AVA-VLA: Improving Vision-Language-Action models with Active Visual Attention

Supplementary Material

A. Implementation Details

We report the implementation details of our proposed AVA-VLA framework based on the OpenVLA-OFT architecture, and the training details of all experiments.

Base OpenVLA-OFT architecture. Our main experiments are based on the OpenVLA-OFT architecture. It integrates a shared SigLIP-DINOv2 backbone for multi-image processing, a Llama-2 7B language model, a 3-layer MLP projector with GELU activation for mapping visual features into the language embedding space, a 2-layer MLP with GELU activation for projecting robot proprioceptive state to the language embedding space, and a 4-layer MLP with ReLU activation for continuous action generation. Distinct from the standard OpenVLA, this architecture replaces causal attention with bidirectional attention to enable parallel decoding, outputting chunks of L_c actions at each timestep.

AVA-VLA framework modifications. Our main experiments introduce the following modifications for deploying the AVA-VLA framework on the OpenVLA-OFT foundation model: 1) a 2-layer MLP with SiLU activation for mapping hidden state to the aforementioned recurrent state, 2) three 2-layer MLPs with SiLU activation for mapping visual features, instruction feature, and recurrent state from d-dimension to d'-dimension, respectively, 3) a feature-wise linear modulation (FiLM) to condition the visual features on the language instruction, 4) a cross-attention layer, a self-attention layer, a FFN, and a linear layer with Softmax activation for predicting the logits for enhancing or weakening each visual token, 5) replacement of the empty placeholder embedding with the recurrent state, 6) modification of the final attention weight matrix based on calculated soft weights vector from the AVA module.

The proposed AVA-VLA framework introduces only lightweight additional components on top of OpenVLA-OFT. In total, these AVA-related modules add fewer than 50M parameters, accounting for less than 1% of the full model size. Therefore, the parameter and compute overhead introduced by our modifications are negligible relative to the backbone model.

Training Details. For the experiments on the LIBERO benchmark, we use their corresponding official OpenVLA-OFT checkpoints. To fine-tune the AVA-VLA model, we apply LoRA [14] with a rank of 32 to the LLM backbone, vision encoder, action head and proprioceptive projector, while fully optimizing the proposed AVA mechanism. We set the observation sequence length $K = 4$. For efficient training, the gradient is detached between the second and the third timestep. Hyperparameters are set as follows: $\lambda = 1.0$, $c = 0.6$, $\gamma = [1.9, 0.1]$. The action chunk size is set to $\mathrm{L}_c = 8$. The batch size is set to 64. The model is trained for 40,000 gradient steps with an initial learning rate of 5e-4, which includes a warm-up phase by 10% of the value for stability. Additionally, a cosine learning rate scheduler and a maximum gradient norm of 1.0 is used. For the ablation study on the model backbone in Section 4.3, OpenVLA-OFT models follow standard implementation details [20] with varying initializations, and the AVA-VLA models are trained with implementation details described above using the corresponding OpenVLA-OFT models as initialization.

For the CALVIN benchmark, we train the base OpenVLA-OFT architecture following standard settings in [20] using official checkpoints. The AVA-VLA model is trained using the same configuration as the LIBERO benchmark, with the exception of setting $c = 0.2$ to account for the smaller region of interest.

For Mobile ALOHA real-world experiments, inputs include one third-person and two wrist-mounted camera images (left wrist + right wrist), we provide the implementation details of the three comparison methods. For the UniVLA baseline, following [5], we fine-tune the pre-trained checkpoint using the recommended configuration. We employ the latent action decoder on primary images to obtain latent action supervision. We incorporate proprioceptive states into the action head and integrate dual wrist camera feeds as additional LLM inputs. The action chunk size is set to 25. The model is fine-tuned for 30,000 steps with a learning rate of 3.5e-4, which is decayed to 3.5e-5 after 24,000 steps. For the OpenVLA-OFT baseline, following [20], we use the official OpenVLA checkpoints as initialization, and apply the LoRA technique with a rank of 32 to the vision encoder and LLM backbone, while the action head and proprioceptive projector are fully optimized. The action chunk size is set to $\mathrm{L}_c = 25$. The model is trained for 100,000 gradient steps with an initial learning rate of 5e-4. The learning rate is decayed to 5e-5 after 50,000 steps. The batch size is set to 64. For the AVA-VLA model, we utilize the trained OpenVLA-OFT model as initialization, and train the model for 20,000 gradient steps following our LIBERO hyperparameter settings, maintaining an action chunk size of $\mathrm{L}_c = 25$.

Training initialization and continued training. In our main training recipe, AVA-VLA is initialized from a post-

trained OpenVLA-OFT checkpoint. This design is intended to provide a better recurrent-state initialization and improve optimization efficiency of the proposed modules, rather than to gain performance simply by extending training. To verify this explicitly, we additionally compare OpenVLA-OFT and AVA-VLA under matched training settings in Appendix D.

B. Real-World Experiment Details

In this section, we report the additional details of the Mobile ALOHA real-world experiments, including the task suites and execution trajectories. We adopt AgileX Cobot Magic platforms: Based on Stanford's Mobile ALOHA project [1], this platform includes a differential-drive AGV base Tracer, dual-arm manipulators, and RGB-D sensors. A platform demonstration can be seen in Figure 5.

Figure 5. **AgileX Cobot Magic platforms.**

B.1. Real-World Task Suites

We introduce the detailed specifications for each task suite in our Mobile ALOHA real-world experiments:

Pick and Place

- Instructions: "put X into bucket".
- Task: Place the bucket in the center and put the simulated toy objects of which the instruction has given (yellow banana, green pepper, purple eggplant) into the bucket.
- Dataset: 450 demonstrations (150 per target).
- Episode length: 700 timesteps (28 seconds).
- Evaluation: 30 trials (10 for each).

Sequenced Instruction Understanding

- Instruction: "stack tower of hanoi".
- Task: Stack the medium tower on top of the large one first, and then stack the small one on top of the medium one.
- Dataset: 60 demonstrations (10 per formulation).
- Episode length: 600 timesteps (24 seconds).
- Evaluation: 24 trials (4 for each).

Flexible Object Folding

- Instruction: "fold towel twice".
- Task: First fold the towel vertically, then fold horizontally, and finally flatten it.
- Dataset: 30 demonstrations.
- Episode length: 900 timesteps (36 seconds).
- Evaluation: 24 trials.

Dexterous Action

- Instructions: "scoop X into bowl".
- Task: Move the bowl to the center of vision, pick up and use the shovel to scoop up different objects (corn, sesame, sunflower seeds) and transfer them into the bowl.
- Dataset: 60 demonstrations (20 of each small object).
- Episode length: 1000 timesteps (40 seconds).
- Evaluation: 24 trials (8 for each).

B.2. Execution Trajectories

We provide the execution trajectories of the four real-world task suites in Figure 6. The proposed AVA-VLA method can perform various tasks in real-world scenarios.

C. Additional Discussions

The proposed AVA-VLA framework is different from recent memory-augmented VLA models, such as MemoryVLA [43]. MemoryVLA relies on an explicit, large-scale memory bank for retrieval-based feature augmentation, while AVA-VLA adopts a formal POMDP formulation to compress historical interactions into a compact, implicit recurrent state. Moreover, MemoryVLA focuses on augmenting current features with historical tokens, while our AVA-VLA method utilizes the recurrent state to dynamically modulate and prune visual tokens at the input level, enabling active visual perception that focuses on task-relevant regions.

The significance of temporal modeling and memory mechanisms is well-established across various fields, such as Vision-Language Navigation (VLN) [13, 59, 60] and Reinforcement Learning [12, 32]. Unlike the explicit memory-bank architectures in VLN-BERT [13] and SafeVLA [59] or the LSTM-based aggregation in NaVid [60], our approach explicitly incorporates a recurrent state based on POMDP to enhance visual representations in a simple yet effective manner. Furthermore, while conceptually related to POMDP-inspired RL algorithms such as Recurrent-PPO [32] or DRQN [12], AVA-VLA is specifically tailored for VLA tasks, prioritizing visual processing efficiency and the focus on task-relevant features over general policy stability.

[1] "https://global.agilex.ai/products/cobot-magic"

Table 6. **Model performance under different perturbations in the LIBERO+ benchmark.** For each column, the average task success rate (%) of four task suites (LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long) under the given perturbation type is reported. The last column reports the average task success rate over seven perturbation types. The best results in each column of each group are highlighted in **bold**.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Average
One policy for all 4 suites								
WorldVLA [6]	0.1	**27.9**	41.6	43.7	17.1	10.9	38.0	25.0
π_0 [3]	13.8	6.0	58.8	85.0	81.4	**79.0**	68.9	53.6
π_0-FAST [36]	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
OpenVLA-OFT [20]	55.6	21.7	81.0	92.7	**91.0**	78.6	68.7	67.9
AVA-VLA (Ours)	55.5	25.9	**85.6**	**95.5**	88.9	78.0	**74.1**	**70.1**
One policy per suite								
OpenVLA [19]	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
NORA [15]	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA [5]	1.8	**46.2**	69.6	69.0	81.0	21.2	31.9	42.9
OpenVLA-OFT [20]	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
RIPT-VLA [48]	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
AVA-VLA (Ours)	**69.4**	34.9	**81.5**	**97.5**	**94.1**	**79.1**	**78.3**	**74.7**

Table 7. **Comparison under matched training settings.** The results on the LIBERO benchmark in terms of success rates (%) under the "one policy for all 4 suites" setting are reported. Both OpenVLA-OFT and AVA-VLA are initialized from the same pretrained OpenVLA checkpoint and trained with 100K gradient steps in a batch size of 256. The best results in each column are highlighted in **bold**.

Method	Spatial SR (%)	Object SR (%)	Goal SR (%)	Long SR (%)	Avg. SR (%)
OpenVLA-OFT	97.0	98.8	96.0	95.2	96.8
AVA-VLA	**98.4**	**99.4**	**98.4**	**96.8**	**98.3**

Table 8. **Ablation study of the loss design on the LIBERO benchmark.** The results in terms of success rates (%) under the "one policy for all 4 suites" setting are reported. We remove the L2 penalty regularizer L_ω while keeping all other training settings unchanged. The best results in each column are highlighted in **bold**.

Method	Spatial SR (%)	Object SR (%)	Goal SR (%)	Long SR (%)	Avg. SR (%)
AVA-VLA	**97.4**	**99.4**	**97.4**	**97.6**	**98.0**
AVA-VLA w/o L_ω	97.4	98.8	97.2	96.4	97.5

D. Additional Experimental Results

In this section, we provide additional evidence for three aspects of AVA-VLA: (i) the gain is not explained by extra training compute, (ii) the proposed design generalizes across benchmarks and controlled perturbations, and (iii) the learned attention is interpretable.

D.1. Comparison Under Matched Training Settings

To rule out the confounding effect of additional training compute, we conduct an additional matched-setting comparison (Table 7), where OpenVLA-OFT and AVA-VLA are initialized from the same pretrained OpenVLA checkpoint and trained under identical settings, including the same equivalent batch size and the same number of optimization steps. Under this controlled setup, AVA-VLA consistently outperforms OpenVLA-OFT (and its performance is even better than that reported in Table 1), indicating that the gain is not explained by a larger training compute alone. Combined with the small parameter overhead of AVA-VLA (<50M, <1% of the full model), these results suggest that the performance gain mainly comes from the proposed architectural synergy between recurrent-state initialization and active visual attention.

Table 9. **Ablation study of the two modules on the CALVIN ABC→D benchmark.** "+init" denotes enabling state-based initialization only, and "+ava" denotes enabling the AVA module only. The results are reported in terms of success rates (%) and average length. The best results in each column are highlighted in **bold**.

CALVIN ABC→D	Task completed in a row ↑ 1	2	3	4	5	Avg. len ↑
OpenVLA-OFT	96.9	92.0	85.7	80.4	72.9	4.28
+init	99.5	96.9	93.4	**90.0**	83.6	4.63
+ava	99.1	96.5	93.1	89.2	82.7	4.61
AVA-VLA	**99.6**	**97.6**	**94.1**	89.9	**84.1**	**4.65**

D.2. Module Ablation on CALVIN Benchmark

To further evaluate whether the effects of the two modules generalize beyond LIBERO, we conduct the same ablation study on the CALVIN benchmark. As shown in Table 9, both state-based initialization and AVA consis-

tently improve over OpenVLA-OFT, while their combination achieves the best overall results. Importantly, the gains become more pronounced as the task horizon increases. These results support the same conclusion as Table 4 in the main paper: state-based initialization preserves temporal belief across steps, AVA refines perception by suppressing irrelevant visual content, and the two components are complementary, especially in long-horizon settings.

D.3. Robustness on LIBERO+ Benchmark

The LIBERO+ [11] benchmark enables us to perform a systematic vulnerability analysis by introducing controlled perturbations across seven dimensions: camera viewpoints (change the viewpoint/pose and field-of-view of the third-person camera), robot initial states (change the manipulator's initial pose), language instructions (rewrite task instructions to increase linguistic richness and complexity), light conditions (vary illumination intensity, direction, color, and shadow patterns), background textures (modify table/scene textures and materials), sensor noise (inject photometric distortions into input images), and object layout (add confounding objects and/or shift the target object's position).

We evaluate the proposed method on the LIBERO+ benchmark using the AVA-VLA models trained on the LIBERO benchmark. We do not use additional data to train these models. The evaluation results of two different settings: "one policy for all 4 suites" and "one policy per suite" are reported in Table 6. The results of baselines in LIBERO+ benchmarks are based on original references [11]. The results show that the proposed AVA-VLA method achieves the best total results over the seven perturbation types on two different settings, demonstrating the superiority of the proposed framework. Notably, the AVA-VLA model exhibits strong robustness under the Light and the Layout perturbations, further demonstrating that the proposed AVA module helps the model enhance the important visual information and reduce the interference of unimportant parts on prediction, thereby improving the model's robustness under visual interference.

D.4. Effect of the L_ω Regularizer

We further analyze the regularization term L_ω introduced in Section 3.4 of the main paper. Specifically, we remove the L2 penalty on the soft attention weights while keeping all other training settings unchanged. Quantitative results on the LIBERO benchmark are reported in Table 8. Removing L_ω reduces the average success rate from 98.0% to 97.5%, with the most noticeable drop on the LIBERO-Long suite.

We also visualize the effect of removing the regularization term L_ω on the same task instance used in Figure 4 of the main paper. As shown in Figure 12, without L_ω, the learned soft attention becomes noticeably more dispersed and allocates more mass to irrelevant background regions. Compared with the full model visualization in Figure 4, this suggests that L_ω stabilizes the sparsity pattern of AVA and helps the model maintain task-relevant focus over time.

D.5. Attention Visualization Across Tasks

In Section 4.4, we visualize the soft weights $\boldsymbol{\omega}^t$ related to the corresponding visual tokens during the inference of one example from the LIBERO benchmark. Additionally, we present further visualizations of the soft weights calculated by the AVA module across a broader set of examples to demonstrate the proposed method's consistency.

Figure 7 and Figure 8 illustrate results from Mobile ALOHA real-world experiments, covering two task suites across three viewpoints. The results demonstrate the proposed framework's ability to focus on important visual information. Specifically, in the "put yellow banana into bucket" task, the model consistently locates and focuses on the objects requiring interaction: the yellow banana and the bucket. Similarly, for the "scoop sesame into bowl" task, the model accurately pinpoints the interaction target, such as the ladle handle in frame 375 of the right wrist-mounted camera.

Extended visualization results for simulated environments are presented in Figures 9, 10, and 11. Figure 9 displays the results for the continuous tasks "Lift red block table" and "Place in slider" from two viewpoints for the experiment on the CALVIN benchmark. Figure 10 and Figure 11 display the results for two tasks from two viewpoints for the experiment on the LIBERO benchmark, respectively. These additional visualization results on the simulation environments consistently corroborate our findings. The proposed AVA-VLA method can enable the VLA model to effectively enhance the perception of critical visual information while suppressing irrelevant regions, thereby improving the model's performance.

E. Limitations

Despite its strong performance, AVA-VLA still faces a fundamental challenge of POMDP modeling: small perception or state-estimation errors may accumulate over long horizons, gradually leading to belief drift and failures in precision-sensitive manipulation such as grasping or placement. See the provided qualitative failure cases in Figure 13. This issue is especially pronounced in long-horizon tasks such as LIBERO-Long, where performance drops more markedly under visual token reduction. A promising direction for future work is to improve the stability of recurrent state propagation, for example, through more robust state-update mechanisms, explicit error-correction strategies, or longer-horizon training schemes that better align the recurrent state with task-relevant environment dynamics.

Figure 6. **Real-world task execution.** Key observations from four long-horizon manipulation tasks.

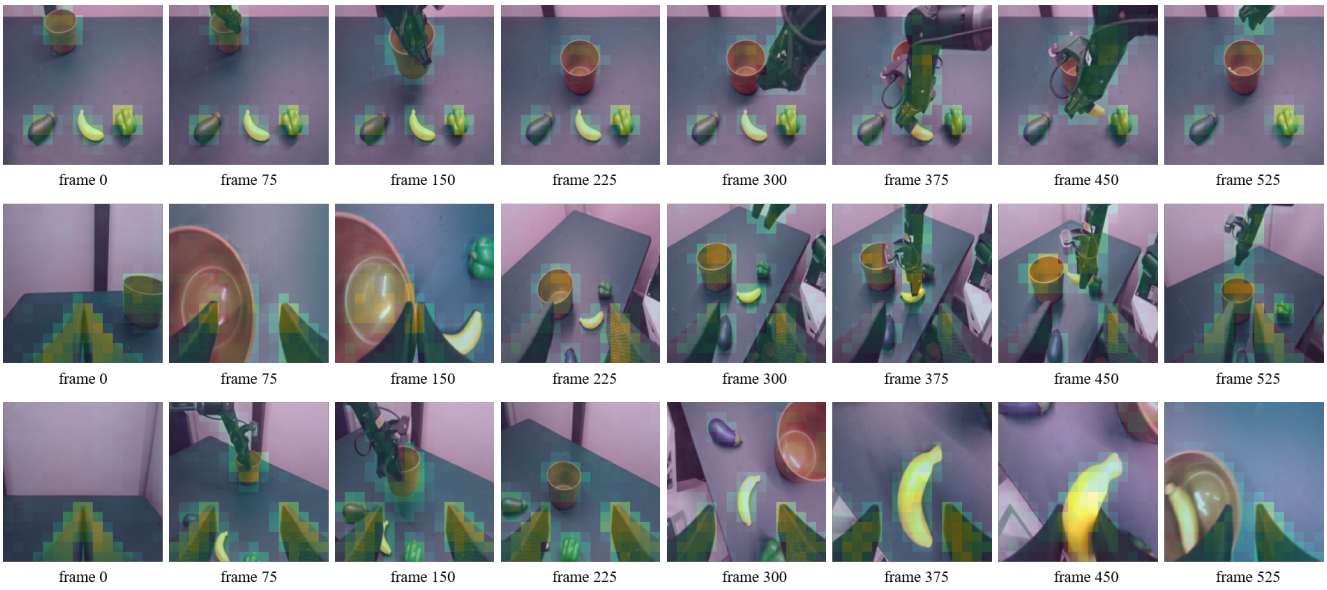


Figure 7. **Attention dynamics on Mobile ALOHA.** Soft weights for "put yellow banana into bucket" from three viewpoints.

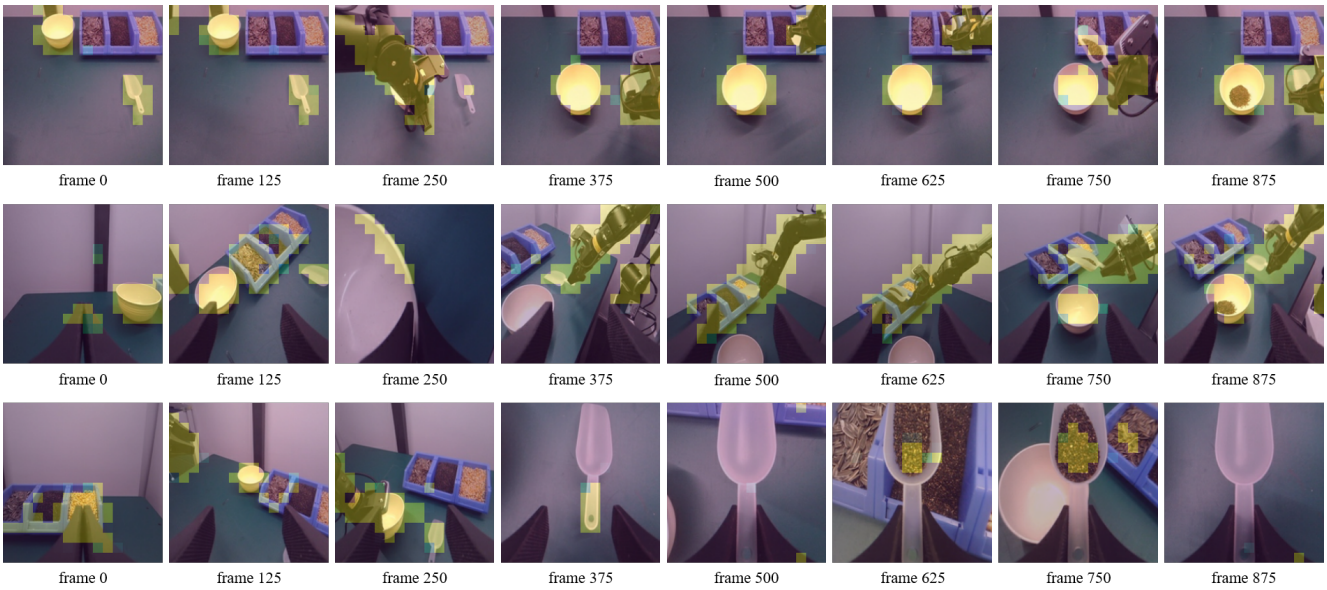


Figure 8. **Attention dynamics on Mobile ALOHA.** Soft weights for "scoop sesame into bowl" from three viewpoints.

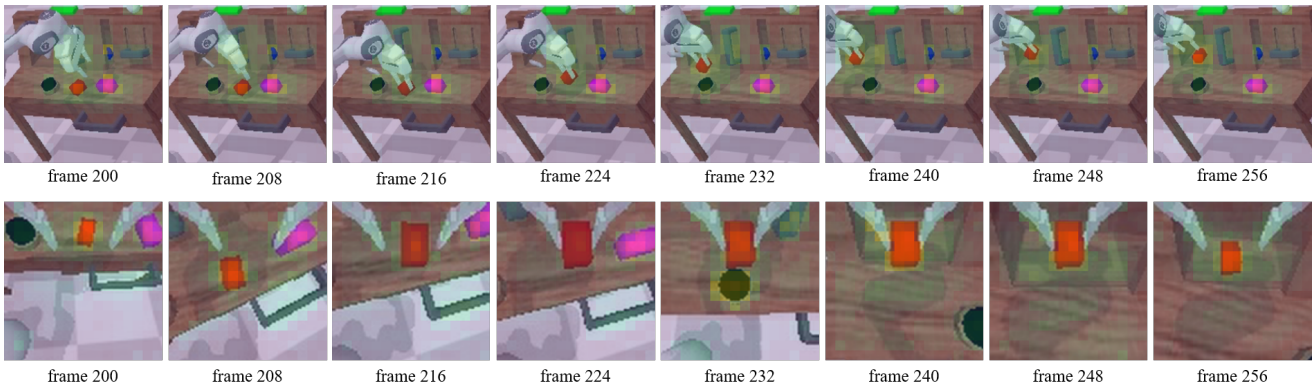


Figure 9. **Attention dynamics on CALVIN.** Soft weights for the continuous tasks "Lift red block table" and "Place in slider" from two viewpoints.

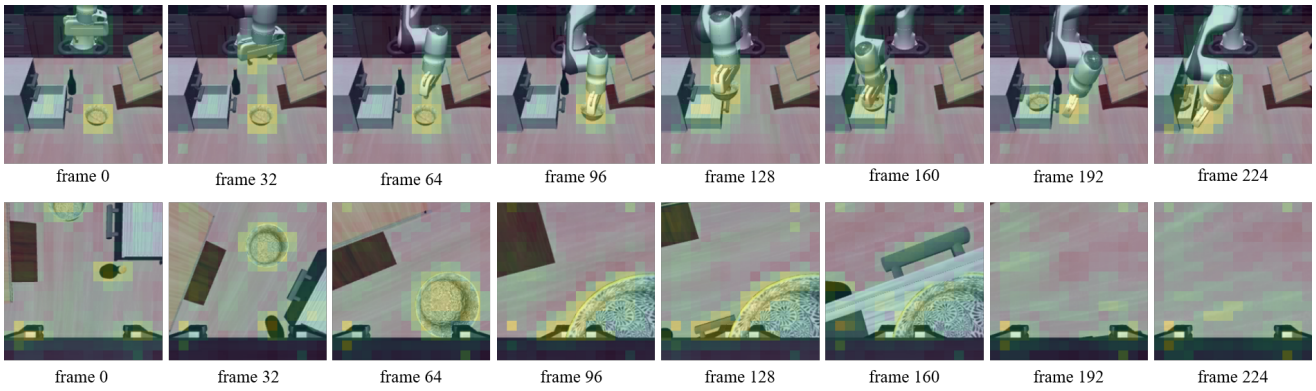


Figure 10. **Attention dynamics on LIBERO.** Soft weights for "put the black bowl in the bottom drawer of the cabinet and close it" from two viewpoints.

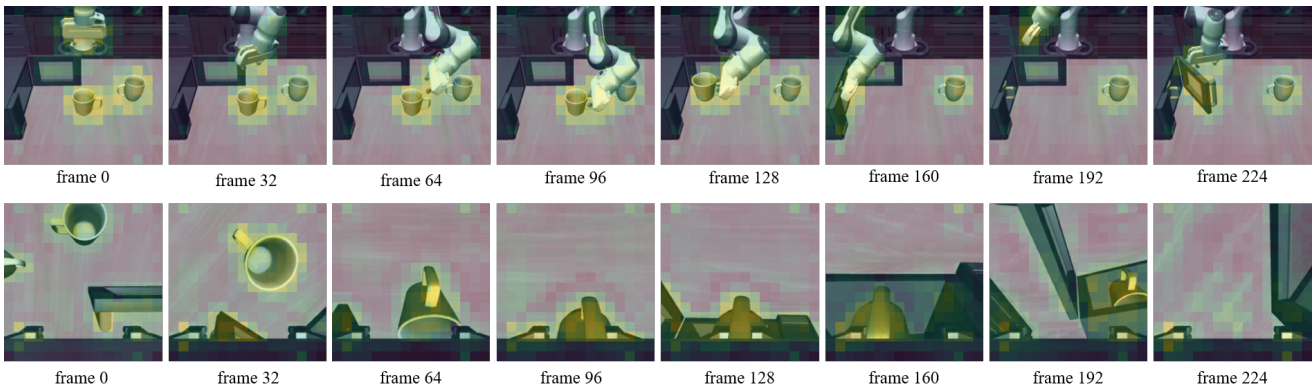


Figure 11. **Attention dynamics on LIBERO.** Soft weights for "put the yellow and white mug in the microwave and close it" from two viewpoints.

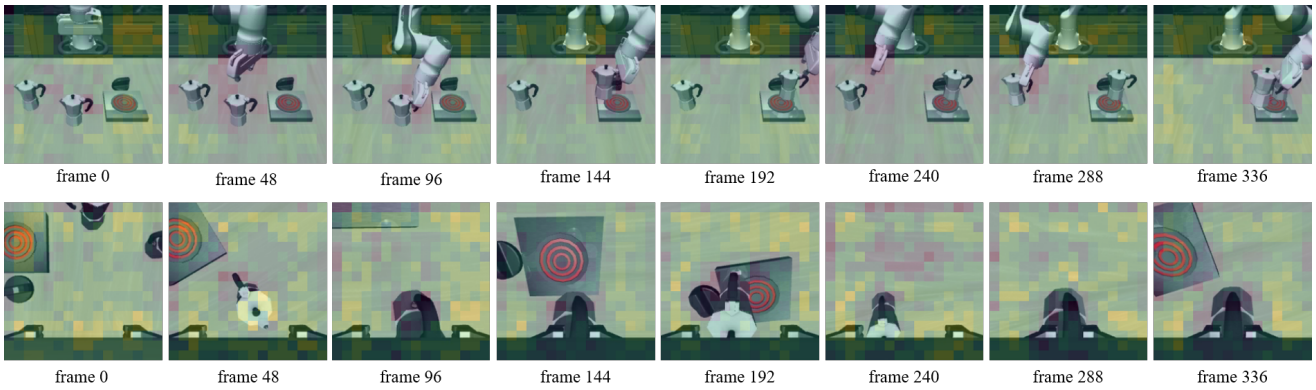
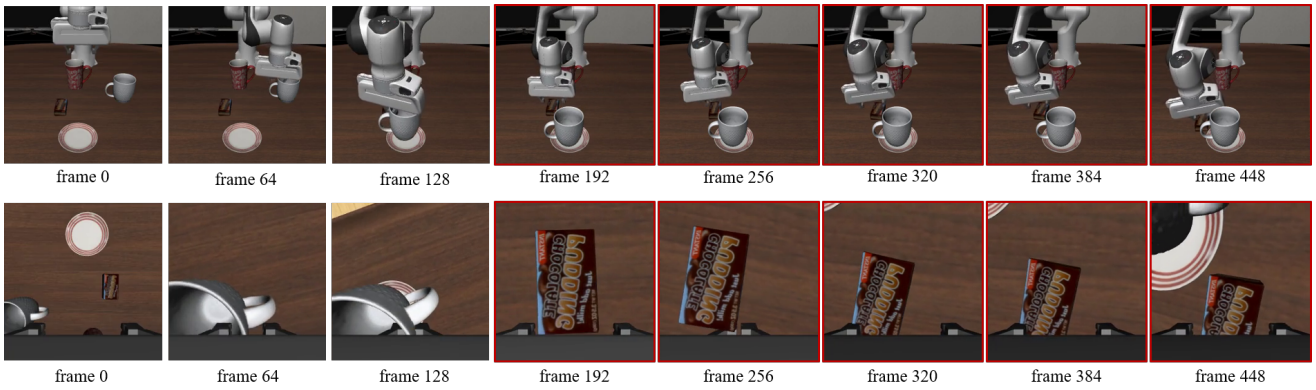
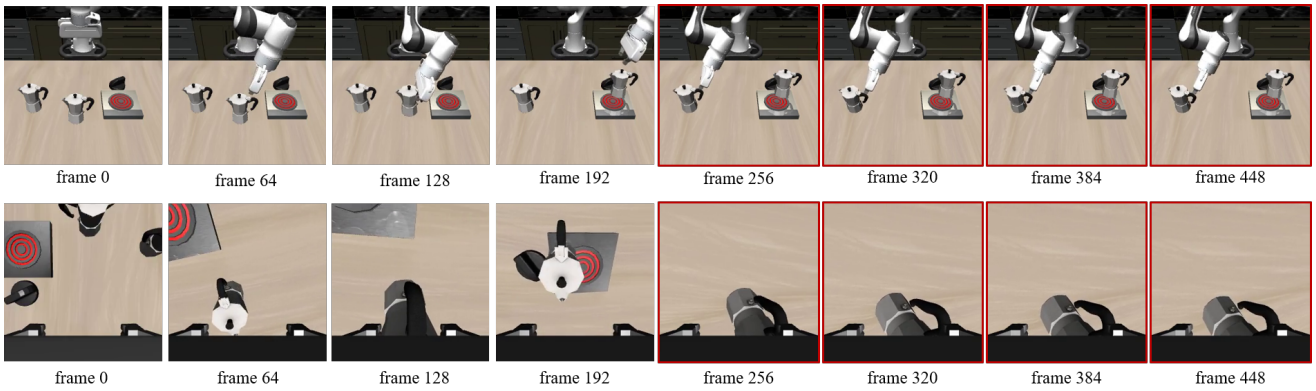


Figure 12. **Visualization of the soft weights without the regularizer L_ω on LIBERO.** Compared with the full AVA-VLA result shown in Figure 4, removing L_ω leads to more dispersed attention and increased responses on irrelevant background regions, indicating that the regularizer helps maintain more selective and structurally robust attention masks.



(a) Task: Put the white mug on the plate and put the chocolate pudding to the right of the plate.



(b) Task: Put both moka pots on the stove.

Figure 13. **Failure cases of AVA-VLA on LIBERO.** (a) The gripper fails to align with the chocolate pudding due to drifted spatial belief. (b) A slight positional deviation prevents the robot from securely grasping the moka pot handle. These cases illustrate how minor perceptual inaccuracies accumulate in the recurrent state, leading to drifted object/contact beliefs and eventual failures in precision-sensitive long-horizon tasks.