CORE-Pruning: Preference-Conditioned Structured Pruning for Hallucination-Aware Vision-Language Models

Anonymous ACL submission

Abstract

Structured pruning is attractive for deploying vision-language models (VLMs), but equal sparsity does not imply equal behavior: a pruned VLM can retain task accuracy while suffering a large drop in object-hallucination robustness. This makes layer allocation a central design variable, not a bookkeeping detail. We present **CORE-Pruning**, a preference-conditioned framework for choosing layer-wise structured pruning ratios under a global sparsity budget. CORE represents an entire pruning plan with a low-dimensional latent code decoded by a transformer, allowing correlated layer decisions to be generated jointly instead of sampling global and per-layer budgets independently. It further combines three search signals: (i) an *Adaptive Saliency Module* (ASM) that updates layer importance with plan-conditioned graph message passing; (ii) a *Stability Surrogate Regularizer* (SSR) that discourages unstable plans with a spectral-radius penalty; and (iii) a *Recovery-Aware Meta-Objective* (RAMO) that uses a lightweight proxy trained from LoRA recovery runs to favor plans that fine-tune well. We optimize the policy with plan-level Group Relative Policy Optimization (GRPO) using a return over POPE-measured hallucination robustness, ScienceQA utility, and compression. On LLaVA-1.5-7B and Qwen2.5-VL-3B, CORE-Pruning improves over the direct preference-conditioned baseline by up to $+$**5.1 POPE points** and $+$**2.7 SQA points** at matched sparsity and matched recovery budget. A single trained policy also supports zero-shot preference queries across the measured hallucination–utility–compression trade-off simplex.

1 Introduction

Large vision-language models (VLMs) underpin open-ended multimodal assistants(Liu et al., 2023), yet their scale makes deployment costly. Model pruning is an attractive remedy, but for VLMs *compression alone is not enough*. At the same overall sparsity, pruned models can exhibit markedly different behaviors: conventional task-accuracy may remain nearly intact while hallucination-oriented evaluations degrade sharply (Figure 1). This decoupling is particularly concerning for VLM assistants, where object hallucination—confidently describing visual entities absent from the image—is measurable and systematically benchmarked via POPE(Li et al., 2023). These observations motivate treating hallucination robustness as an explicit optimization objective rather than a post-hoc diagnostic, complementing inference-time mitigation work(Leng et al., 2024). Throughout the paper, "robustness" refers to this measured object-hallucination axis; we do not claim comprehensive coverage of all multimodal safety failures.

The above issue reflects a broader challenge in *multi-objective pruning*(Roijers et al., 2013). Robustness, utility, and compression often conflict, and the desirable trade-off depends on deployment constraints and risk tolerance. Pruning decisions are also highly non-uniform across layers(Han et al., 2015; Frankle and Carbin, 2018), and post-training pruning methods for LLMs show that data-dependent criteria can preserve accuracy even at high sparsity(Sun et al., 2023; Frantar and Alistarh, 2023). Together, these results suggest that the central question is not only *how much* to prune, but *where* to allocate sparsity.

Inspired by resource allocation in complex systems, we view VLM pruning as a *conditional resource allocation* problem: under a fixed compression budget, different preferences over robustness and utility should induce different layer-wise sparsity plans. Unlike text-only pruning, VLM pruning must protect components critical to visual grounding. Static visual sensitivity (e.g., precomputed attention mass (Abnar and Zuidema, 2020)) becomes inaccurate as pruning alters attention patterns. We therefore introduce an **Adaptive Saliency Module (ASM)**: a graph-convolutional module that dynam-

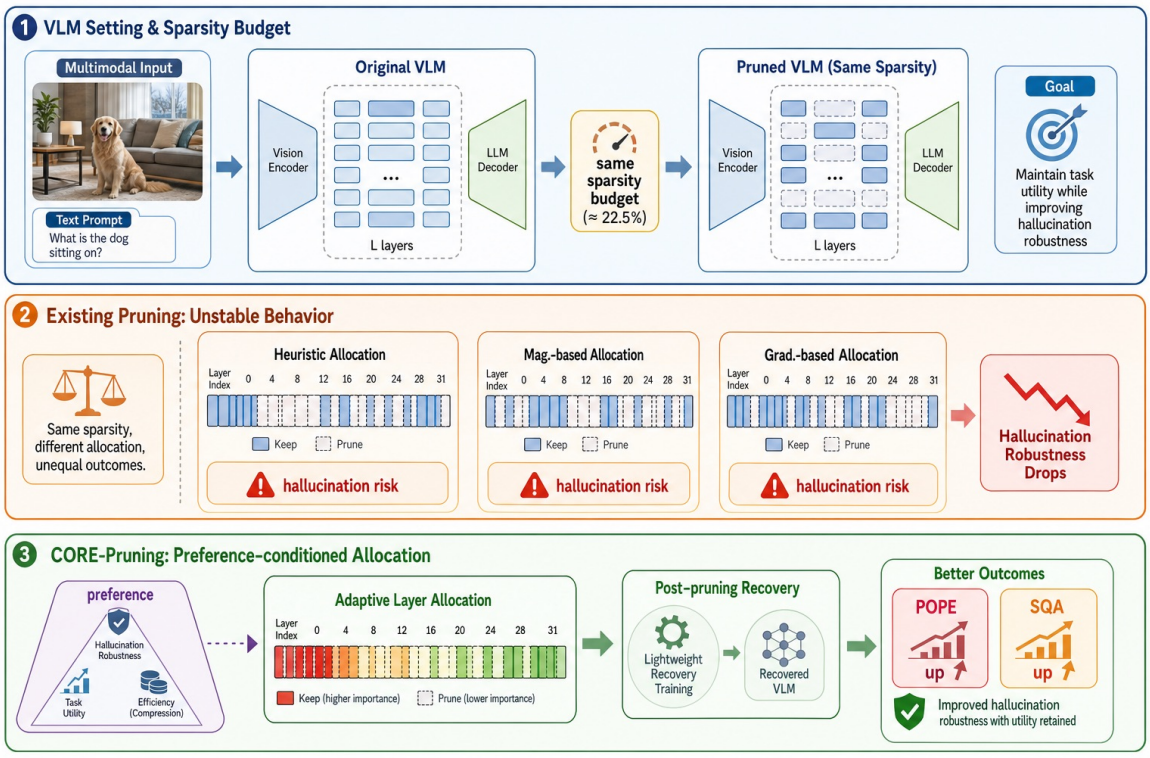

Figure 1: **Equal sparsity, unequal object-hallucination behavior.** At a fixed sparsity budget, different layer-wise allocations can yield different POPE and ScienceQA outcomes. CORE-Pruning searches for preference-conditioned allocations and accounts for post-pruning recovery during plan selection.

ically re-weights each layer's importance conditioned on the current pruning plan, ensuring that vision-critical components are protected precisely when robustness is prioritized.

Preference-conditioned frameworks such as HiPP-Prune(Bai et al., 2026) already show that a single policy can produce pruning plans for varying trade-off vectors, avoiding a separate search for each fixed scalarization. Nevertheless, HiPP-Prune and its predecessors share three structural limitations: (i) they factor layer ratios as independent Beta and Dirichlet samples, ignoring cross-layer dependencies; (ii) their sensitivity estimates are static and stale; and (iii) they rely on episodic stability gating that discards entire episodes and produces a discontinuous optimization landscape.

We propose **CORE-Pruning**, which makes three changes to this setup:

- **Hierarchical Variational Allocation (HVA).** A low-dimensional latent code, decoded by a differentiable transformer, generates the complete layer-wise plan in one shot, capturing cross-layer dependencies and enforcing global budget compliance without independent factorization.

- **Adaptive Saliency Module (ASM) and Stability Surrogate Regularizer (SSR).** ASM dynamically re-estimates layer importance via graph message passing conditioned on current pruning ratios. SSR replaces episodic gating with a smooth spectral-radius penalty, discouraging unstable high-sparsity plans without discarding episodes.

- **Recovery-Aware Meta-Optimization (RAMO).** A lightweight neural proxy predicts post-pruning fine-tuning performance and augments the policy reward, biasing search toward plans that respond well to a fixed LoRA recovery budget.

CORE-Pruning is optimized with plan-level GRPO under a multi-objective return combining ScienceQA utility, POPE object-hallucination robustness, and a convex compression reward. Experiments on LLaVA-1.5-7B and Qwen2.5-VL-3B show consistent gains over static and preference-conditioned baselines at matched sparsity and matched post-pruning recovery budgets.

2 Related Work

Efficiency and compression for vision-language models. Two bottlenecks shape VLM efficiency: the quadratic cost of attention over multimodal contexts, including KV-cache pressure during serving (Kwon et al., 2023), and the parameter footprint of large language backbones. Recent work pursues *token-level sparsification*, reducing visual tokens or sparsifying the vision–language context to accelerate attention and KV-cache usage(Huang et al., 2024a; Ye et al., 2025b; Huang et al., 2024b; Alvar et al., 2025; Ye et al., 2025a; Zhuang et al., 2025; Sun et al., 2025; Huang et al., 2025; Jiang et al., 2025; Tang et al., 2025; Apedo et al., 2026; Takezoe et al., 2026). A complementary line targets *structural* compression, removing redundant layers or modules for hardware-friendly speedups(Ma et al., 2025; He et al., 2024). Learning-based VLM pruning further generalizes pruning decisions across prompts and tasks(Liang et al., 2025). These methods mainly reduce visual-token computation, study modality-level sparsity/restoration, or search fixed structural policies. Our focus is *preference-controllable, layer-wise structured weight allocation*: one policy must choose a full sparsity plan for different robustness–utility–compression preferences.

Post-training pruning for large language models. Data-aware methods such as Wanda(Sun et al., 2023) and SparseGPT(Frantar and Alistarh, 2023) show that accurate one-shot updates preserve performance at high sparsity, while structured LLM pruning (e.g., LLM-Pruner(Ma et al., 2023)) shows the practical value of neuron-level sparsity. Signal-propagation analyses (SynFlow(Tanaka et al., 2020), SNIP(Lee et al., 2018), GraSP(Wang et al., 2020)) emphasize that preserving gradient flow is critical for avoiding degenerate subnetworks. CORE-Pruning differs from these static criteria by penalizing unstable plans with a spectral-radius proxy and by adapting layer importance as the cross-modal pathway changes under pruning.

Post-pruning recovery. Parameter-efficient adaptation (LoRA(Hu et al., 2022)) provides a practical recovery mechanism under a limited trainable-parameter budget. Prior VLM studies discuss pruning with restoration strategies(He et al., 2024), and LLM work has examined recovery protocols under constrained budgets(Seale Smith et al., 2025). CORE-Pruning moves recovery into plan search via RAMO: a proxy trained on LoRA recovery outcomes is incorporated into the policy reward, so the search can prefer plans that are easier to repair under a fixed recovery budget.

3 Method

3.1 Problem Formulation

Let f_{θ_0} be a pretrained VLM (e.g., LLaVA(Liu et al., 2023)) whose language backbone contains L structured-prunable linear projections. A *pruning plan* is a vector $\mathbf{r} = [r_1, \ldots, r_L]^\top \in [0,1]^L$, where r_ℓ is the fraction of rows removed from layer ℓ. Applying plan $\mathbf{r}$ yields a pruned model $\theta(\mathbf{r}) = \mathcal{P}(\theta_0, \mathbf{r})$. We evaluate each plan with a three-dimensional objective vector

$$\mathbf{J}(\mathbf{r}) = \left[J_{\text{rob}}(\mathbf{r}),\ J_{\text{util}}(\mathbf{r}),\ J_{\text{comp}}(\mathbf{r})\right]^\top, \quad (1)$$

measuring hallucination robustness (POPE balanced accuracy), task utility (ScienceQA accuracy), and compression benefit, respectively.

Preference-conditioned policy. A preference vector $\mathbf{w} \in \Delta^2$ ($w_{\text{rob}} + w_{\text{util}} + w_{\text{comp}} = 1, \mathbf{w} \geq \mathbf{0}$) encodes deployment-specific trade-offs, following the standard preference-vector view of multi-objective decision making (Roijers et al., 2013). We learn a policy $\pi_\phi(\mathbf{r} \mid \mathbf{s}, \mathbf{w})$ that outputs a *complete* layer-wise plan in one shot, where $\mathbf{s}$ is a layer-wise state representation comprising activation statistics and visual sensitivity signals. The policy is optimized to maximize the scalarized objective

$$J_{\mathbf{w}}(\mathbf{r}) = \mathbf{w}^\top \operatorname{norm}(\mathbf{J}(\mathbf{r}))\,, \quad (2)$$

with $\operatorname{norm}(\cdot)$ denoting quantile-based online normalization (Section 3.5).

Key departure from HiPP-Prune. HiPP-Prune samples a global sparsity $s \sim \text{Beta}(\alpha_s, \beta_s)$ and a per-layer allocation $\mathbf{p} \sim \text{Dirichlet}(\boldsymbol{\alpha})$ *independently*, then forms $\mathbf{r} = s \cdot \mathbf{p}$. This independent factorization cannot capture structural correlations across layers—e.g., an attention head and its subsequent MLP projection often require correlated budgets to remain coherent. CORE-Pruning instead treats $\mathbf{r}$ as a deterministic function of a low-dimensional latent code:

$$\mathbf{r} = \mathcal{T}_\psi(\mathbf{z}), \qquad \mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{s}_{\text{ASM}}, \mathbf{w}), \quad (3)$$

where $\mathbf{z} \in \mathbb{R}^M$ ($M \ll L$) is sampled from a learned variational posterior and $\mathcal{T}_\psi$ is a differentiable transformer-based decoder. Cross-layer dependencies are captured through the shared latent $\mathbf{z}$; global budget compliance is enforced analytically inside $\mathcal{T}_\psi$.

3.2 CORE-Pruning Overview

CORE-Pruning comprises three modules, each targeting one limitation of HiPP-Prune:

1. **Hierarchical Variational Allocation (HVA)** models global sparsity and cross-layer dependencies through a reparameterized joint latent distribution decoded by a transformer (Section 3.4).

2. **Adaptive Saliency Module (ASM)** dynamically estimates layer-wise importance conditioned on the current plan via graph message passing (Section 3.3).

3. **Stability Surrogate Regularizer (SSR)** replaces episodic gating with a differentiable spectral-radius penalty, complemented by the **Recovery-Aware Meta-Objective (RAMO)**, which internalizes post-pruning fine-tuning into policy search (Section 3.6–3.7).

The full pipeline is illustrated in Figure 2. One-shot generation means that a single policy sample produces the full layer-wise plan; ASM is then evaluated on that sampled plan to build the reward-side state used by QuANT, SSR, and RAMO before the GRPO update.

3.3 Adaptive Saliency Module (ASM)

Static sensitivity scores—such as attention mass over visual tokens(Abnar and Zuidema, 2020)—become inaccurate as pruning alters the network's attention landscape. We address this with a lightweight graph convolutional network (GCN) that computes a *plan-conditional* saliency $\tilde{\mathcal{S}}_\ell(\mathbf{r})$ for each layer:

$$\begin{aligned} \tilde{\mathcal{S}}_\ell(\mathbf{r}) &= \sigma\Big(\mathrm{MLP}\Big(\mathbf{h}_\ell \,\|\, \mathbf{m}_\ell\Big)\Big)\,, \\ \mathbf{m}_\ell &= \mathrm{Agg}_{j\in\mathcal{N}(\ell)}\big(\mathbf{h}_j \odot \mathrm{sigmoid}(\mathbf{W}_a\mathbf{r})\big). \end{aligned} \tag{4}$$

where $\mathbf{h}_\ell = [\|\mathbf{W}_\ell\|_1,\, \mathrm{RMS}(\mathbf{x}_\ell),\, \mathcal{S}_\ell^{(\text{static})}]^\top$ are per-layer static descriptors, $\mathcal{N}(\ell)$ is the neighborhood of layer ℓ in the computational graph (same-stage attention and MLP blocks), $\odot$ is element-wise multiplication, σ is the logistic sigmoid, and Agg is mean-pooling. The gating term $\mathrm{sigmoid}(\mathbf{W}_a\mathbf{r})$ modulates each neighbor's contribution by its current sparsity ratio, so the saliency estimate adapts as the plan changes.

The ASM is pre-trained for 10K gradient steps on 20 randomly sampled pruning simulations using a meta-objective that predicts post-pruning accuracy drop, then frozen during policy optimization to prevent instability. The resulting dynamic saliency $\tilde{\mathbf{s}}(\mathbf{r}) = [\tilde{\mathcal{S}}_1(\mathbf{r}), \ldots, \tilde{\mathcal{S}}_L(\mathbf{r})]^\top$ is concatenated with static features: $\mathbf{s}_{\text{ASM}} = [\mathbf{s}_{\text{static}};\, \tilde{\mathbf{s}}(\mathbf{r})]$.

3.4 Hierarchical Variational Allocation (HVA)

The variational encoder $q_\phi(\mathbf{z} \mid \mathbf{s}_{\text{ASM}}, \mathbf{w})$ outputs a diagonal Gaussian posterior:

$$\mathbf{z} \sim \mathcal{N}\big(\boldsymbol{\mu}_\phi(\mathbf{s}_{\text{ASM}}, \mathbf{w}),\ \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{s}_{\text{ASM}}, \mathbf{w}))\big). \tag{5}$$

The transformer decoder $\mathcal{T}_\psi$ maps $\mathbf{z}$ to a budget-compliant plan:

$$\begin{aligned} \tilde{\mathbf{r}} &= \mathrm{softplus}\big(\mathrm{MLP}_\psi(\mathbf{z})\big), \\ \mathbf{r} &= \mathrm{proj}_{[0,1]^L}\left(\frac{\tilde{\mathbf{r}}}{\max(\|\tilde{\mathbf{r}}\|_1,\ \epsilon)} \cdot C_{\text{target}}\right), \end{aligned} \tag{6}$$

where $C_{\text{target}} = c_{\min} + \mathrm{sigmoid}(v)(c_{\max} - c_{\min})$ with a learnable scalar v that adapts the global sparsity target. The projection $\mathrm{proj}_{[0,1]^L}$ clips each entry then rescales to meet the budget exactly, enforcing a hard global constraint. The latent space is regularized via a KL penalty against a standard Gaussian prior:

$$\mathcal{L}_{\text{KL}} = \mathrm{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{s}_{\text{ASM}}, \mathbf{w}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big), \tag{7}$$

which encourages a compact latent representation and prevents posterior collapse. By routing through a latent of fixed dimension $M = 16$ (independent of L), CORE-Pruning reduces the effective action space from $\mathcal{O}(L)$ independent continuous variables to $\mathcal{O}(M)$, reducing the number of continuous choices explored by the policy.

3.5 Multi-Objective Returns: QuANT Normalization and Convex Compression

Quantile-based Adaptive Normalization (QuANT). Running-mean normalization, as used in HiPP-Prune, suffers from non-stationarity when the policy distribution shifts rapidly. We instead normalize each objective within the current group of G plans using robust quantile statistics. For raw returns $\{J^{(g)}\}_{g=1}^G$, define

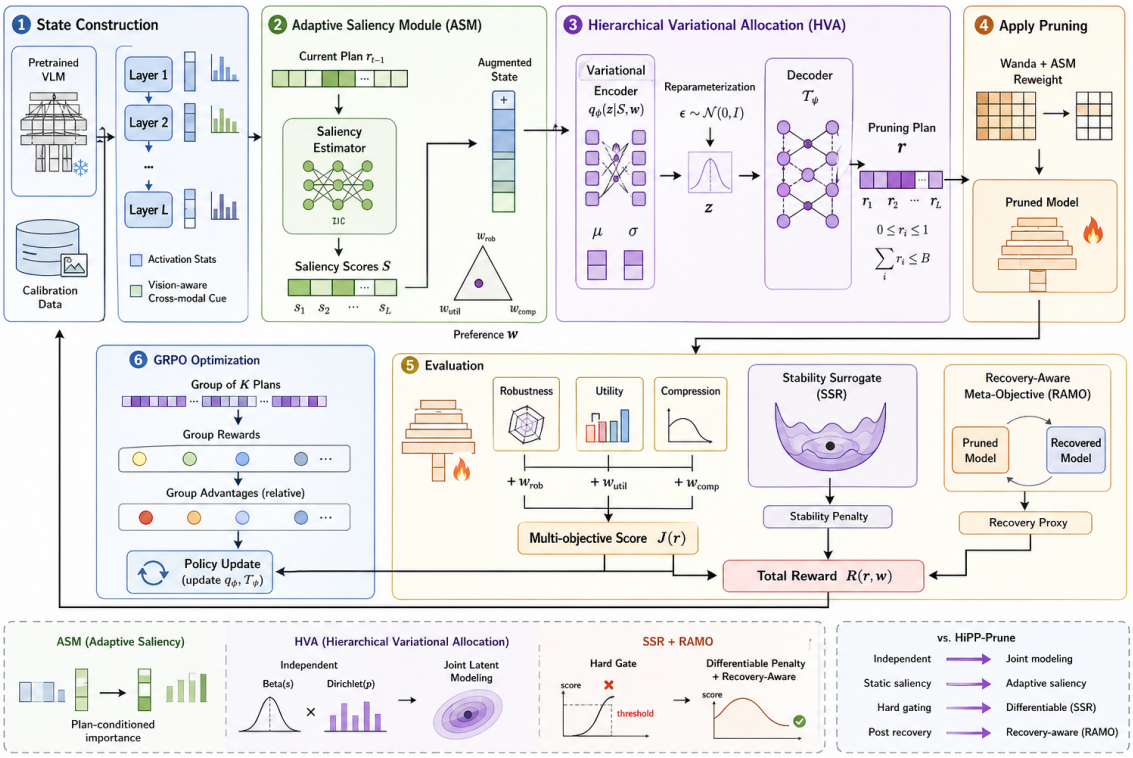

Figure 2: **CORE-Pruning training workflow.** *(a) Policy encoder (HVA):* a variational encoder maps the augmented state $\mathbf{s}_{\text{ASM}}$ and preference vector $\mathbf{w}$ to a diagonal-Gaussian posterior over a M-dimensional latent code $\mathbf{z}$; the transformer decoder $\mathcal{T}_\psi$ maps $\mathbf{z}$ to a budget-compliant pruning plan $\mathbf{r}$. *(b) Adaptive Saliency Module (ASM):* a graph-convolutional network re-estimates layer importance for the sampled plan. *(c) Reward computation:* the pruned model is evaluated on POPE and ScienceQA; rewards are assembled from QuANT-normalized objectives, augmented by RAMO's recovery proxy and penalized by SSR's spectral-radius term. *(d) GRPO update:* group-relative advantages propagate gradients through the reparameterized sample $\mathbf{z}$, jointly updating ϕ and ψ.

group statistics $m_G = \text{median}(\{J^{(g)}\}_{g=1}^G)$ and $q_G = \text{IQR}(\{J^{(g)}\}_{g=1}^G)$:

$$\hat{J}^{(g)} = \text{clip}_{[-3,3]}\left(\frac{J^{(g)} - m_G}{q_G + \epsilon}\right). \tag{8}$$

Clipping at ± 3 limits the influence of extreme plans and provides an outlier-resistant normalization computed within each group.

Convex compression reward.

$$J_{\text{comp}}(\mathbf{r}) = 1 - \exp\left(-\alpha \cdot \frac{\bar{r} - c_{\min}}{c_{\max} - c_{\min}}\right), \quad \bar{r} = \frac{1}{L}\sum_\ell r_\ell, \tag{9}$$

with $\alpha = 2.5$. This function is continuously differentiable, strictly increasing in $\bar{r}$, and concave, so it rewards additional sparsity while reducing the marginal incentive for extreme pruning.

Scalarized reward.

$$R(\mathbf{r};\mathbf{w}) = w_{\text{rob}}\,\hat{J}_{\text{rob}} + w_{\text{util}}\,\hat{J}_{\text{util}} + w_{\text{comp}}\,J_{\text{comp}}(\mathbf{r}). \tag{10}$$

3.6 Stability Surrogate Regularizer (SSR)

HiPP-Prune uses a multiplicative episodic gate that discards entire episodes violating a SynFlow stability threshold. Hard gating wastes informative (if fragile) plans and introduces a discontinuous optimization landscape that inflates gradient variance. We replace it with a *differentiable surrogate* based on the spectral radius of the pruned network's input–output Jacobian.

For the pruned model $f_{\theta(\mathbf{r})}$, let $\mathcal{J}_\theta$ denote the Jacobian of the logit output with respect to the input embedding, averaged over a calibration mini-batch. We estimate the spectral radius $\rho_{\text{sp}}(\mathbf{r}) = \max|\lambda(\mathcal{J}_\theta)|$ via one power-iteration step, used as a low-cost search signal rather than a complete stability certificate. The stability penalty is

$$\mathcal{L}_{\text{stab}}(\mathbf{r}) = \lambda_{\text{stab}} \cdot \max\big(0,\ \rho_{\text{sp}}(\mathbf{r}) - \rho_{\text{thr}}\big), \tag{11}$$

where ρ_{thr} is calibrated on the unpruned model (empirically $\approx$0.95). Unlike the multiplicative gate, this penalty is additive and subdifferentiable, dis-

couraging dynamically unstable plans without discarding entire episodes. The combined actor loss is

$$\begin{aligned}\mathcal{L}_{\text{actor}} &= -\mathbb{E}_{\mathbf{w},\,\mathbf{z}\sim q_\phi}\left[\ell_{\text{pg}}\right] + \beta_{\text{KL}}\,\mathcal{L}_{\text{KL}} \\ &\quad + \mathbb{E}_{\mathbf{w},\mathbf{r}}[\mathcal{L}_{\text{stab}}(\mathbf{r})]\,, \\ \ell_{\text{pg}} &= \frac{1}{G}\sum_{g=1}^{G}\log q_\phi\big(\mathbf{z}^{(g)} \mid \mathbf{s}_{\text{ASM}}, \mathbf{w}\big)A^{(g)}.\end{aligned} \tag{12}$$

where $A^{(g)} = R^{(g)} - \frac{1}{G}\sum_{g'} R^{(g')}$ is the group-relative advantage. Both the KL term and the stability penalty are back-propagated through the reparameterized sample $\mathbf{z}$, enabling end-to-end training.

3.7 Recovery-Aware Meta-Optimization (RAMO)

Standard post-pruning recovery is executed *after* plan selection and therefore cannot guide the search. RAMO bridges this gap by integrating a differentiable surrogate for lightweight fine-tuning into the reward signal.

We train a neural proxy $\mathcal{M}_\xi(\mathbf{r})$ offline on 200 randomly sampled plans with actual 10-epoch LoRA recovery (rank-16, applied to the final four language backbone layers) on a mixed POPE+ScienceQA validation set. $\mathcal{M}_\xi$ learns to predict post-recovery performance from the plan alone, making the recovery signal differentiable and available during policy optimization. The scalarized reward (Eq. 10) is augmented as

$$\begin{aligned}R_{\text{total}}(\mathbf{r};\mathbf{w}) &= R(\mathbf{r};\mathbf{w}) + \gamma_{\text{rec}}\cdot\mathcal{M}_\xi(\mathbf{r}), \\ \gamma_{\text{rec}} &= 0.3.\end{aligned} \tag{13}$$

$\mathcal{M}_\xi$ is held fixed after its initial training. This biases the policy toward plans that both score well before recovery and are predicted to respond well to lightweight recovery, without adding inference-time modules.

3.8 Structured Pruning Operator

Given a plan $\mathbf{r}$, the pruning operator $\mathcal{P}$ applies row-wise removal to each linear layer. We adopt the Wanda weight–activation criterion(Sun et al., 2023), multiplied by the ASM saliency to re-weight pruning pressure:

$$\text{score}_{\ell,i}^{(\text{CORE})} = \|\mathbf{W}_{\ell,i:}\|_1 \cdot \text{RMS}(\mathbf{x}_\ell)\cdot\tilde{\mathcal{S}}_\ell(\mathbf{r}). \tag{14}$$

Layers with higher adaptive saliency receive lower effective pruning pressure even if their raw Wanda score is moderate, thereby protecting vision-critical pathways. Contiguous row removal yields hardware-friendly, directly materializable masks for inference.

3.9 Training Procedure

CORE-Pruning is trained in three stages: ASM pre-training, recovery proxy training, and preference-conditioned policy optimization. The last stage uses plan-level GRPO, a group-relative variant of PPO-style policy optimization(Schulman et al., 2017; Shao et al., 2024; Mroueh et al., 2025). Full training details are provided in Appendix A.

4 Experiments

4.1 Experimental Setup

Backbones. We evaluate on **LLaVA-1.5-7B**(Liu et al., 2023) and **Qwen2.5-VL-3B**(Bai et al., 2025), two VLMs differing in scale, architecture, and multimodal integration strategy. Pruning targets the language backbone's MLP projections via structured row-wise reduction. All data-aware methods use Wanda-style activation statistics(Sun et al., 2023) estimated on a matched calibration set.

Benchmarks. We assess two critical post-pruning behaviors: *object-hallucination robustness* via POPE balanced accuracy (BalAcc)(Li et al., 2023) using the adversarial, popular, and random splits; and *general task utility* via ScienceQA (SQA) accuracy(Lu et al., 2022).

Evaluation protocol. All methods are compared under matched structured sparsity budgets ($\approx$22.5% and $\approx$32.5%). For CORE-Pruning and HiPP-Prune, operating points are obtained by querying the preference-conditioned policy with four fixed vectors $\mathbf{w}$ spanning the Pareto simplex. All methods undergo an identical post-pruning recovery phase (800 steps of LoRA with rank 16 applied to the final four language backbone layers) while the sparsity mask is kept fixed. Reported deviations are not interchangeable: heuristic baselines vary calibration seeds, whereas CORE-Pruning and HiPP-Prune vary queried preference vectors under a fixed trained policy.

Baselines. We compare against: (i) **Random** layer allocation; (ii) **Wanda**(Sun et al., 2023); (iii) **SliceGPT**(Ashkboos et al., 2024); (iv) **LLM-Pruner**(Ma et al., 2023); and (v) **HiPP-Prune**(Bai et al., 2026), our direct predecessor.

4.2 Main Results

Thus the learned-method standard deviations measure preference-query sensitivity, not training-seed variance.

Overall picture. Figure 3 shows three patterns. First, heuristic baselines cluster together at low performance: the best non-learned baseline (SliceGPT) leaves a POPE gap of more than 20 points to the dense model on LLaVA. Second, preference-conditioned policies (HiPP-Prune, CORE-Pruning) sit in a higher accuracy band, indicating that *where* sparsity is allocated dominates performance once *how much* is fixed. Third, CORE-Pruning is the only method whose POPE bar exceeds 76% on LLaVA and 66% on Qwen while also leading the SQA axis; exact LLaVA values are reported in Appendix C.

Geometry of the trade-off. Figure 4 visualizes results as a two-dimensional scatter in the (SQA, POPE) plane. Two structural properties are evident. (i) Heuristic baselines are confined to the bottom-left corner in mean performance, where neither robustness nor utility exceeds 60% / 38%; they are simultaneously Pareto-dominated by both learned methods. (ii) The step from HiPP-Prune to CORE-Pruning is not a trade-off along a single axis: CORE-Pruning achieves simultaneous gains of $+3.6$ POPE and $+1.8$ SQA on LLaVA (and $+4.6$ / $+2.2$ on Qwen). No baseline mean lies within the red-shaded dominance region, giving a direct visual check of the numerical comparison while leaving seed-level uncertainty to future evaluation.

Qwen and high-sparsity regime. Table 1 summarizes Qwen2.5-VL-3B under two compression budgets. The POPE advantage over HiPP-Prune widens from $+4.6$ to $+5.1$ points when sparsity increases from $\approx 22.5\%$ to $\approx 32.5\%$, and the SQA advantage widens from $+2.2$ to $+2.7$ points; Appendix B provides the corresponding slope plot.

4.3 Ablation Studies

Appendix D provides detailed ablations. In brief, fixed preference queries move the expected operating point toward POPE, ScienceQA, or compression without retraining, and dense preference sampling forms a smooth operating cloud under the target sparsity budget. Component ablations show that removing HVA, ASM, SSR, or RAMO hurts at least one reported metric; HVA gives the largest accuracy drop when removed, and removing SSR gives the largest increase in preference-query variance.

5 Conclusion

We presented **CORE-Pruning**, a structured pruning framework for hallucination-aware VLM compression that treats plan generation as preference-conditioned hierarchical variational inference. By replacing the independent Beta+Dirichlet factorization of HiPP-Prune with a low-dimensional latent encoder–decoder (HVA), augmenting the policy state with plan-conditional graph-convolutional saliency (ASM), replacing episodic gating with a differentiable spectral-radius penalty (SSR), and internalizing post-pruning recovery into the reward signal (RAMO), CORE-Pruning improves the measured POPE–ScienceQA trade-off under matched sparsity and recovery budgets.

Experiments on LLaVA-1.5-7B and Qwen2.5-VL-3B show that a single CORE-Pruning agent supports zero-shot preference querying over the measured POPE–SQA–compression simplex, outperforming HiPP-Prune by up to $+5.1$ POPE points and $+2.7$ SQA points. Component ablations show positive contributions from all four modules under the tested setting, and SSR reduces preference-query variance relative to episodic gating.

Limitations

CORE-Pruning is evaluated on LLaVA-1.5-7B and Qwen2.5-VL-3B with POPE and ScienceQA as the primary objectives. This scope isolates the interaction between object hallucination and task utility, while leaving larger backbones, additional multimodal benchmarks (e.g., MMBench, HallusionBench, and CHAIR), and measured deployment latency as natural extensions. We also focus on language-backbone pruning; extending the same preference-conditioned formulation to vision encoders and cross-modal projection layers is a promising direction.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4190–4197.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. Divprune: Diversity-based visual token pruning for large multimodal mod-

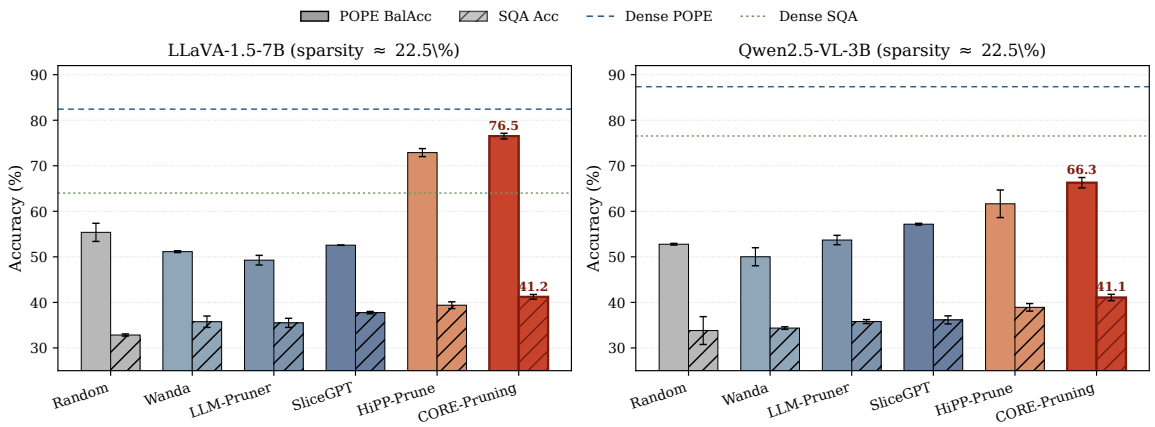

Figure 3: **POPE and ScienceQA results at $\approx 22.5\%$ structured sparsity.** Solid bars denote POPE BalAcc, hatched bars denote SQA Acc, and dashed lines mark dense-model ceilings. The plotted means for CORE-Pruning exceed HiPP-Prune by $+3.6/+1.8$ points on LLaVA and $+4.6/+2.2$ on Qwen (POPE/SQA).

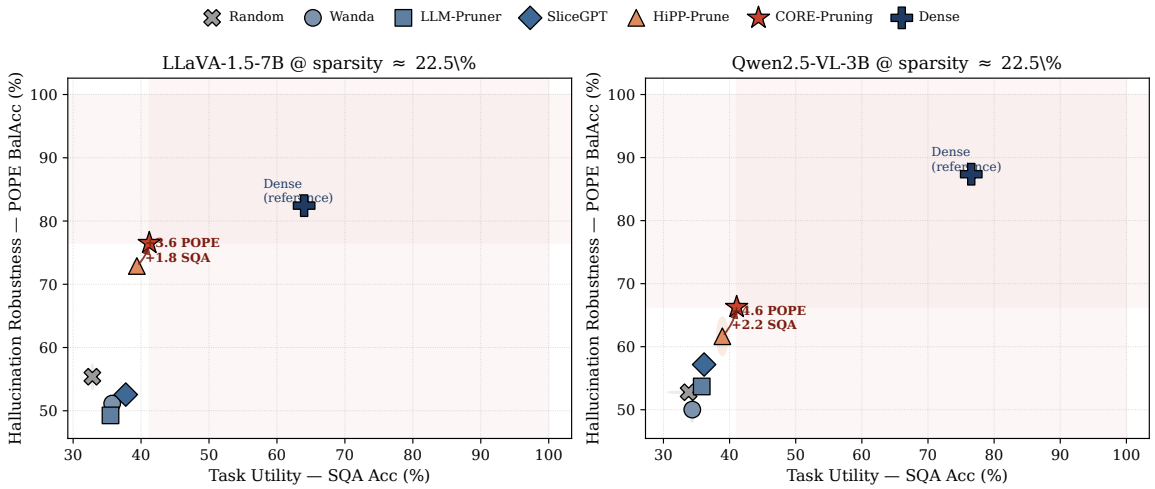

Figure 4: **POPE–ScienceQA trade-off at fixed sparsity ($\approx 22.5\%$).** Markers denote methods and ellipses show $\pm$std. For the plotted means, CORE-Pruning improves over HiPP-Prune on both POPE and SQA for both backbones instead of trading one metric for the other.

Method	**Sparsity** ≈ 0.225		**Sparsity** ≈ 0.325	
	POPE (%) ↑	SQA (%) ↑	POPE (%) ↑	SQA (%) ↑
Dense (Unpruned)	87.36	76.54	87.36	75.42
Random	52.77 ± 0.19	33.81 ± 3.06	50.86 ± 2.26	31.67 ± 2.10
Wanda	50.02 ± 1.98	34.36 ± 0.28	50.02 ± 1.98	29.73 ± 3.47
LLM-Pruner	53.69 ± 1.03	35.79 ± 0.43	50.39 ± 1.33	26.79 ± 0.57
SliceGPT	57.17 ± 0.19	36.15 ± 0.89	51.17 ± 0.19	31.15 ± 0.11
HiPP-Prune	61.65 ± 3.03	38.91 ± 0.83	53.14 ± 1.37	32.94 ± 0.88
CORE-Pruning (Ours)	$\mathbf{66.28 \pm 1.14}$	$\mathbf{41.07 \pm 0.69}$	$\mathbf{58.24 \pm 0.91}$	$\mathbf{35.63 \pm 0.57}$

Table 1: **Qwen2.5-VL-3B results under two sparsity budgets.** POPE BalAcc and SQA Acc are mean$\pm$std. For CORE-Pruning and HiPP-Prune, std is computed across four queried preference points. All metrics are in %.

els. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401.

Yvon Apedo, Martyna Poreba, Michal Szczepanski, and Samia Bouchafa. 2026. Svd-prune: Training-free token pruning for efficient vision-language models. *arXiv preprint arXiv:2604.11530*.

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James

Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*.

Lincen Bai, Hedi Tabia, and Raul Santos-Rodriguez. 2026. Hipp-prune: Hierarchical preference-conditioned structured pruning for vision-language models. *arXiv preprint arXiv:2603.06270*.

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, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Shwai He, Ang Li, and Tianlong Chen. 2024. Rethinking pruning for vision-language models: Strategies for effective sparsity and performance restoration. *arXiv preprint arXiv:2404.02424*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. 2024a. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European conference on computer vision*, pages 214–230. Springer.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, Yao Hu, and Shaohui Lin. 2024b. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. *arXiv preprint arXiv:2412.00876*.

Xiaohu Huang, Hao Zhou, and Kai Han. 2025. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19959–19973.

Lei Jiang, Zixun Zhang, Yuting Zeng, Chunzhao Xie, Tongxuan Liu, Zhen Li, Lechao Cheng, and Xiaohua Xu. 2025. DCP: Dual-cue pruning for efficient large vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21191–21204. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305.

Yinan Liang, Ziwei Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu. 2025. Efficientllava: generalizable auto-pruning for large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9445–9454.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

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

Ji Ma, Wei Suo, Peng Wang, and Yanning Zhang. 2025. Short-lvlm: Compressing and accelerating large vision-language models by pruning redundant layers. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3575–3584.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Greenewald, Jiri Navratil, Jerret Ross, and Jesus Rios. 2025. Revisiting group relative policy optimization: Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

James Seale Smith, Chi-Heng Lin, Shikhar Tuli, Haris Jeelani, Shangqian Gao, Yilin Shen, Hongxia Jin, and Yen-Chang Hsu. 2025. Flexigpt: Pruning and extending large language models with low-rank weight sharing. *arXiv e-prints*, pages arXiv–2501.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Theresa Batista-Navarro. 2025. Lvpruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4299–4308.

Rinyoichi Takezoe, Yaqian Li, Zihao Bo, Anzhou Hou, Guang Mo, and Kaiwen Long. 2026. Learnpruner: Rethinking attention-based token pruning in vision language models. *arXiv preprint arXiv:2604.23950*.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. 2020. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in neural information processing systems*, 33:6377–6389.

Zicong Tang, Ziyang Ma, Suqing Wang, Zuchao Li, Lefei Zhang, Hai Zhao, Yun Li, and Qianren Wang. 2025. CoViPAL: Layer-wise contextualized visual token pruning for large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20701–20714. Association for Computational Linguistics.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. 2020. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2025a. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22128–22136.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. 2025b. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24972–24982.

Jiedong Zhuang, Lu Lu, Ming Dai, Rui Hu, Jian Chen, Qiang Liu, and Haoji Hu. 2025. St3: Accelerating multimodal large language model by spatial-temporal visual token trimming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11049–11057.

A Training Procedure

Stage 1 — ASM pre-training. The ASM GCN is trained for 10K gradient steps on 20 randomly sampled pruning simulations using a meta-accuracy-drop objective, then frozen.

Stage 2 — Recovery proxy training. $\mathcal{M}_\xi$ is trained offline from 200 plans (100 epochs each) before policy optimization begins.

Stage 3 — Policy optimization. The variational encoder q_ϕ and decoder $\mathcal{T}_\psi$ are jointly optimized with GRPO (group size $G = 8$, learning rate 3×10^{-4}) under a Dirichlet$(1, 1, 1)$ preference distribution for 5000 episodes. Training preferences are sampled from a mixture of discrete anchors and Dirichlet$(1, 1, 1)$ to ensure Pareto coverage.

After convergence, the highest-reward plan for each preference vector of interest is selected and subjected to the same 800-step LoRA recovery used to train $\mathcal{M}_\xi$ to produce final evaluation numbers.

One policy update. Each update samples a preference vector, samples G latent codes from q_ϕ, decodes G complete pruning plans, computes ASM saliency for each sampled plan, evaluates POPE/SQA/compression rewards with SSR and RAMO terms, and applies the group-relative policy-gradient update. Thus "one-shot" refers to complete-plan decoding per policy sample; it does not preclude plan-conditioned reward and saliency evaluation.

B High-Sparsity Visualization

This appendix expands the high-sparsity result summarized in Section 4.2. The comparison from $\approx$22.5% to $\approx$32.5% sparsity is a stress test for layer allocation: once the pruning budget becomes more aggressive, small mistakes in where parameters are removed are amplified. We therefore plot each method as a slope rather than only reporting the two endpoints.

Figure 5 shows that all methods degrade at higher sparsity, but the degradation is not uniform. CORE-Pruning keeps a flatter POPE curve than HiPP-Prune and the heuristic baselines, indicating that preference-conditioned allocation remains useful when the feasible set is tighter. The SQA trend is smaller in absolute value but consistent with the main claim: the POPE gain is not obtained by simply giving up task utility.

C Additional Main Results

Table 2 gives the numerical counterpart of the LLaVA-1.5-7B bars in Figure 3. We keep the table in the appendix to preserve the eight-page main body while still making the exact means and standard deviations available. The comparison uses the same matched sparsity and recovery protocol as the main text. As noted in Section 4.1, the uncertainty columns summarize different sources of variation: calibration seeds for heuristic baselines, and preference-query sensitivity for HiPP-Prune and CORE-Pruning.

The table shows the same pattern as the main figure. Among non-learned baselines, SliceGPT gives the strongest POPE and SQA values, but it remains far below the preference-conditioned methods. CORE-Pruning improves over HiPP-Prune on both axes at matched sparsity, supporting the interpretation that the gains come from better layer allocation and recovery-aware search rather than a different compression budget.

D Ablation Studies

D.1 Preference Controllability

To evaluate zero-shot controllability, we fix a single trained CORE-Pruning model and query it with four preference vectors spanning the Pareto simplex. Table 3 reports raw numbers; Figure 6 provides two complementary views.

Per-axis monotonicity. Panel (a) of Figure 6 shows that each query deforms the radar polygon predictably: the Rob-heavy polygon is uniquely maximal along the POPE axis (78.17%), the Util-heavy polygon along SQA (42.90%), and the Uniform polygon along compression (sparsity 0.2289). Panel (b) quantifies this: as w_{rob} increases from 0.30 to 0.60, POPE rises by $+2.8$ points; as w_{util} increases from 0.30 to 0.60, SQA rises by $+2.4$ points; and as w_{comp} increases from 0.10 to 0.34, realized sparsity rises by ~ 0.004. These three monotone relations hold simultaneously without retraining in this anchor set, supporting the use of HVA's shared latent space for preference navigation.

D.2 Operating-Point Cloud and Pareto Navigation

To complement the four fixed queries, we sample a dense grid of preference vectors from Dirichlet$(1, 1, 1)$ and evaluate the plan produced

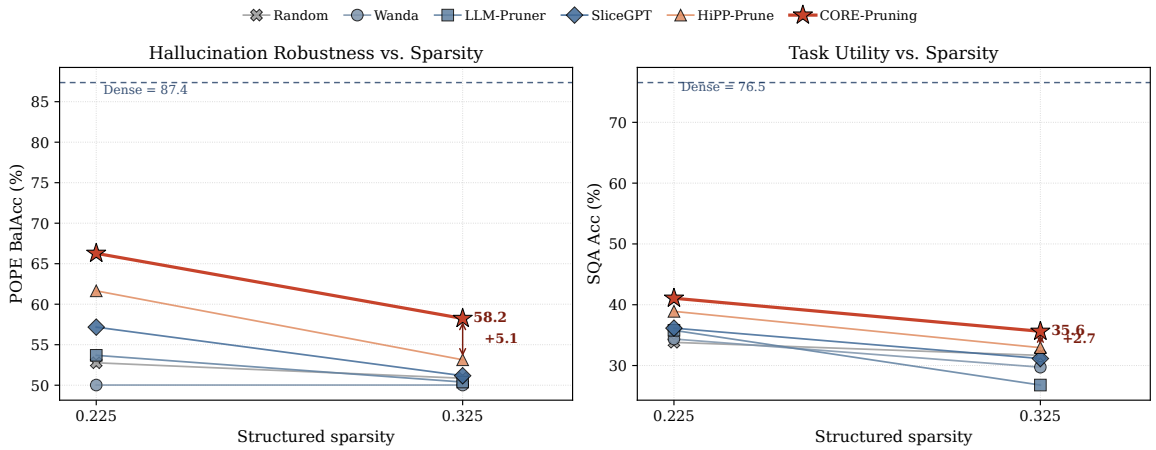

Figure 5: **Higher-sparsity behavior on Qwen2.5-VL-3B.** Left: POPE BalAcc; right: SQA Acc. Each line traces the change from ≈22.5% to ≈32.5% sparsity. The mean gap between CORE-Pruning and HiPP-Prune increases from +4.6 to +5.1 POPE points and from +2.2 to +2.7 SQA points.

Method	Sparsity ↓	POPE BalAcc (%) ↑	SQA Acc (%) ↑
Dense (Unpruned)	0.000	82.43	64.00
Random	0.225	55.38 ± 1.99	32.83 ± 0.24
Wanda	0.225	51.14 ± 0.20	35.75 ± 1.25
LLM-Pruner	0.215	49.27 ± 1.06	35.50 ± 1.00
SliceGPT	0.229	<u>52.56 ± 0.04</u>	<u>37.75 ± 0.25</u>
HiPP-Prune	0.225	72.89 ± 0.88	39.38 ± 0.74
CORE-Pruning (Ours)	0.225	**76.51 ± 0.62**	**41.21 ± 0.53**

Table 2: **LLaVA-1.5-7B results at matched sparsity** (≈22.5%)**.** POPE BalAcc and SQA Acc are mean±std; for CORE-Pruning and HiPP-Prune, std is computed across queried preference points. <u>Underline</u>: best non-learned baseline; **Bold**: best overall.

by the same fixed CORE-Pruning agent. Figure 7 plots the resulting operating-point cloud.

The cloud confirms two properties. Structurally, plans remain within the prescribed sparsity band, so preference querying respects the compression budget. Behaviourally, the color gradient suggests a gradual relationship between w_{rob} and POPE across the sampled simplex, extending the discrete trend observed in Figure 6(b).

D.3 Component Ablation

To isolate the contribution of each module, we deactivate each component individually and replace it with its HiPP-Prune counterpart: (a) **w/o HVA**: replaces the variational encoder–decoder with the independent Beta+Dirichlet factorization of HiPP-Prune; (b) **w/o ASM**: removes the GCN saliency module and falls back to static vision-attention cues; (c) **w/o SSR**: removes the spectral-radius penalty and restores HiPP-Prune's episodic SynFlow gate; (d) **w/o RAMO**: removes the recovery proxy from the reward. All variants are trained for 5000 episodes and subjected to the same 800-step LoRA recovery.

Attribution analysis. Figure 8(a) decomposes the +3.62 POPE / +2.16 SQA gap into per-module contributions. The ranking is the same on both axes: **HVA** (+1.35 POPE, +0.77 SQA) > **ASM** (+1.00, +0.50) > **SSR** (+0.72, +0.29) > **RAMO** (+0.55, +0.60). This ordering matches the design hierarchy: HVA restructures the action space (the largest departure from HiPP-Prune); ASM refines the state; SSR regularizes exploration; and RAMO re-weights the reward. Every module yields a positive, additive contribution with no redundant overlap.

Variance analysis. Figure 8(b) reports a property hidden in mean-only tables: removing SSR inflates POPE std from 0.62 to 1.41 (+0.79), whereas removing any other module leaves std within $[0.87, 1.34]$. This pattern is consistent with the claim that SSR's smooth penalty reduces the preference-query sensitivity introduced by episodic

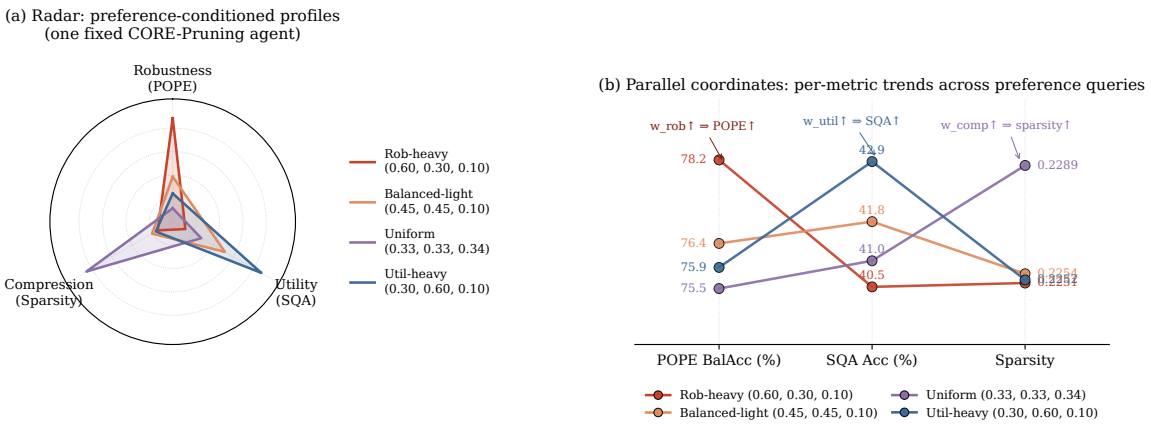

Figure 6: **Preference-query behavior of a single CORE-Pruning agent on LLaVA-1.5-7B.** *(a) Radar view:* each polygon corresponds to one preference query $\mathbf{w}$; radial extent encodes the realized metric normalized to the observed range. Among the four anchors, the robustness-heavy query has the highest POPE, the utility-heavy query has the highest SQA, and the uniform query has the highest realized sparsity. *(b) Parallel-coordinates view:* raw metric values for the same four queries. Cross-query std is 0.62 on POPE vs. 0.88 for HiPP-Prune, suggesting a smoother preference-to-plan mapping under these queries.

Preference $\mathbf{w}$	Sparsity ↓	POPE BalAcc (%) ↑	SQA Acc (%) ↑
(0.60, 0.30, 0.10)	0.2251	**78.17**	40.50
(0.45, 0.45, 0.10)	0.2254	76.43	41.75
(0.33, 0.33, 0.34)	**0.2289**	75.49	41.00
(0.30, 0.60, 0.10)	0.2252	75.93	**42.90**
mean ± std	0.2262 ± 0.0017	76.51 ± 0.62	41.54 ± 0.89

Table 3: **Preference querying on LLaVA-1.5-7B.** A fixed CORE-Pruning policy is queried with different preference weights. Bold indicates the highest value in each column.

Variant	POPE BalAcc (%) ↑	SQA Acc (%) ↑
CORE-Pruning (Full)	**76.51 ± 0.62**	**41.54 ± 0.89**
w/o HVA	73.84 ± 1.34	40.17 ± 1.09
w/o ASM	74.52 ± 1.18	40.65 ± 0.92
w/o SSR	75.08 ± 1.41	41.03 ± 1.12
w/o RAMO	75.43 ± 0.87	40.48 ± 0.79
HiPP-Prune (reference)	72.89 ± 0.88	39.38 ± 0.74

Table 4: **Component ablation on LLaVA-1.5-7B at sparsity ≈22.5% (numerical anchor for Figure 8).** Mean ± std across four preference queries with matched recovery.

gating. The reduction does not sacrifice the mean in this setting.

Summary. The visualizations give a compact summary: (i) grouped bars (Figure 3) establish Pareto dominance at fixed sparsity on both backbones; (ii) 2-D scatter (Figure 4) visualizes dominance of the means in the robustness–utility plane; (iii) slope plot (Figure 5) shows the advantage grows under more aggressive compression; (iv) radar + parallel coordinates (Figure 6) and 3-D cloud (Figure 7) show preference navigation from a single fixed agent; (v) waterfall + variance view (Figure 8) decomposes gains into four modules and shows SSR's role in reducing preference-query variance.

E Broader Impact and Limitations

Broader impact and limitations. CORE-Pruning is intended for settings where structured compression and object-hallucination behavior must be considered jointly. By reducing the cost of strong VLMs, such methods can broaden access to multimodal systems; they should still be paired with downstream validation in the target deployment domain. Our experiments focus on two representative backbones and two primary objectives, POPE and ScienceQA. Future work can extend the evaluation to larger models, additional hallucination and reasoning benchmarks, measured latency/throughput, and vision-encoder or cross-modal projection pruning. We release code and pre-trained policies to facilitate replication and extension.

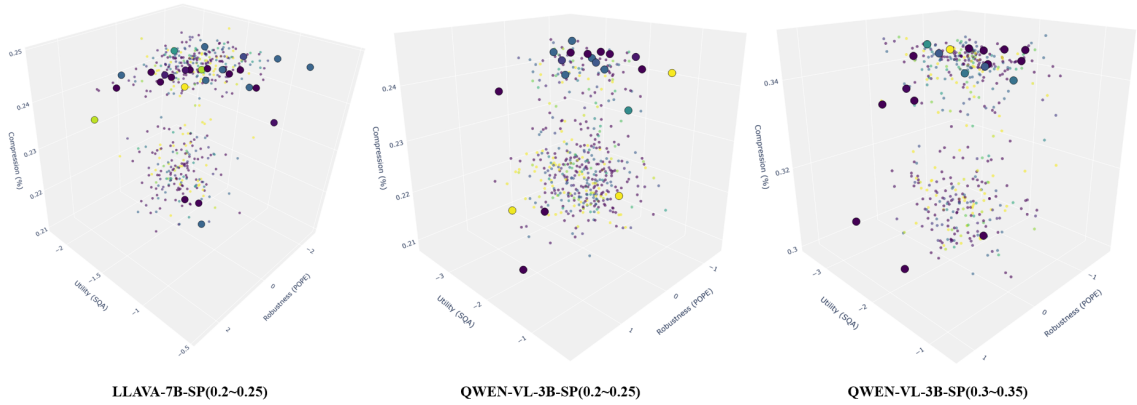

Figure 7: **Preference-conditioned operating points in 3-D (POPE, SQA, Compression) space.** Small dots are candidate plans sampled during GRPO; large dots denote Pareto-optimal blueprints selected for representative preference queries. Color encodes the robustness weight w_{rob} (dark = low, yellow-green = high). The smooth color gradient—visible beyond the four anchors—suggests that POPE varies gradually with w_{rob} over the sampled simplex. The CORE-Pruning cloud is shifted toward higher POPE/SQA values relative to HiPP-Prune under the plotted settings. Plans remain within the prescribed sparsity band.

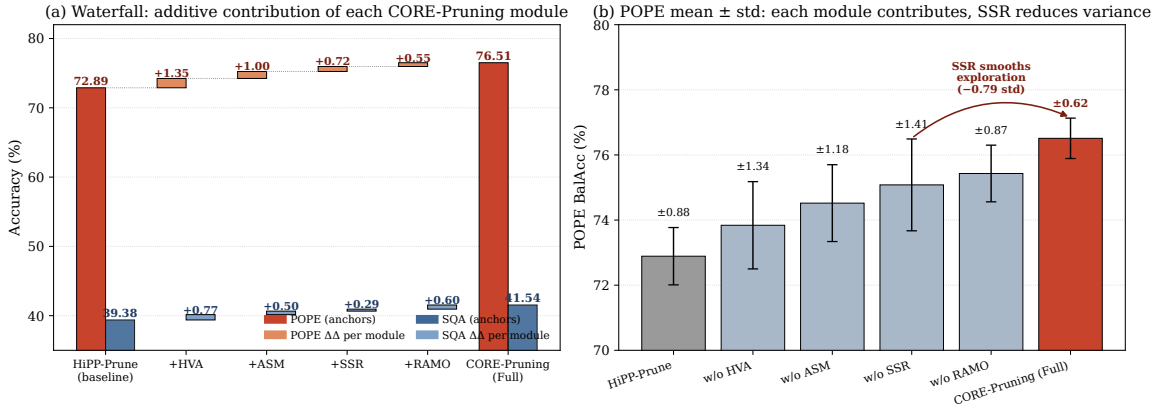

Figure 8: **Component ablation on LLaVA-1.5-7B ($\approx 22.5\%$ sparsity).** *(a) Waterfall view:* measured changes when individual modules are removed from the full system. HVA produces the largest mean drop in this setting, followed by ASM, SSR, and RAMO. *(b) Variance view:* POPE mean $\pm$ std across four preference queries for each variant. Removing SSR inflates std from 0.62 to 1.41 (the largest variance jump of any ablation), while removing any other module leaves std in $[0.87, 1.34]$. This pattern is consistent with the design goal of replacing discontinuous episodic gating with a smoother penalty.