MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Test-Time Attention: Can Robots Better Follow Commands?

Liu, Jing; Wang, Ye; Suzuki, Kei; Koike-Akino, Toshiaki

TR2026-142 September 29, 2026

Abstract

Generalist Vision-Language-Action (VLA) policies promise broad manipulation capabilities, but the same breadth can weaken fidelity to a narrow deployment command. Repeated scene–object–action correlations and the large number of visual tokens relative to language tokens may cause visually suggested behaviors to compete with the operator's current command. We ask whether an already deployed VLA can be made more command-focused by redistributing attention only at inference time, without new demonstrations or policy retraining. Using pretrained pi 0.5 policy on a R1 Pro wheeled humanoid, we propose and evaluate 7 heuristic training-free interventions that either strengthen the current subtask text or reduce the influence of less command-relevant visual information in an outfitting toolbox setting. Because task success alone can conceal interactions with objects not specified by the current command, we introduce the Command Focus Ratio (CFR), which measures interaction selectivity, and the Target Interaction Rate (TIR), which measures how often the commanded target is engaged. Empirical studies suggest that test-time attention methods can improve the command fidelity of VLA policies without additional data collection or retraining, while the remaining gap in CFR highlights the need for stronger test-time mechanisms to suppress interactions with non-target objects.

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) IARL Workshop 2026

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Test-Time Attention: Can Robots Better Follow Commands?

Jing Liu[1], Ye Wang[1], Kei Suzuki[1], and Toshiaki Koike-Akino[1]

***Abstract*— Generalist Vision-Language-Action (VLA) policies promise broad manipulation capabilities, but the same breadth can weaken fidelity to a narrow deployment command. Repeated scene–object–action correlations and the large number of visual tokens relative to language tokens may cause visually suggested behaviors to compete with the operator's current command. We ask whether an already deployed VLA can be made more command-focused by redistributing attention only at inference time, without new demonstrations or policy retraining. Using pretrained $\pi_{0.5}$ policy on a R1 Pro wheeled humanoid, we propose and evaluate 7 heuristic training-free interventions that either strengthen the current subtask text or reduce the influence of less command-relevant visual information in an outfitting toolbox setting. Because task success alone can conceal interactions with objects not specified by the current command, we introduce the Command Focus Ratio (CFR), which measures interaction selectivity, and the Target Interaction Rate (TIR), which measures how often the commanded target is engaged. Empirical studies suggest that test-time attention methods can improve the command fidelity of VLA policies without additional data collection or retraining, while the remaining gap in CFR highlights the need for stronger test-time mechanisms to suppress interactions with non-target objects.**

I. Introduction and Motivation

Vision-Language-Action (VLA) policies promise a common interface for deploying one robot across many tasks. In a factory, for example, the same policy might organize tools, operate buttons, and execute multi-step assembly procedures. However, deploying a generalist policy across many tasks also creates a safety-relevant command-following problem. Suppose an operator asks the robot to pick up a wrench while a visually salient machine-start button is in view. A policy that associates the button with another task in its training data may press it even though the current command does not request that action. Likewise, an assembly procedure may require actions in a prescribed safe order, but a policy may act first on whichever familiar object or affordance is most visually salient. Such behavior can violate an operator's intent, process constraints, or safety interlocks even when the selected action is individually feasible.

Industrial robot safety has traditionally emphasized collision avoidance, physical separation, and safe human–robot collaboration [1]–[12]. Learning-enabled robots introduce an additional requirement: their behavior must remain faithful to the current command in scenes containing multiple valid affordances. Binary task success provides only a partial view of this requirement. It does not reveal whether a robot interacted with unrelated objects before succeeding, nor whether a failure arose because the policy pursued behavior associated with another trained task. We refer to this behavioral property as *command focus*: the extent to which a policy's physical interactions remain directed toward the object specified by the current command.

The need to evaluate command focus grows with the trend toward generalist robot manipulation policies. Recent work suggests that expanding the training distribution beyond narrow, single-task datasets to heterogeneous data spanning many scenes and tasks can both increase the range of behaviors available out of the box and improve generalization to new environments and tasks. For example, $\pi_{0.5}$ co-trains on data from multiple robots together with web data, high-level semantic predictions, and low-level actions to support open-world and long-horizon manipulation [13]. This increased breadth, however, also introduces repeated correlations among scenes, objects, language, and actions. The resulting architectures can further favor vision: language inputs typically contain only dozens of tokens, whereas visual inputs often contribute hundreds or thousands [20]. Under a narrow deployment command, these data and modality imbalances can cause visually suggested behavior to compete with the language instruction [20].

Several works try to improve instruction following through additional training, but target different forms of steerability. CAST introduces counterfactual language–action supervision and trains a CounterfactualVLA to make the commanded task identifiable when the same scene supports multiple plausible behaviors [17]. Its distractor-object experiments demonstrate gains in binary pick or pick-and-place success, but do not measure whether non-target contacts occur before task completion. Steerable Policies [18] broaden the command interface itself: their low-level policies are trained on synthetic instructions spanning task, subtask, atomic-motion, and grounded-coordinate abstractions, enabling a high-level controller to redirect behavior at different temporal and spatial resolutions. FineVLA [19] instead augments existing robot trajectories with fine-grained, action-aligned language annotations and evaluates steerability with respect to object color, object pose, approach direction, rotation, and active arm. However, CAST, Steerable Policies, and FineVLA all obtain these capabilities by changing the training data and learned policy weights. These approaches therefore do not directly address improving command focus for an already deployed VLA while keeping its existing policy parameters fixed.

Counterfactual Action Guidance (CAG) [20] provides the closest training-free alternative. CAG runs a language-conditioned branch and a language-masked counterfactual

[1]The authors are with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA.

branch, then extrapolates their predicted actions to amplify the component attributed to language. Unlike the three methods above, it requires neither new demonstrations nor policy-weight updates, making it directly applicable to a pretrained policy at inference time. Its evaluation reports the *grounding rate*, defined as whether the gripper contacts the target specified by the instruction regardless of task completion, together with the task success rate. However, neither metric necessarily records all unintended contacts during a rollout: a robot may touch several non-target objects before eventually contacting the target or completing the task. This limitation motivates us to define new metrics to measure interaction selectivity. Nevertheless, the reported simulation and real-world evaluations in [20] use fixed-base single-arm manipulators; locomotion, active viewpoint acquisition, and whole-body control are not involved. This distinction matters for humanoids because the visual target may initially lie outside the camera view and must be found through closed-loop movement. In our experiments, when another tool is visible but the commanded tool is not (see Figure 2 as an example), CAG can still move toward and attempt to pick up the visible non-target tool. This behavior may arise because CAG amplifies the language-induced difference between conditional and vision-only action predictions while retaining the underlying vision-conditioned action prior; it cannot ground a target that is absent from the current visual observation.

Our study therefore examines the closed-loop behavior of an unchanged pretrained VLA in a mobile humanoid setting and asks two practical questions: how focused are the robot's physical interactions on the commanded object, and can simple test-time attention interventions improve this focus without retraining? The target objects and scene are familiar to the policy: pretraining includes multiple tasks in the same scene, including long-horizon toolbox organization in which the robot opens the toolbox and places several tools inside in a demonstrated sequence. At evaluation time, however, the instruction may directly request a tool that appeared only later in that sequence, without requesting the preceding subtasks. The challenge is thus not novel-object generalization, but whether the current language command can override a learned temporal action-order prior and select a familiar object out of its demonstrated order. Unlike success-only evaluation, we account for target and non-target interactions over the entire rollout. We jointly report subtask success, Command Focus Ratio (CFR), and Target Interaction Rate (TIR) to determine whether the robot follows the current command directly, instead of following familiar task sequence, manipulating other objects, or remaining inactive.

II. Metrics and Methods

A. Command Focus and Interaction Metrics

Success alone does not reveal whether a policy consistently directs its behavior toward the commanded object. We therefore count target interactions, N_{target}, in which the robot physically interacts with the commanded object, and non-target interactions, N_{other}, involving any other object. A target interaction includes an attempted pickup or other physical contact and does not require the robot to contact the target successfully. Each trial contributes at most one target interaction count, regardless of repeated contact with the target object. Consequently, the number of successful trials is at most N_{target}.

We define the Command Focus Ratio (CFR) as

$$\text{CFR} = \frac{N_{\text{target}}}{N_{\text{target}} + N_{\text{other}}}. \tag{1}$$

A CFR of one means that every observed object interaction concerns the commanded target, whereas a value of zero means that all interactions concern other objects. The ratio becomes nan when the policy does not interact with any object, i.e., $N_{\text{target}} = N_{\text{other}} = 0$.

However, CFR alone does not indicate how often the policy interacts with the commanded object across trials. We therefore define the Target Interaction Rate (TIR) as

$$\text{TIR} = \frac{N_{\text{target}}}{N_{\text{trials}}}. \tag{2}$$

Thus, TIR is the fraction of trials containing a target-object interaction. It lies in $[0, 1]$, with larger values indicating better target engagement. Note that TIR defined here is more general than the Grounding Rate defined in [20] which measures whether the gripper contacts the target object. This is due to the *target interaction* used in TIR includes an attempted contact (e.g., trying to press the button but failed to touch it), therefore TIR captures a broader range of engagement than Grounding Rate. Together, subtask success, CFR, and TIR distinguish task completion, interaction selectivity, and target engagement.

B. Test-Time Attention Methods

We investigate two categories of test-time attention methods, emphasizing the text command or weaker visual influence from less-relevant image tokens. All interventions are applied only at inference time; the pretrained policy parameters remain fixed.

1) Emphasizing the Text Command: Note that many state-of-the-art VLA policies take in a main-task description and a subtask command. The main-task description provides context for the overall task, while the subtask command specifies the immediate behavior the robot should execute. We investigate the following methods that emphasize the subtask command at test time.

- **Additional Warning.** We append the explicit constraint "Do not interact with other objects!" to the current subtask command. This intervention tests whether an ordinary natural-language constraint is sufficient to suppress interactions with non-target objects.
- **Subtask Repeat.** In this approach, we repeat the current subtask command multiple times hoping to receive more attention. Let $Text_{main}$ denote the main-task prompt and $Text_{subtask}$ denote the subtask prompt. When there is no main-task prompt, the text input becomes $[Text_{subtask}, ..., Text_{subtask}]$.

When there is main-task prompt, the text input becomes $[Text_{main}, Text_{subtask}, ..., Text_{subtask}]$. This repetition increases the number of command-related text tokens without changing the model architecture or attention computation.

- **Subtask Emphasis.** Another way to emphasize the current subtask is to increase the attention received by the subtask command during action generation (e.g., via cross-attention in flow-matching process) by adding a positive bias β to their attention-key logits. For query q and text key i, the modified attention weight is

$$\alpha_{qi} = \mathrm{softmax}_i \left(\ell_{qi} + \beta \, \mathbf{1}[i \in \mathcal{S}] \right), \tag{3}$$

where ℓ_{qi} is the original attention logit and $\mathcal{S}$ is the set of subtask-token indices. This directly increases attention to the command span while leaving the text token sequence unchanged.

- **Subtask Only.** Intuitively, this approach removes the main-task description, leaving only the subtask command as the text input. This can be viewed as an extreme case of the Subtask Emphasis method, where the main-task description receives zero attention.

2) Suppressing Less-Relevant Image Tokens: In this category of methods, the three image interventions use the same prompt-conditioned relevance score. Let z_i be image token i and let p be the mean-pooled embedding of the current subtask command. We first compute cosine similarity and normalize it within each image:

$$s_i = \frac{z_i^\top p}{\|z_i\|_2 \|p\|_2}, \qquad r_i = \frac{s_i - \min_j s_j}{\max(\max_j s_j - \min_j s_j, \epsilon)}. \tag{4}$$

where the denominator is lower-bounded by $\epsilon = 10^{-6}$ for numerical stability. Thus, $r_i \in [0, 1]$ measures the relative alignment of image token i with the current subtask command.

- **Prune.** This approach aims to prune less relevant image tokens regarding the current command. We rank image tokens by s_i and retain certain percentage of image tokens (e.g., 75%). The retained tokens and their input masks are passed to the VLA in their original spatial order. This hard intervention removes the least command-relevant visual tokens.
- **Softprune.** Instead of hard removing certain percentage of image tokens, this approach retains all image tokens but downweights each image token's contribution according to the normalized relevance r_i defined in (4). More specifically, for each image token, we subtract a relevance-dependent bias from its attention-key logits:

$$\ell'_{qi} = \ell_{qi} - \lambda(1 - r_i). \tag{5}$$

Tokens with $r_i = 1$ receive no penalty, while tokens with $r_i = 0$ receive the full penalty $-\lambda$. Softprune therefore reduces visual competition without discarding scene information completely.

- **Blur.** This intervention uses token relevance to guide a spatially varying blur in pixel space; it does not blur the token embeddings directly. Each 224×224 input image produces a 16×16 grid of 256 patch tokens because the vision encoder uses 14×14 patches. Recall that each image token gets a corresponding relevance value r_i defined in (4). We reshape the corresponding $1 - r_i$ into this patch grid and linearly upsample it to a 224×224 pixel-level mask. Let $R \in \mathbb{R}^{16 \times 16}$ denote the relevance grid and let m_{xy} be the mask value at pixel (x, y):

$$\begin{aligned} \mathbf{m} &= \gamma \, \mathrm{Resize}_{\mathrm{bilinear}}(1 - R), \\ I'_{xy} &= (1 - m_{xy}) I_{xy} + m_{xy} B(I)_{xy}. \end{aligned} \tag{6}$$

Here, $B(I)$ is a 7×7 box-blurred image. Bilinear interpolation makes the mask $\mathbf{m}$ vary smoothly across pixels rather than remain constant within each patch. Highly relevant regions (e.g., $m_{xy} \approx 0$) remain comparatively sharp, whereas less-relevant regions lose visual detail. The modified image I' is then passed through the vision encoder again to produce the image tokens used by the policy.

Note that the Subtask Only approach can be straightforwardly combined with any other test-time attention methods. Further, all first category text-based methods can also be combined with image-based interventions.

III. Experiments

A. Task and Evaluation Protocol

We evaluate pretrained $\pi_{0.5}$ policy on an outfitting toolbox scene in a utility room from BEHAVIOR-1K simulation [14] using R1 Pro wheeled huamnoid. The scene contains toolbox and 5 different tools on a countertop, as well as two machines right under the countertop which have buttons that can be pressed (see Figure 1 for an example). We tested two policies provided by BEHAVIOR-1K 2025 Challenge NVIDIA Comet team [15], [16]. One policy was trained on all 50 tasks including the outfitting-toolbox task and tasks involving button pressing. The other policy was trained on 12 tasks that includes button-pressing and object-pickup behaviors but no toolbox-related tasks. The complete list of these 12 tasks is provided in [16]. The complete outfitting-toolbox task asks the R1 Pro robot to open the toolbox, pick up and place several objects into it. At test time, however, we issue a later subtask command that requests picking up one object: a wrench, screwdriver, or pliers, right at the beginning of the trial. The subtask command is like "Now pick up the Allen wrench from the countertop" for the wrench subtask. This setting tests whether a policy pretrained on a longer, multi-object task as well as other tasks follows the current subtask or instead continues interacting with other objects that were relevant during pretraining. To complete the command, the robot needs to navigate to the countertop, find and pick up the requested tool. This subtask is considered successful if the robot picks up the requested tool and keeps holding it at the end of the trial. We set the time limit of the trial to be 20,000 environment steps to allow the robot enough time to complete the subtask command. The robot is free to

Fig. 1. Representative R1 Pro observations of the evaluation scene. Top: head-camera view; Bottom Left: left-wrist view; Bottom Right: right-wrist view.

Fig. 2. Example of picking up Allen wrench subtask executed by the base policy trained on all 50 tasks. The Allen wrench (to the left of the toolbox) is not in the current view. The humanoid pressed the button of the machine and picked up the drill instead.

navigate and interact with other objects, but such interactions are counted as non-target interactions.

Figure 1 shows an example of the robot's three camera views of the evaluation scene, where multiple tools and machine controls provide competing visual affordances. Figure 2 illustrates the concerned command-focus problem: the base policy (trained on all 50 tasks) interacts with many visible non-target objects despite being commanded to pick up the Allen wrench. This example motivates evaluating target and non-target interactions over the full rollout rather than relying on terminal subtask success alone.

For the Subtask Repeat method, we repeat the subtask command 3 time, so the subtask command appears 4 times in the text prompt. For the Subtask Emphasis method, we set the attention bias $\beta = 0.1$. For the Prune method, we prune 25% image tokens. For the Softprune method, we set the attention penalty $\lambda = 0.3$. For the Blur method, we set the blur strength $\gamma = 0.75$. As the Subtask Only approach can be combined with any other test-time attention method, we test each method with and without the main-task description. For example, we evaluate the Blur method with the main-task description plus subtask command, and also evaluate the Blur method with the subtask command only. Note that the official $\pi_{0.5}$ policy was pretrained with main-task description plus subtask command [13].

For the Counterfactual Action Guidance (CAG) method [20], we notice that the R1 Pro humanoid has zero subtask success rate if applying it on all action dimensions including the base movement/navigation, so we only apply CAG on non-base action dimensions. We carefully tuned its guidance parameter to be 1.05. The base movement/navigation actions are controlled by language conditioning and are not modified by CAG.

For each requested object (wrench, screwdriver, or pliers), method, and prompt condition, we evaluate on the 10 validation instances provided by the BEHAVIOR-1K 2025 Challenge [14], where the object initial locations and robot initial positions vary. So the total number of trials N_{trials} for each method and prompt condition is 3 objects $\times$ 10 instances $=$ 30 trials. And we record the number of target interactions N_{target} and non-target interactions N_{other} during these 30 trials.

B. Results

We first evaluate on the policy which was trained on all 50 tasks, including outfitting-toolbox task and some tasks involving pressing the buttons. Fig. 3 shows the subtask success rate, Command Focus Ratio, and Target Interaction Rate for each method and prompt condition. Each aggregate point in Fig. 3 contains 30 trials across the three requested objects.

First, we can see that most of the methods performs better than the base policy using standard main-task description plus subtask command (black filled circle in Fig. 3), especially in terms of Command Focus Ratio.

Second, we observe that the text prompt with subtask command only, achieves higher Command Focus Ratio than the main-task plus subtask prompt for all methods except the Softprune approach. This is reasonable as the main-task description provides additional context that may encourage the robot to interact with other objects that are not requested in the current subtask command.

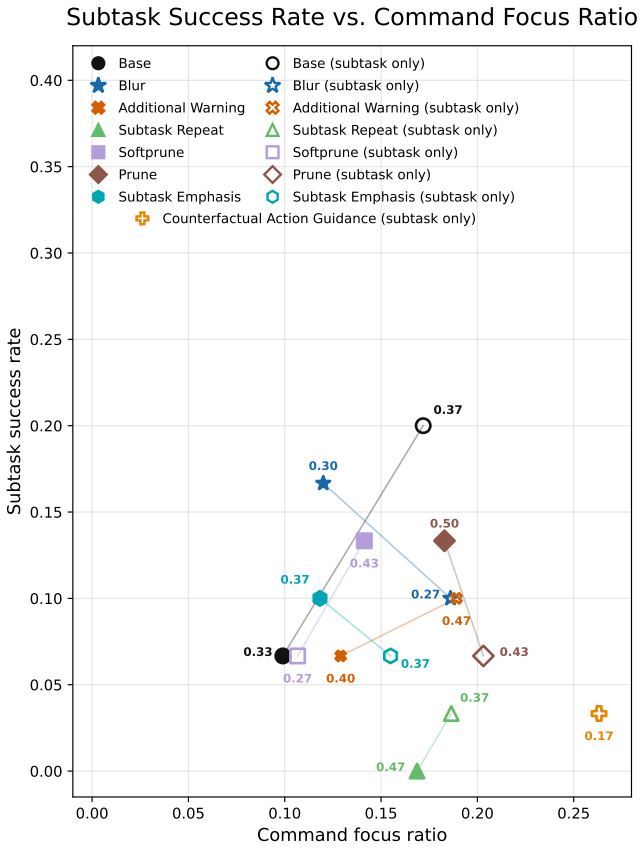

Fig. 3. Subtask success rate versus Command Focus Ratio (CFR) across three requested toolbox objects, using the policy trained on all 50 tasks from BEHAVIOR-1K 2025 Challenge. Open markers remove the main-task description in the prompt. We do not provide main-task description in the Counterfactual Action Guidance method. Each numeric point label is the Target Interaction Rate (TIR), the fraction of trials with target-object interaction. Lines connect the two prompt conditions for the same test-time method.

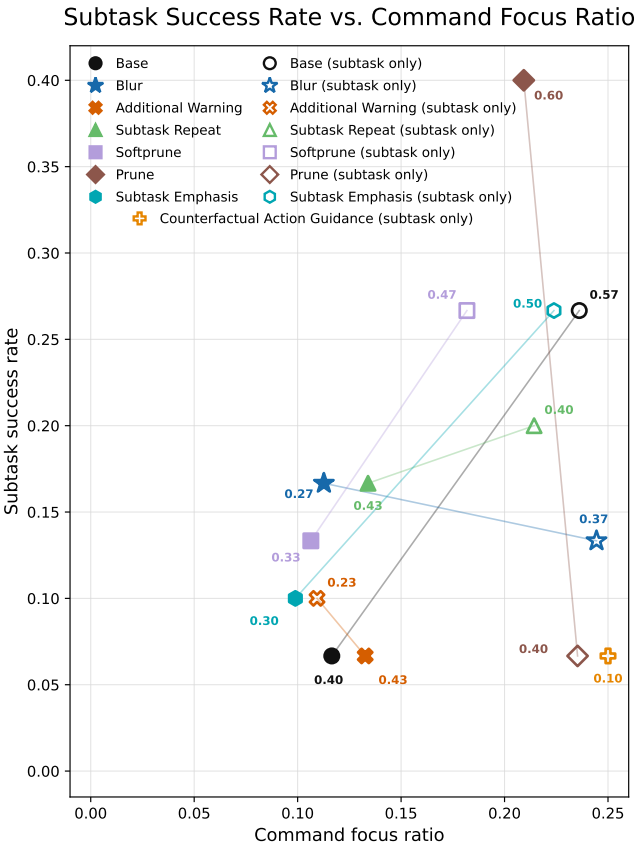

Fig. 4. Subtask success rate versus Command Focus Ratio (CFR) across three requested toolbox objects, using the policy trained on 12 tasks from BEHAVIOR-1K 2025 Challenge, which do not include outfitting-toolbox task. Open markers remove the main-task description in the prompt. We do not provide main-task description in the Counterfactual Action Guidance method. Each numeric point label is the Target Interaction Rate (TIR), the fraction of trials with target-object interaction. Lines connect the two prompt conditions for the same test-time method.

Third, the Subtask Only method with base model achieves highest subtask success rate (0.37), while the Prune method with main-task description achieves the highest TIR (0.50) among all methods. The Counterfactual Action Guidance (CAG) method achieves the highest CFR, however, both its success rate and TIR are significantly lower than most of other methods.

Overall, none of the methods achieve a high Command Focus Ratio. This motivates us to further evaluate on another policy that was trained on 12 tasks, which *does not* include outfitting-toolbox task, to see whether this would achieve higher CFR, as the policy did not learn demonstrated temporal action-order prior on outfitting-toolbox task from its training data. Fig. 4 shows the subtask success rate, Command Focus Ratio, and Target Interaction Rate for each method and prompt condition. Each aggregate point in Fig. 4 contains 30 trials across the three requested objects.

As expected, all methods with subtask command only, except the CAG and Additional Warning methods, indeed improve the CFR using this policy, compared to using the policy trained on all 50 tasks.

Besides, we can see that most of the methods performs better than the base policy using standard main-task description plus subtask command (black filled circle in Fig. 4).

Third, we observe that the text prompt with subtask command only, achieves higher Command Focus Ratio than the main-task plus subtask prompt for all methods except the Additional Warning approach. This observation is similar to that of Fig. 3.

Overall, the Prune method with main-task description achieves the highest success rate and TIR among all methods. The Counterfactual Action Guidance (CAG) method achieves the highest CFR, however, its success rate and TIR are the lowest among all methods. The Blur method without main-task description achieves comparable CFR to the CAG method, but its success rate and TIR are significantly higher than those of the CAG method.

However, none of the methods achieve a high Command Focus Ratio. Soft manipulating attention strength, e.g., Subtask Emphasis and Softprune methods, does not even improve CFR over the base policy in Fig. 4. They only show some improvement, in terms of subtask success rate, over the

bsae policy when main-task description presents. However, note that the policy was not pretrained with such soft manipulated attention mechanism. Light fine-tuning may be needed to adapt the policy to such soft-manipulated attention mechanism. We also expect light fine-tuning may help further improve the performance of the Additional Warning, Subtask Repeat, Blur, and Prune methods.

IV. Discussion and Future Work

This work studies command focus in pretrained VLA policies controlling a wheeled humanoid. Because task success alone does not reveal interactions with objects unrelated to the current command, we introduce the Command Focus Ratio (CFR) and Target Interaction Rate (TIR) to measure interaction selectivity and target engagement, respectively. We propose and evaluate several training-free test-time attention interventions that strengthen subtask text or suppress less command-relevant visual information. We also tested an approach that uses a policy which was *not* trained on related task. Our results show that these interventions as well as using alternative policy can improve command following, although further advances are needed to achieve consistently high command focus and task success.

Among all tested methods, under the policy that was not trained on related task, pruning less-relevant image tokens based on the subtask command while retaining the main-task description achieves a significantly higher success rate and a competitive Command Focus Ratio. Visual-token pruning is a popular approach to accelerate VLM inference [21]–[24]. Our use instead examines whether subtask command conditioned pruning can improve behavioral focus of the robot. One future work is to further develop the Prune method by leveraging the objects mentioned in the main-task description or training data, to prune image tokens that have *high* relevance to the distracting objects. We can also expand the evaluation metrics for the test-time attention methods to include the inference efficiency aspect.

Another future work is to further study the combination of the proposed test-time attention methods, e.g., combining the text-based methods with image-based methods. We also plan to study the effect of light fine-tuning on the proposed test-time attention methods, e.g., fine-tuning the policy to adapt to soft-manipulated attention mechanisms.

References

[1] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018, doi: 10.1016/j.mechatronics.2018.02.009.

[2] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human–robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008, doi: 10.1016/j.mechmachtheory.2007.03.003.

[3] M. Zinn, B. Roth, O. Khatib, and J. K. Salisbury, "A new actuation approach for human-friendly robot design," *Int. J. Robot. Res.*, vol. 23, nos. 4–5, pp. 379–398, 2004, doi: 10.1177/0278364904042193.

[4] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *Int. J. Robot. Res.*, vol. 22, no. 5, pp. 281–297, 2003, doi: 10.1177/0278364903022005001.

[5] J. Heinzmann and A. Zelinsky, "Quantitative safety guarantees for physical human–robot interaction," *Int. J. Robot. Res.*, vol. 22, nos. 7–8, pp. 479–504, 2003, doi: 10.1177/02783649030227004.

[6] A. Bicchi and G. Tonietti, "Fast and soft-arm tactics: Dealing with the safety–performance trade-off in robot arms design and control," *IEEE Robot. Autom. Mag.*, vol. 11, no. 2, pp. 22–33, 2004, doi: 10.1109/MRA.2004.1310939.

[7] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2006, pp. 1623–1630, doi: 10.1109/IROS.2006.282053.

[8] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human–robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2008, pp. 3356–3363, doi: 10.1109/IROS.2008.4650764.

[9] R. Schiavi, A. Bicchi, and F. Flacco, "Integration of active and passive compliance control for safe human–robot coexistence," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2009, pp. 259–264, doi: 10.1109/ROBOT.2009.5152571.

[10] A. De Luca and F. Flacco, "Integrated control for pHRI: Collision avoidance, detection, reaction and collaboration," in *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, 2012, pp. 288–295, doi: 10.1109/BIOROB.2012.6290917.

[11] J. A. Marvel, "Performance metrics of speed and separation monitoring in shared workspaces," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 405–414, 2013, doi: 10.1109/TASE.2013.2237904.

[12] F. Flacco, T. Kroeger, A. De Luca, and O. Khatib, "A depth space approach for evaluating distance to objects with application to human–robot collision avoidance," *J. Intell. Robot. Syst.*, vol. 80, suppl. 1, pp. 7–22, 2015, doi: 10.1007/s10846-014-0146-2.

[13] Physical Intelligence *et al.*, "$\pi_{0.5}$: A vision-language-action model with open-world generalization," arXiv:2504.16054, 2025.

[14] C. Li *et al.*, "BEHAVIOR-1K: A human-centered, embodied AI benchmark with 1,000 everyday activities and realistic simulation," arXiv:2403.09227, 2024.

[15] J. Bai *et al.*, "OpenPI Comet: Competition solution for 2025 BEHAVIOR Challenge," arXiv:2512.10071, 2025.

[16] Team Comet, "OpenPI-Comet," GitHub repository. [Online]. Available: https://github.com/mli0603/openpi-comet

[17] C. Glossop, W. Chen, A. Bhorkar, D. Shah, and S. Levine, "CAST: Counterfactual labels improve instruction following in vision-language-action models," arXiv:2508.13446, 2025.

[18] W. Chen *et al.*, "Steerable vision-language-action policies for embodied reasoning and hierarchical control," arXiv:2602.13193, 2026.

[19] X. Hu *et al.*, "FineVLA: Fine-grained instruction alignment for steerable vision-language-action policies," arXiv:2605.27284, 2026.

[20] Y. Fang, Y. Feng, D. Jing, J. Liu, Y. Yang, Z. Wei, D. Szafir, and M. Ding, "When vision overrides language: Evaluating and mitigating counterfactual failures in VLAs," arXiv:2602.17659, 2026.

[21] L. Chen, H. Zhao, T. Liu, S. Bai, J. Lin, C. Zhou, and B. Chang, "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[22] Y. Shang, M. Cai, B. Xu, Y. J. Lee, and Y. Yan, "LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models," arXiv:2403.15388, 2024.

[23] Y. Zhang *et al.*, "SparseVLM: Visual token sparsification for efficient vision-language model inference," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2025.

[24] S. Yang, Y. Chen, Z. Tian, C. Wang, J. Li, B. Yu, and J. Jia, "VisionZip: Longer is better but not necessary in vision-language models," arXiv:2412.04467, 2024.