, while visual and language conditioning reshape the heatmap toward task-relevant evidence. The heatmap is used in two complementary ways: it induces a continuous coordinate for a high-dimensional local feature read, and it is retained as a distribution for posterior-aware contextual reasoning.

Directly aggregating the original high-dimensional feature map under every heatmap would, however, offset much of the computational benefit of token compression. We therefore introduce a **Recursive Conditional Feature Field (RCFF)**, which performs posterior-wide reasoning in a shared low-rank spatial field. Each output token consequently combines one high-dimensional continuous source read with broader context selected by the full spatial posterior.

The following sections describe how AQH constructs anchored heatmaps, converts them into differentiable continuous reads, and retains the remaining posterior information through RCFF. Implementation details are provided in Appendix B.

3.2 ANCHORED HEATMAPS AND DIFFERENTIABLE SPATIAL ROUTING

For each camera view, let

$$\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}, \qquad \mathbf{x}_n \in \mathbb{R}^C, \tag{1}$$

denote dense visual features arranged on an $H \times W$ grid, where $N = HW$. Each feature is associated with a normalized image-plane coordinate $\mathbf{r}_n \in [0,1]^2$. AQH compresses this representation into K visual slots.

Anchored heatmap prediction. Each slot i is associated with an anchor $\mathbf{a}_i \in [0,1]^2$, which provides an initial spatial reference rather than a fixed region assignment. Visual and language conditioning adapt this reference to the current task, producing $\tilde{\mathbf{a}}_i$. The conditioned slot representation forms a query $\mathbf{q}_i$, while dense visual features are projected into keys $\mathbf{k}_n$.

We define the heatmap logits and normalized spatial posterior as

$$\ell_{i,n} = \frac{\mathbf{q}_i^\top \mathbf{k}_n}{\tau} - \beta \left\| \tilde{\mathbf{a}}_i - \mathbf{r}_n \right\|_2^2, \qquad \mathbf{p}_i = \mathrm{softmax}(\boldsymbol{\ell}_i), \tag{2}$$

where τ controls content-matching sharpness and β controls the strength of the spatial preference.

The two terms play complementary roles. The adapted anchor provides a soft starting point for spatial search, while query–key compatibility allows probability mass to move toward task-relevant visual evidence. Thus, the anchor organizes the search without constraining the slot to a fixed image region.

From heatmaps to continuous feature reads. A key requirement is to convert the spatial posterior into an actual high-dimensional feature without reverting to discrete patch selection. We first compute

its spatial expectation

$$\boldsymbol{\mu}_i = \sum_{n=1}^{N} p_{i,n} \mathbf{r}_n, \tag{3}$$

which defines a continuous coordinate in the image plane even though $\mathbf{p}_i$ is represented on a discrete feature grid.

We denote by $\mathcal{I}_{\mathbf{X}}(\boldsymbol{\mu}_i)$ the feature obtained by bilinearly interpolating the dense feature grid at this coordinate. Because Eq. 3 is differentiable with respect to the heatmap probabilities and bilinear interpolation is differentiable with respect to the sampling coordinate, the downstream action loss provides a direct learning signal to the heatmap predictor. AQH therefore preserves the end-to-end differentiable continuous routing property of coordinate-based sampling while learning a richer spatial representation.

The posterior mean is useful for smooth continuous routing, but collapsing a multimodal posterior to its mean can place the read between distinct high-probability regions. AQH therefore transitions toward a mode-oriented straight-through readout during optimization, while retaining the full posterior for representation. We defer the precise posterior-to-coordinate routing procedure to Appendix B.3.

This distinction is central to AQH: the coordinate is used only when a precise high-dimensional source read is required, whereas the full heatmap remains available for reasoning about spatially distributed evidence.

3.3 Posterior-Aware Context through RCFF

The full heatmap contains information that cannot be represented by its sampling coordinate alone. A straightforward approach would be to aggregate the original dense features according to the posterior, $\sum_n p_{i,n}\mathbf{x}_n$. However, doing so for every compact slot requires posterior-wide aggregation at the full feature width C and mixes high-dimensional features across potentially broad spatial regions.

RCFF instead separates precise feature acquisition from posterior-wide contextual reasoning. The dense feature map is projected into a shared low-rank spatial field

$$\mathbf{F}_t \in \mathbb{R}^{N \times r}, \qquad r \ll C, \tag{4}$$

where N is the number of spatial locations and r is the field dimension. Lightweight spatial processing propagates contextual information across the field, and $\mathbf{F}_t$ denotes the field used at refinement stage t.

Let $\mathbf{h}_i^{(t)}$ denote the state of slot i at stage t. Its compatibility with the spatial field is computed as

$$\mathbf{g}_i^{(t)} = \frac{\mathbf{F}_t W_q \mathbf{h}_i^{(t)}}{\sqrt{r}}, \qquad \boldsymbol{\pi}_i^{(t)} = \mathrm{softmax}\left(\log \mathbf{p}_i + \mathbf{g}_i^{(t)}\right), \tag{5}$$

where $\mathbf{p}_i \in \mathbb{R}^N$ is the anchored AQH posterior and $W_q \mathbf{h}_i^{(t)} \in \mathbb{R}^r$ projects the slot state into the field dimension. The refined posterior $\boldsymbol{\pi}_i^{(t)} \in \mathbb{R}^N$ therefore reweights the original AQH posterior using contextual evidence from the field.

The refined posterior then retrieves a low-rank contextual feature,

$$\mathbf{z}_i^{(t)} = \mathbf{F}_t^\top \boldsymbol{\pi}_i^{(t)} \in \mathbb{R}^r, \tag{6}$$

which is used to update the slot state before the next refinement stage. This creates a recurrent interaction between spatial inference and representation: the current posterior selects contextual evidence, while the resulting slot state further refines which spatial regions are relevant. Because each refinement is conditioned on the original AQH posterior, RCFF enriches rather than replaces the spatial routing learned by the anchored heatmap.

Crucially, all posterior-wide aggregation occurs at width r instead of the original visual width C. The low-rank field is therefore used to preserve *spatial coverage*: it allows the full heatmap to gather distributed contextual evidence without repeatedly accessing full-dimensional features. RCFF does not replace the original dense representation. Instead, it complements a single high-dimensional source read that preserves fine local visual evidence. This separation between low-rank contextual reasoning and full-width local feature acquisition is central to the final token construction.

3.4 Local Precision and Posterior Context

AQH combines two complementary representations in each compact token. The RCFF branch captures broad posterior-selected context efficiently in the low-rank field, while a single continuous read from the original dense feature map preserves high-dimensional detail at a precise location. This design separates spatial coverage from feature fidelity rather than forcing both roles into either a single coordinate or a full-dimensional posterior aggregation.

Let $\boldsymbol{\pi}_i$ denote the final refined posterior and $\hat{\mathbf{c}}_i$ the continuous read coordinate derived from it. The local branch reads the original high-dimensional feature map only once,

$$\mathbf{x}_i^{\text{local}} = \mathcal{I}_{\mathbf{X}}\left(\hat{\mathbf{c}}_i\right). \tag{7}$$

Let $\mathbf{z}_i$ denote the posterior-selected contextual representation produced by the final RCFF refinement. The compact output token is

$$\mathbf{y}_i = \mathcal{I}_{\mathbf{X}}\left(\hat{\mathbf{c}}_i\right) + \alpha W_{\text{out}}\mathbf{z}_i + E_{\text{pos}}\left(\hat{\mathbf{c}}_i\right). \tag{8}$$

The first term provides full-width visual evidence at the selected continuous location, while the second transfers broader posterior-selected context from the low-rank field into the same compact token. The two branches therefore play complementary roles: the local read preserves precise appearance information that benefits from the original feature width, whereas RCFF retains spatially distributed evidence that would be lost if the heatmap were reduced to its sampling coordinate alone. Since sparse sampling removes the implicit geometry of the original regular grid, the positional term explicitly records where the local feature was acquired.

4 Experiments

4.1 Experimental setup

We evaluate AQH from three complementary perspectives: accuracy–efficiency trade-offs on LIBERO-2000, robustness on LIBERO-Plus, and transfer to a different policy and observation setup on RoboTwin. We additionally analyze the learned token allocation and RCFF refinement process, and provide a qualitative closed-loop rollout on a physical robot.

For LIBERO-2000, we follow the four standard suites—Spatial, Object, Goal, and Long—with 500 episodes per suite, for a total of 2,000 episodes.

LIBERO-Plus evaluates robustness under a substantially larger common manifest of 10,030 cases. Because its four suites contain different numbers of episodes, the reported Overall score is computed over all episodes rather than by averaging the four suite-level percentages. In this experiment, AQH without RCFF, the Feature-Field variant, and the complete AQH model are evaluated on the same manifest.

Finally, RoboTwin evaluates a different setting based on X-VLA with three camera inputs. Each of the three tasks contains 100 episodes. All efficiency numbers in Table 3 are measured using the same isolated H20 profile and the same 10-step action-generation setting, making the FLOPs and latency comparisons within this table directly comparable.

4.2 Accuracy–efficiency trade-off on LIBERO-2000

Table 1 shows that the main benefit of AQH emerges as the visual-token budget becomes restrictive. At 256 tokens, VLA-Pruner achieves the highest average success rate, while AQH is slightly below the dense reference. This changes in the more compressed regime: AQH reaches 95.90%, 95.45%, and 95.65% average success at 128, 64, and 32 tokens, respectively, all while retaining only a small fraction of the original visual sequence. In particular, the 32-token setting uses only $1/16$ of the dense token budget, yet slightly exceeds the 512-token reference in average success.

The important observation is not the small difference with respect to Dense, but the *flatness* of the AQH performance curve. From 256 down to 32 tokens, its average success remains within a narrow range despite a $8\times$ reduction in token count. In contrast, several competing compression strategies become noticeably more sensitive as the budget is reduced. VLA-Pruner performs particularly well

Table 1: LIBERO-2000 success rates (%) and efficiency at different visual-token budgets. All methods use OpenVLA-OFT. Acc. is normalized to the evaluated vanilla OpenVLA-OFT baseline (95.38% success = 100%). Best and second-best results within each token budget are shown in **bold** and underline, respectively.

Method	Tokens	Spatial	Object	Goal	Long	Average	Acc. (%)↑	FLOPs (T)↓	Latency (ms)↓
512-token budget									
OpenVLA-OFT	512	**95.8**	**98.7**	96.3	**90.7**	**95.38**	**100.00**	3.903	135.78
VLA-Cache (50% reuse)	512†	95.4	96.0	**96.7**	90.2	94.58	99.16	2.730	101.01
VLA-Cache (75% reuse)	512†	85.6	85.3	84.1	80.3	83.83	87.89	1.938	85.02
VLA-Cache (87.5% reuse)	512†	57.3	58.7	64.8	26.2	51.75	54.26	**1.373**	**76.15**
256-token budget									
FastV	256	94.6	96.8	92.7	87.0	92.78	97.27	2.219	88.23
SparseVLM	256	94.1	93.7	91.2	85.1	91.03	95.44	2.289	90.22
DivPrune	256	90.8	91.1	89.9	83.1	88.73	93.03	**2.126**	**88.01**
VLA-Pruner	256	**97.3**	98.6	96.8	92.6	**96.33**	**101.00**	2.234	92.86
GridS	256	93.00	98.40	**98.20**	**93.40**	95.75	100.39	2.186	99.62
AQH + RCFF	256	92.80	**98.80**	96.00	93.20	95.20	99.81	2.186	126.22‡
128-token budget									
FastV	128	87.8	81.8	87.6	74.6	82.95	86.97	1.401	72.73
SparseVLM	128	89.8	84.8	82.7	79.1	84.10	88.17	1.459	74.71
DivPrune	128	83.9	72.6	79.3	72.5	77.08	80.81	1.389	**72.30**
VLA-Pruner	128	93.5	96.2	95.2	90.2	93.78	98.32	1.420	75.64
GridS	128	92.00	97.60	**96.20**	92.60	94.60	99.18	**1.339**	75.55
AQH + RCFF	128	**95.40**	**99.60**	95.20	**93.40**	**95.90**	**100.55**	**1.339**	88.08‡
64-token budget									
FastV	64	60.6	59.2	60.6	28.6	52.25	54.78	1.082	**66.11**
SparseVLM	64	73.1	65.0	67.5	32.8	59.60	62.49	1.136	67.65
DivPrune	64	61.4	61.8	64.9	30.8	54.73	57.38	1.059	66.46
VLA-Pruner	64	88.1	87.6	84.9	68.8	82.35	86.34	1.096	68.95
GridS	64	92.60	**98.40**	**95.40**	92.20	94.65	99.23	**0.918**	72.15
AQH + RCFF	64	**96.60**	98.20	94.60	**92.40**	**95.45**	**100.07**	**0.918**	86.24‡
32-token budget									
GridS	32	91.60	96.40	96.00	87.00	92.75	97.24	**0.709**	83.92*
AQH + RCFF	32	**96.40**	**97.80**	**97.40**	**91.00**	**95.65**	**100.28**	**0.709**	**71.72**‡
16-token budget									
GridS	16	92.60	96.80	93.20	90.40	93.25	97.77	**0.548**	**80.47**‡
AQH + RCFF	16	**93.20**	**99.00**	**95.60**	**91.00**	**94.70**	**99.29**	**0.548**	97.83‡

Table 2: LIBERO-Plus success rates (%) on the same 10,030-case manifest. The overall rate is weighted by episode count.

Method	Tokens	Spatial	Object	Goal	Long	Overall	Retention
OpenVLA-OFT	512	74.35	77.44	58.86	67.96	69.52	1.000
GridS	32	69.11	72.99	53.38	56.05	62.74	0.902
AQH w/o RCFF	32	72.19	74.34	54.03	58.48	64.60	0.929
AQH	32	73.69	74.94	57.31	58.48	65.95	0.949

at 256 tokens, but degrades substantially at 64 tokens; GridS is more stable, but its performance also drops at 32 tokens. These results suggest that, once the representation becomes highly compressed, *where and how visual information is aggregated* matters more than simply preserving a larger number of isolated tokens.

This effect is clearest at 32 tokens. AQH improves over GridS on all four LIBERO suites, with the largest differences on Spatial and Long. Thus, the overall improvement is not produced by a single favorable suite. At the same time, the results should not be interpreted as showing that fewer tokens are intrinsically better: the AQH curve is non-monotonic across independently trained budgets, and its 16-token configuration falls below the 32-token configuration. Instead, the experiments indicate that AQH tolerates severe compression well over a relatively broad operating range, while an extremely small representation eventually becomes a meaningful bottleneck.

The compute trend is correspondingly strong. AQH requires 0.709 T FLOPs at 32 tokens, compared with 3.903 T for the dense policy, an approximately 82% reduction. At 16 tokens, FLOPs decrease further to 0.548 T. Wall-clock latency does not decrease perfectly monotonically with token count, however. For example, the 16-token AQH configuration is slower than the 32-token one despite

Table 3: RoboTwin task success and inference metadata. Each task contains 100 episodes. FLOPs and latency are from an isolated X-VLA profile on one H20 GPU with the same three-camera input shape and 10-step action generation.

Method	Tokens	Token ratio	Bread in skillet	Handover mic	Stack bowls	Average	Queries/episode	FLOPs (T)	Latency (ms/query)
Dense	50	1.00×	68.00%	94.00%	90.00%	84.00%	14.82	2.105	198.74
GridS	16	0.32×	56.00%	96.00%	88.00%	80.00%	15.15	1.476	132.99
Top-K	16	0.32×	61.00%	94.00%	88.00%	81.00%	14.97	1.476	132.56
AQH + RCFF	16	0.32×	67.00%	94.00%	95.00%	85.33%	14.62	1.476	135.77

using fewer FLOPs. This discrepancy highlights the increasing role of fixed kernel, memory, and implementation overhead at very short sequence lengths; FLOPs therefore provide the cleaner view of algorithmic scaling, whereas latency reflects the complete execution stack.

4.3 Robustness on LIBERO-Plus

Table 2 evaluates whether the compact representation remains useful under the more challenging LIBERO-Plus evaluation. Dense OpenVLA-OFT reaches 69.52% overall success, while GridS at 32 tokens drops to 62.74%. AQH without RCFF improves this to 64.60%, and the complete AQH model reaches 65.95%, corresponding to 94.9% retention of the dense policy's overall success rate.

Two aspects of this result are informative. First, the advantage of AQH over uniform grid sampling persists outside the original LIBERO-2000 evaluation, suggesting that its benefit is not limited to reproducing a particular benchmark trajectory distribution. Second, the gap to the dense policy remains visible on LIBERO-Plus. Thus, the compressed representation retains a large fraction of dense performance but does not remove the cost of severe compression under stronger distribution shift.

4.4 Transfer to RoboTwin

We further test whether the same compression principle remains effective outside the OpenVLA-OFT/LIBERO setting. Table 3 evaluates X-VLA on three RoboTwin tasks using a three-camera observation. The dense model uses 50 tokens, whereas GridS, Top-K, and AQH use 16 tokens, or 0.32× the dense token count.

AQH achieves the highest average success among the evaluated methods at 85.33%, compared with 84.00% for the dense model. Importantly, this average does not reflect uniform gains on every task: AQH nearly matches Dense on Bread in skillet and Handover mic, while its advantage is concentrated on Stack bowls. The result is therefore better interpreted as evidence that the compressed representation can preserve task performance in this setting, rather than as a general claim that compression improves control accuracy.

The efficiency measurements are more direct because all RoboTwin variants are profiled under the same hardware and inference protocol. Reducing the token count from 50 to 16 lowers FLOPs from 2.105 T to 1.476 T and reduces per-query latency from 198.74 ms to 135.77 ms for AQH. Hence, in this three-camera setting, the representation reduction translates into both lower model computation and a measurable end-to-end inference benefit. Together with the LIBERO results, this experiment suggests that AQH is not tied to a single policy backbone or a particular two-view observation format, although the limited three-task evaluation does not by itself establish broad cross-domain generalization.

4.5 What does AQH preserve?

The qualitative comparisons in Fig. 5 help explain the different behavior of the compression strategies. GridS and AQH without RCFF can be viewed as selecting a small collection of discrete sampling locations, while VLA-Pruner retains a sparse set of image patches. AQH instead associates each of its 16 read tokens per view with a spatial posterior. The high-dimensional reads are concentrated around task-relevant regions, but their posterior support extends beyond a single point or patch.

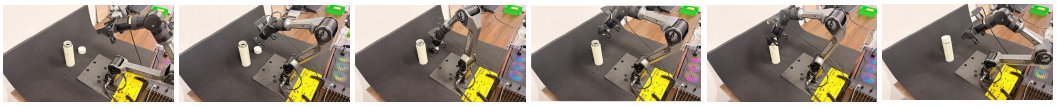


Figure 3: **Representative real-robot rollout with AQH.** Frames are shown chronologically from left to right. The policy operates with AQH-compressed visual observations throughout closed-loop execution, demonstrating that the compact visual representation transfers from benchmark evaluation to physical robot manipulation.

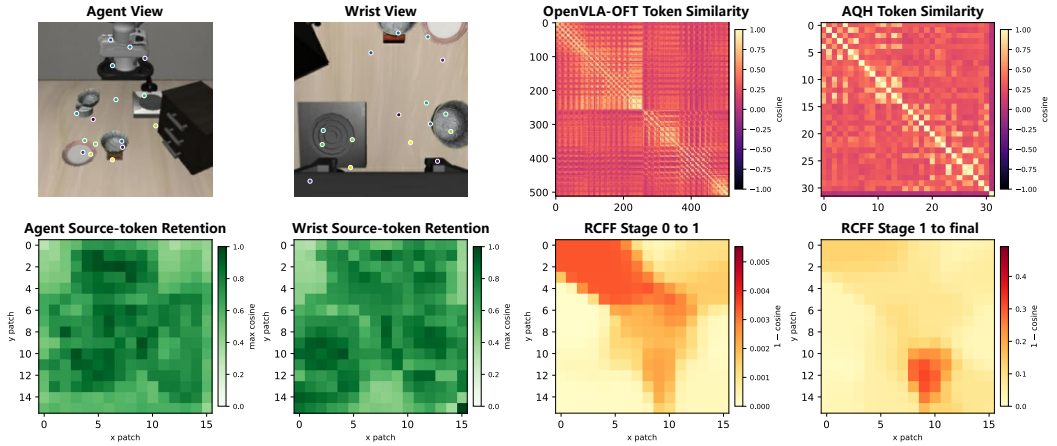


Figure 4: **Representation analysis of AQH and RCFF.**

Consequently, a small number of tokens can preserve evidence from a broader spatial neighborhood without reverting to the original dense sequence.

Figure 4 provides a complementary view in feature space. The pairwise similarity matrix of AQH tokens contains less repeated structure than the much larger dense token set, indicating that the compressed sequence is not simply composed of near-duplicate visual descriptors. At the same time, the source-token retention maps show that dense source features from both camera views remain represented by the compact token set over broad image regions. These observations are consistent with AQH trading dense, redundant sampling for a smaller set of spatially informative representations.

4.6 Physical-robot rollout

Finally, Fig. 3 shows a representative closed-loop rollout of AQH on a physical robot. The policy receives AQH-compressed observations throughout the trajectory and completes the demonstrated manipulation sequence without switching back to a dense visual representation. This example confirms that the compact representation can be executed in a real perception–action loop rather than only in offline benchmark evaluation.

5 Conclusion

We presented AQH, a compact visual representation for vision-language-action policies that replaces dense visual tokens with a small set of spatially informative reads and selectively refines them with RCFF. Across LIBERO-2000, LIBERO-Plus, and RoboTwin, AQH preserves strong task performance under aggressive token budgets while substantially reducing computation, with its advantage most evident in the highly compressed regime. Qualitative analyses further suggest that AQH retains broad task-relevant spatial support rather than isolated sampling locations, while RCFF performs increasingly localized refinement.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, 2025. doi: 10.1109/CVPR52734.2025.00877. 3

Kevin Black et al. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024. URL https://doi.org/10.48550/arXiv.2410.24164. 3

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu. 2

Anthony Brohan et al. RT-1: Robotics transformer for real-world control at scale. *CoRR*, abs/2212.06817, 2022. URL https://doi.org/10.48550/arXiv.2212.06817. 3

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. *CoRR*, abs/2505.06111, 2025. doi: 10.48550/arXiv.2505.06111. 3

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision (ECCV)*, pp. 19–35, 2024. doi: 10.1007/978-3-031-73004-7_2. 2, 3

Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8628–8638, 2021. doi: 10.1109/CVPR46437.2021.00852. 3

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, 2017. doi: 10.1109/ICCV.2017.89. 3

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, pp. 8469–8488, 2023. 3

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026. 2

Yixu Feng, Zinan Zhao, Yanxiang Ma, Chenghao Xia, Chengbin Du, Yunke Wang, and Chang Xu. See what matters: Differentiable grid sample pruning for generalizable vision-language-action model. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.11817. 2, 3

Andrew Jaegle, Felix Gimeno, Andrew Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, pp. 4651–4664, 2021. 3

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.017. 2

Moo Jin Kim et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, pp. 2679–2713, 2024. 2, 3

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *CoRR*, abs/2411.19650, 2024. doi: 10.48550/arXiv.2411.19650. 3

Ye Li, Yuan Meng, Zewen Sun, Kangye Ji, Chen Tang, Jiajun Fan, Xinzhu Ma, Shutao Xia, Zhi Wang, and Wenwu Zhu. SP-VLA: A joint model scheduling and token pruning approach for vla model acceleration. *CoRR*, abs/2506.12723, 2025. doi: 10.48550/arXiv.2506.12723. 2, 3

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 44776–44791, 2023. doi: 10.52202/075280-1939. 2

Ziyan Liu, Yeqiu Chen, Hongyi Cai, Tao Lin, Shuo Yang, Zheng Liu, and Bo Zhao. Bridging the semantic-action gap in visual token pruning for efficient vla inference. *CoRR*, abs/2511.16449, 2025. doi: 10.48550/arXiv.2511.16449. 3

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, pp. 405–421, 2020. doi: 10.1007/978-3-030-58452-8_24. 3

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *CoRR*, abs/2106.11297, 2021. 2, 3

Hanzhen Wang, Jiaming Xu, Jiayi Pan, Yongkang Zhou, and Guohao Dai. SpecPrune-VLA: Accelerating vision-language-action models via action-aware self-speculative pruning. *CoRR*, abs/2509.05614, 2025. doi: 10.48550/arXiv.2509.05614. 3

Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4784–4793, 2022. doi: 10.1109/CVPR52688.2022.00475. 3

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. VLA-Cache: Efficient vision-language-action manipulation via adaptive token caching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 3

Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. EfficientVLA: Training-free acceleration and compression for vision-language-action models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 45758–45781, 2025. doi: 10.52202/085713-1365. 2, 3

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 22128–22136, 2025. 3

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning (ICML)*, 2025. 3

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 3

Brianna Zitkovich et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, pp. 2165–2183, 2023. 3

Table 4: LIBERO-2000 ablation study of AQH+RCFF at a 32-token budget.

Variant	Coordinate	Field	Spatial	Object	Goal	Long	Average	Acc. (%)
AQH bilinear	AQH expectation	No	91.80	96.60	96.80	93.20	94.6	99.18
Coordinate-only	RCFF coordinate	No	92.40	97.00	97.00	93.00	94.85	99.44
Field-only	AQH coordinate	Yes	94.00	99.00	96.40	91.00	95.10	99.71
AQH + RCFF	RCFF refinement	Yes	96.40	97.80	97.40	91.00	95.65	100.28

Coordinate-only and field-only were trained but could not be evaluated because the required sampler checkpoint was missing. Their success rates are therefore left blank.

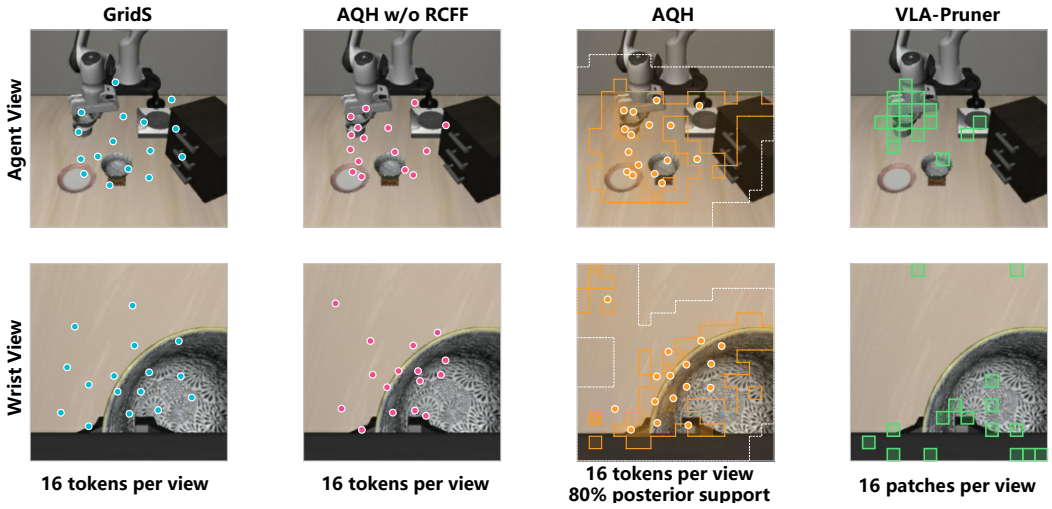


Figure 5: **Qualitative comparison of visual-token allocation across compression strategies.**

A Reproducibility and Claim Ledger

B Additional Details of Anchored Query Heatmaps

AI Statement: We use AI for language polishing and editing.

This section provides additional architectural and optimization details of Anchored Query Heatmaps (AQH). The main text focuses on the motivation and core design of AQH: representing each compact visual slot with a task-conditioned spatial posterior, refining posterior-aware context in a shared low-rank field, and combining this context with a single high-dimensional source read. Here, we describe the construction of these components in greater detail.

B.1 Anchor and Query Construction

For each camera view, AQH receives a dense visual representation

$$\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}, \qquad \mathbf{x}_n \in \mathbb{R}^C,$$

arranged on an $H \times W$ spatial grid, where $N = HW$. Each source feature is associated with a normalized coordinate $\mathbf{r}_n \in [0,1]^2$.

AQH maintains K compact slots per camera view. Each slot i is associated with an anchor

$$\mathbf{a}_i \in [0,1]^2,$$

which provides an initial spatial reference in the image plane. The anchors are distributed across the image plane to give the compact slots diverse starting locations. They do not define hard spatial partitions and do not constrain a slot to remain at a fixed coordinate.

For each slot, an encoding of $\mathbf{a}_i$ is combined with a learned slot embedding. The resulting representation is conditioned on pooled visual features and the language instruction to form a task-conditioned

query $\mathbf{q}_i$. A lightweight offset predictor additionally adapts the initial anchor to the current observation and instruction, yielding

$$\tilde{\mathbf{a}}_i \in [0,1]^2.$$

Thus, $\mathbf{a}_i$ determines only the initial spatial reference, while $\tilde{\mathbf{a}}_i$ provides an observation- and task-dependent reference used during posterior construction.

The dense visual features are independently projected into low-dimensional keys $\{\mathbf{k}_n\}_{n=1}^N$. As described in the main text, AQH combines query–key compatibility with a soft spatial preference:

$$\ell_{i,n} = \frac{\mathbf{q}_i^\top \mathbf{k}_n}{\tau} - \beta \left\|\tilde{\mathbf{a}}_i - \mathbf{r}_n\right\|_2^2, \qquad \mathbf{p}_i = \mathrm{softmax}\left(\boldsymbol{\ell}_i\right). \tag{9}$$

The spatial coefficient β corresponds to the Gaussian-style spatial preference used in our implementation. Equivalently, if its scale and strength are parameterized separately as σ_a and λ, then

$$\beta = \frac{\lambda}{2\sigma_a^2}. \tag{10}$$

This compact parameterization is used in the main text to emphasize the role of the spatial term rather than its implementation-specific decomposition.

B.2 Low-Rank Conditional Feature Field

Direct posterior-weighted aggregation over the original C-dimensional feature map would require every compact slot to repeatedly access full-width features over all spatial locations. RCFF avoids this by constructing a shared low-rank spatial field.

The dense visual representation is first projected into

$$\mathbf{F}_0 \in \mathbb{R}^{H\times W\times r}, \qquad r \ll C. \tag{11}$$

Lightweight spatial propagation then produces a short sequence of contextual fields,

$$\mathbf{F}_0,\ \mathbf{F}_1,\ \mathbf{F}_2. \tag{12}$$

The reported AQH configuration uses two recursive refinement stages.

For notational convenience, we flatten the spatial dimensions and write $\mathbf{F}_t \in \mathbb{R}^{N\times r}$. At refinement stage t, the current slot state $\mathbf{h}_i^{(t)}$ is matched against the corresponding field:

$$\mathbf{g}_i^{(t)} = \frac{\mathbf{F}_t W_q \mathbf{h}_i^{(t)}}{\sqrt{r}}, \qquad \boldsymbol{\pi}_i^{(t)} = \mathrm{softmax}\left(\log \mathbf{p}_i + \mathbf{g}_i^{(t)}\right). \tag{13}$$

The logarithm is applied element-wise.

Equation 13 is equivalent to multiplicatively reweighting the base AQH posterior with the contextual compatibility predicted from the field:

$$\pi_{i,n}^{(t)} \propto p_{i,n} \exp\left(g_{i,n}^{(t)}\right).$$

Writing the refinement in logit space makes explicit that RCFF preserves the base spatial posterior while adding contextual evidence, rather than predicting a separate unconstrained attention distribution.

The refined posterior retrieves contextual information from the field as

$$\mathbf{z}_i^{(t)} = \mathbf{F}_t^\top \boldsymbol{\pi}_i^{(t)}. \tag{14}$$

The resulting contextual feature is used to update the slot state before the next refinement stage. Consequently, posterior refinement and representation are coupled recursively: the posterior selects contextual evidence, and the resulting contextualized slot representation influences the next refinement.

All posterior-wide aggregation in RCFF operates at feature width r, rather than at the original visual width C.

B.3 Posterior-to-Coordinate Routing

Although AQH retains the full spatial posterior for representation, a single coordinate is eventually required to access the original high-dimensional feature map.

Given a posterior $\boldsymbol{\pi}_i$, its spatial expectation is

$$\boldsymbol{\mu}_i = \sum_{n=1}^{N} \pi_{i,n} \mathbf{r}_n. \tag{15}$$

The posterior mean provides a differentiable continuous spatial reference and is particularly useful during early optimization.

During training, AQH gradually transitions from expectation-based routing toward a straight-through MAP reference. Let

$$n_i^\star = \arg\max_n \pi_{i,n}, \qquad \mathbf{c}_i^{\mathrm{MAP}} = \mathbf{r}_{n_i^\star}. \tag{16}$$

The expectation provides a smooth spatial learning signal, whereas the MAP-oriented reference encourages the model to commit to a high-probability posterior mode.

A zero-initialized residual correction is further used when constructing the final read coordinate. We denote the resulting coordinate by

$$\hat{\mathbf{c}}_i \in [0,1]^2.$$

The zero initialization keeps the source read close to the posterior-derived reference at the beginning of training while allowing the routing coordinate to adapt during optimization.

The precise transition schedule and residual parameterization are training-time implementation details; neither changes the spatial posterior used by the RCFF contextual branch.

B.4 Continuous Read from the Dense Feature Map

The main text denotes the high-dimensional local feature at coordinate $\mathbf{c}$ by $\mathcal{I}_{\mathbf{X}}(\mathbf{c})$. Here we make this operator explicit.

For a continuous coordinate $\mathbf{c}$, let $\mathcal{N}(\mathbf{c})$ denote the four neighboring locations on the discrete feature grid. Bilinear interpolation gives

$$\mathcal{I}_{\mathbf{X}}(\mathbf{c}) = \sum_{n \in \mathcal{N}(\mathbf{c})} w_n(\mathbf{c}) \mathbf{x}_n, \qquad \sum_{n \in \mathcal{N}(\mathbf{c})} w_n(\mathbf{c}) = 1, \tag{17}$$

where $w_n(\mathbf{c})$ are the standard bilinear interpolation weights.

Only one such read from the original C-dimensional feature map is performed for each compact slot:

$$\mathbf{x}_i^{\mathrm{local}} = \mathcal{I}_{\mathbf{X}}\left(\hat{\mathbf{c}}_i\right). \tag{18}$$

This operation preserves high-dimensional visual information at a precise posterior-selected location while avoiding full-width aggregation over all source locations.

B.5 Posterior-Aware Token Construction

Let $\boldsymbol{\pi}_i$ denote the posterior after the final RCFF refinement and $\mathbf{z}_i$ the corresponding low-rank contextual representation. The contextual branch is projected back to the visual-token dimension:

$$\mathbf{x}_i^{\mathrm{ctx}} = W_{\mathrm{out}} \mathbf{z}_i = W_{\mathrm{out}} \mathbf{F}^\top \boldsymbol{\pi}_i. \tag{19}$$

The final AQH token combines the high-dimensional local read, the posterior-selected contextual feature, and a spatial encoding:

$$\mathbf{y}_i = \mathcal{I}_{\mathbf{X}}\left(\hat{\mathbf{c}}_i\right) + \alpha W_{\mathrm{out}} \mathbf{z}_i + E_{\mathrm{pos}}\left(\hat{\mathbf{c}}_i\right). \tag{20}$$

Here, α controls the contribution of the posterior-context branch.

The first term retains full-width evidence at one precise source location. The second retains broader evidence distributed according to the refined posterior, but performs this aggregation only in the

rank-r feature space. The positional term preserves the spatial reference associated with the compact visual token.

This asymmetric construction is deliberate: local visual fidelity is preserved through a single high-dimensional read, whereas spatially distributed reasoning is delegated to the substantially cheaper low-rank field.

B.6 Learning Objective and Supervision

AQH is optimized jointly with the downstream vision-language-action policy. No auxiliary objective is introduced:

$$\mathcal{L} = \mathcal{L}_{\text{action}}. \tag{21}$$

In particular, AQH does not use keypoint annotations, segmentation masks, affordance labels, target coordinates, or spatial-posterior supervision. The task-dependent references, spatial posteriors, RCFF refinement, and posterior-aware token representations are all learned through the downstream control objective.

The anchors should therefore not be interpreted as supervised spatial targets. Their role is to provide initial spatial references from which the compact slots can develop task-dependent spatial allocation.

B.7 Computational Structure

For each camera view, AQH maintains K spatial posteriors over $N = HW$ source locations. Query–key matching is performed in a low-dimensional space, and posterior-wide contextual aggregation is performed in the shared rank-r field with $r \ll C$.

Each compact slot performs only one access to the original C-dimensional feature map. Thus, the dominant representation paths can be summarized as

$$\underbrace{\text{posterior-wide reasoning at width } r}_{\text{shared contextual path}} \quad + \quad \underbrace{\text{one source read at width } C}_{\text{local high-dimensional path}}.$$

With G camera views and K compact slots per view, the visual prefix passed to the downstream language-action model contains

$$B = GK \tag{22}$$

tokens.

In our main configuration, two camera views each contribute $K = 16$ compact tokens, yielding $B = 32$ visual tokens in total, compared with the original 512-token visual prefix. Camera views are processed independently by AQH before their compact visual tokens are concatenated with the remaining non-visual tokens for downstream policy inference.