

756 A ADDITIONAL EXPERIMENTAL RESULTS

757 A.1 ABLATION STUDIES

758 A.1.1 TOKEN POSITION SELECTION FOR HEAD ACTIVATIONS

759 Table 4: Head Selection Performance across Different Token Positions

760 Token Position	761 Last Token in Prefix	762 Robot State Token	763 First Action Token
764 MSE of Best head	0.004711	0.004544	0.004893
765 MSE of Worst head	0.005423	0.008530	0.007451
766 MSE Range	0.000712 (15.1%)	0.003987 (87.7%)	0.002558 (52.3%)

767 To determine the optimal token position for attention head selection, we evaluated activations extracted from three key positions in the π_0 architecture using our k-NN regression method (see Section 3.2): the last token of the prefix (summarizing multi-modal context), the robot state token (integrating robot state with prefix context), and the first action token (directly driving control actions).

768 As shown in Table 4, the choice of token position significantly affects the discriminative power of attention head selection. Using the method framework described in Section 3.2, the robot state token exhibits the largest performance variance, with an MSE range of 0.003987 (87.7% of the best MSE), indicating substantial differences between the most and least informative heads. In contrast, the last token in prefix shows the smallest variance with an MSE range of only 0.000712 (15.1% of best MSE), suggesting that most heads contain similar information at this position.

769 The large variance observed in the robot state token position makes it particularly suitable for head selection, as it clearly distinguishes between task-relevant and irrelevant attention heads. This aligns with our hypothesis that the robot state token, which has access to both multi-modal context and proprioceptive information, contains the most discriminative representations for robotic control tasks. Consequently, we select the robot state token as our primary source for attention head analysis and selection.

770 A.1.2 DISTANCE METRIC FOR K-NN REGRESSION

771 There are many choices of distance metrics for KNN-based feature comparison. Our features are activations extracted from the state token, which attends to all preceding tokens (i.e., vision and language tokens). Consequently, these activations encode visual, language, and robot-state information. We compared cosine similarity and Euclidean distance as the KNN metric, and found that heads selected using cosine similarity has lower MSE with respect to the ground-truth actions than those selected with Euclidean distance. We therefore choose cosine similarity as the default metric.

772 A.1.3 NUMBER OF NEAREST NEIGHBORS (K)

773 The value of k in our k-NN regression method (see Section 3.2) is not fixed but optimized per task. For each task, we search over $k \in \{10, 20, 30, 40\}$ and select the top 20 heads based on each candidate k value. We then evaluate the MSE performance of each resulting head subset and choose the k that has the lowest MSE loss.

774 A.2 ADDITIONAL EXPERIMENTS

775 A.2.1 HEAD SELECTION VIA CLASSIFICATION

776 This method employs a binary classification approach to identify task-relevant attention heads. We have a small support set that contains 20 episodes for both positive (target task) and negative (non-target tasks). Then, for each head independently, we compute class centroids by averaging activations across positive and negative samples separately. Each head is scored based on its discriminative ability using margin-based metrics, which is computed as the difference between each head’s similarity to the positive class centroid and its similarity to the negative class centroid for all support set samples (with

signs flipped for negative samples). The top-k highest-scoring heads are selected as the sparse subset, typically reducing from 144 to k (here we pick 20) heads while having stronger task discrimination than all heads. During inference, selected heads perform majority voting where each head contributes a vote based on cosine similarity to learned centroids, effectively capturing task-specific semantic patterns with significantly reduced computational overhead.

As shown in Table 5, while the top 20 selected heads achieve the highest performance (86.75% accuracy) compared to using all 144 heads (83.5% accuracy), the improvement is modest. More importantly, random head selection yields surprisingly competitive performance (82.5% accuracy), with only a 4.25% gap compared to the top-performing heads. Even the worst 20 heads achieve reasonable accuracy (81.25%), indicating that most attention heads possess some degree of task-relevant discriminative capability.

The voting margin reported in the performance evaluation represents the difference between correct and incorrect votes across all selected heads for each sample, with values ranging from $-K$ to K where K is the number of selected heads.

This observation suggests that the distinction between "best" and "worst" heads is not sufficiently pronounced for effective head selection. The relatively small performance gap across different head subsets implies that the task-relevant information is distributed across most attention heads rather than concentrated in a few specialized ones. Consequently, we did not adopt this classification-based head selection approach, as the limited discriminative ability between heads undermines the fundamental assumption that sparse head selection can significantly improve performance while reducing computational overhead.

Table 5: Classification Performance of Selected Attention Heads

	Top 20 heads	Worst 20 heads	All 144 heads	Random 20 heads
Accuracy	86.75%	81.25%	83.5%	82.5%
Voting Margin	9.13 / 20	6.57 / 20	45.62 / 144	6.75 / 20

A.2.2 ATTENTION MAP VISUALIZATIONS

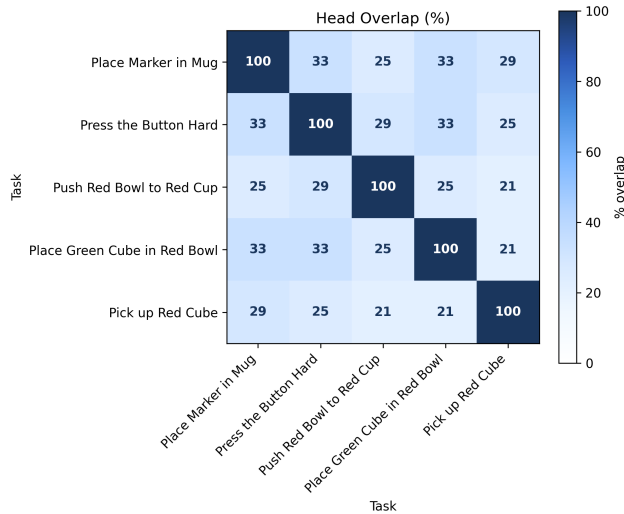


Figure 5: Head overlap percentage across different tasks, with diagonal elements representing 100% self-overlap.

Figure 5 shows that selected heads have low overlap between tasks (21-33% similarity), demonstrating that our method identifies task-specific rather than generic attention heads.

B IMPLEMENTATION ADDITIONAL DETAILS

B.1 FINE-TUNING DETAILS

In this setup we perform Robotic Steering with a mask-aware optimizer over LoRA adapters. Concretely, all non-LoRA weights in the Gemma are frozen by a freeze filter. The entire image encoder (SIGLIP) and the Action Expert branch are also frozen. We then update only the LoRA parameters attached to the query and output attention projections at the heads we selected, while explicitly freezing KV LoRA as they are shared per layer. The mask is constructed over the parameter tree so that gradient updates are applied only to those targeted slices, and ignored elsewhere.

We also allow a small set of non-attention adapters to update, including the action input/output projections, a time-conditioning MLP (in/out), the state projection, and LayerNorm gains, and the Gemma feed-forward (mlp) remains trainable. This enables a tightly controlled head-based LoRA adaptation on the selected attention heads, with all non-targeted weights held fixed.

B.2 OBSERVATION AND ACTION SPACE

We take both wrist image and external right image plus and 7 joint positions and 1 gripper position as observation input. All images are (1080,720), we first center-crop them to (720,720) and then resize to (224, 224). And for action space we are using 7 joint velocities and 1 gripper position as action space.

B.3 KEY HYPERPARAMETERS FOR FINE-TUNING

We perform Robotic Steering fine-tuning for 5000 timesteps with a CosineDecaySchedule as following:

- **Warmup steps:** 200
- **Peak learning rate:** 2.5×10^{-5}
- **Decay steps:** 5000
- **Final learning rate:** 2.5×10^{-6}
- **Total training steps:** 5000
- **Batch size:** 32

C ROBOTIC SETUP ADDITIONAL DETAILS

C.1 PLACE MARKER IN CUP

Task Description. The robot must grasp a small marker and accurately place it inside a target cup. **Evaluation Metrics.** Success rate based on whether the marker is successfully deposited in the cup. **Success Criteria.** Task is considered successful when the marker is fully contained within the cup boundaries.

C.2 PRESS THE BUTTON HARD

Task Description. Evaluation Metrics. Success Criteria.

C.3 PLACE GREEN CUBE IN RED BOWL

Task Description. Evaluation Metrics. Success Criteria.

C.4 PICK UP RED CUBE

Task Description. Evaluation Metrics. Success Criteria.

C.5 PUSH RED BOWL TO RED CUP

Task Description. Evaluation Metrics. Success Criteria.

C.6 TASK VARIATIONS

C.7 LIGHTING VARIATIONS

C.8 OBJECT PROPERTY VARIATIONS

Color Variations. Size Variations. Texture Variations.

C.9 ENVIRONMENTAL VARIATIONS

Camera Viewpoint Changes. Background Clutter. Table Surface Materials.