Confidence Calibration in Vision-Language-Action Models

Anonymous authors
Paper under double-blind review

Abstract

Trustworthy robot behavior requires not only high levels of task success but also that the robot can reliably quantify how likely it is to succeed. To this end, we present a first-of-its-kind study of confidence calibration in vision-language-action (VLA) foundation models, which map visual observations and natural language instructions to low-level robot motor commands. We establish a confidence estimation baseline for VLAs, examine how task success relates to calibration error and how calibration evolves over time, and introduce two lightweight techniques to remedy the miscalibration we observe: prompt ensembles and action-wise Platt scaling. Our aim in this study is to begin to develop the tools and conceptual understanding necessary to render VLAs trustworthy via reliable uncertainty quantification.

1 Introduction

Confidence calibration, or the degree to which a model's predicted probabilities reflect the true likelihood of its predictions being correct, is a cornerstone of reliable machine learning systems (Guo et al., 2017). When a model is well-calibrated, downstream decision-makers (e.g., humans, planning algorithms, or safety monitors) can trust that a confidence estimate of 95% implies that the predicted outcome will occur roughly 95% of the time (see Figure 1). Mismatches between confidence and actual outcomes can have severe negative consequences in high-stakes settings such as medical diagnosis or autonomous driving, motivating a rich literature on improving calibration in deep learning models (Minderer et al., 2021; Gal & Ghahramani, 2016; Tian et al., 2023; Kumar et al., 2019; Kadavath et al., 2022).

Recently, the field of robotics has embraced a new class of *vision-language-action* (VLA) foundation models (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2024; 2025). Leveraging large-scale, multistage pretraining, these models translate visual observations and natural language instructions into low-level joint-space commands, unifying multimodal perception with motor control. Current VLA systems already demonstrate previously unattainable generalization across environments, tasks, and robot embodiments. Because these models encode broad semantic and visuomotor priors from large-scale pretraining, they can be efficiently fine-tuned to new robots and downstream tasks, yielding large gains over training from scratch.

Since VLAs are designed for closed-loop interaction with the world, knowing when and how strongly to trust their actions is critical. Consider a robot performing a task in a safety-critical environment, or attempting to manipulate a valuable, fragile object. In these scenarios, if the policy can accurately express uncertainty, then costly or dangerous accidents can be avoided, such as by refining the instruction or deferring the task to a human. Despite the importance of calibrated confidence, basic questions remain largely unaddressed, e.g., whether VLAs are calibrated, how calibration evolves over the task horizon, and how it can be improved.

Contributions. To bridge this gap, we present a first-of-its-kind study of calibration in VLAs, identifying key open questions and introducing practical remedies for miscalibration. Our contributions include: **(1)** We evaluate the relationship between task success and calibration error across multiple benchmarks and VLA variants, finding that the model architecture and training objective may play a role in determining this relationship. **(2)** We establish and validate confidence estimation baselines for an important class of VLA models. **(3)** We analyze calibration over task time, identifying natural points for risk-aware intervention. **(4)** We propose a lightweight method for ensembling confidence scores across semantically equivalent

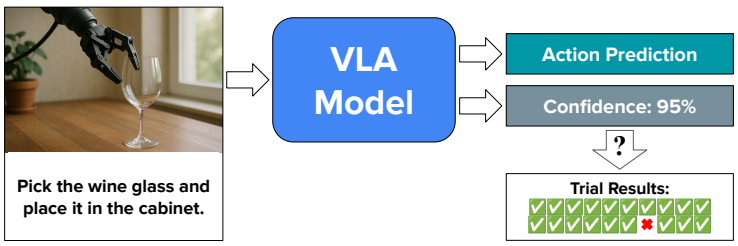

Figure 1: To be trustworthy, a robotic system must be able to reliably express its confidence in its ability to successfully complete a task trial, especially in high-stakes and open-world domains. A well-calibrated robot policy produces confidence estimates that align with its probability of trial success, ideally updating its estimate as the task progresses. For example, the robot should succeed on 95% of instances for which it expresses 95% confidence.

instructions, consistently cutting calibration error across models and tasks. **(5)** We discover systematic over-/underconfidence in different action dimensions and propose a method to recalibrate each action dimension independently to produce more reliable confidence estimates. Our aim in this study is to begin to develop the tools and conceptual understanding necessary to render VLAs not only highly performant but also highly trustworthy via reliable uncertainty quantification.

2 Related Work

Confidence Calibration A model is considered well-calibrated when the confidence (i.e., probability) it assigns to an outcome matches the long-run frequency of that outcome. Deviation from this condition is often quantified via expected calibration error (ECE) (Guo et al., 2017) while other metrics such as maximum calibration error (Guo et al., 2017), Brier score (Brier, 1950), and negative log-likelihood (NLL) are used to capture related and complementary notions of miscalibration. Despite their impressive accuracy and trends toward better calibration, modern neural networks still display some persistent miscalibration (Minderer et al., 2021), especially when the task distribution does not match the training distribution (Ovadia et al., 2019). Consequently, a rich toolbox of post hoc fixes has emerged: Platt scaling (Platt, 1999), temperature scaling (Guo et al., 2017), histogram binning (Zadrozny & Elkan, 2001), and other recalibration methods (Kumar et al., 2019; Naeini et al., 2015; Zollo et al., 2024) can be applied to an already trained model to reduce calibration error without altering its decision rule. Other popular methods to improve calibration of neural networks include dropout (Gal & Ghahramani, 2016) and ensembling (Lakshminarayanan et al., 2017).

Uncertainty Quantification in Robotics Uncertainty quantification has long been a central theme in robotics, particularly through the lens of probabilistic robotics (Thrun et al., 2005; Deisenroth & Rasmussen, 2011). Recent work has extended this to modern learning-based robot systems, where uncertainty can guide cautious execution, human assistance, exploration, or online adaptation (Chua et al., 2018; Pathak et al., 2019; Wang et al., 2024; Ataei & Dhiman, 2024). One direction uses conformal prediction to provide distribution-free guarantees in embodied decision-making. For example, Ren et al. (2023) introduce KnowNo, a conformal framework for aligning the uncertainty of LLM-based robot planners to ask for help under ambiguity while maintaining statistical guarantees on task completion. Similarly, conformal methods have been used for clarification-seeking in human-robot interaction (Lidard et al., 2024) and online adaptation in imitation learning (Zhao et al., 2025).

A complementary line of work focuses on detecting or managing failures during policy execution. Xu et al. (2025) propose a runtime method for imitation-learning policies that aims to identify failures using only successful training data, while Gu et al. (2025) introduce SAFE, a failure detector meant to generalize across tasks by leveraging internal VLA representations. In other recent work, Valle et al. (2025) propose several uncertainty and execution-quality metrics for VLA manipulation tasks and evaluate how these metrics correlate with human labels for execution quality. Finally, Yuan et al. (2026) study uncertainty-aware policy

steering, where a verifier decides whether to execute an action, ask a clarifying language question, or request action-level intervention depending on semantic and low-level action uncertainty.

3 Calibration in Vision-Language-Action Models

In this section, we formalize the problem of confidence calibration for vision-language-action models. We define calibration, describe a typical example of how contemporary VLA architectures generate actions and how to extract a corresponding confidence estimate, and introduce the standard calibration metrics used in our empirical study (see Figure 2 for a visual summary of this material). Finally, we present two lightweight remedies for the miscalibration we observe in practice: *prompt ensembles* and *action-wise Platt scaling*.

3.1 Calibration

Let $C \in [0, 1]$ denote the confidence reported by a robot policy and $Y \in \{0, 1\}$ the binary indicator of task success (we will use uppercase for random variables and lowercase for their realizations). A perfectly calibrated predictor (Guo et al., 2017; Minderer et al., 2021; Fisch et al., 2022) satisfies

$$\mathbb{P}(Y = 1 \mid C = c) \;=\; c, \qquad \forall\, c \in [0, 1]. \tag{1}$$

If this condition is met, then for the subset of trials on which the robot reports 80% confidence, we should observe successful completion 80% of the time.

3.2 Vision-Language-Action Models

Vision-Language-Action models take visual data and natural language instructions as input and output robot actions (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2024). They are typically initialized from a visually-conditioned language model (VLM) that, in turn, is initialized from a pretrained language model. This allows VLAs to leverage rich multimodal priors in joining perception and action generation into a single end-to-end pipeline. While some VLAs retain the token-based output paradigm inherited from these base models (Zitkovich et al., 2023; Kim et al., 2025; Lee et al., 2025), others have augmented the architecture with, e.g., flow matching action experts for smooth, high-frequency control (Black et al., 2024; 2025).

At task timestep t, a VLA policy π_θ has access to the observed history $o_t = (v_{\leq t}, l_{\leq t}, a_{<t})$, where v is a visual observation, l is the (possibly fixed) natural language instruction, and a is an action. The policy induces a distribution over the next action $\pi_\theta(a \mid o_t)$. When performing a task, the policy executes the most likely action $a_t^\star = \arg\max_a \pi_\theta(a \mid o_t)$. Given the need for calibration, we must extract a scalar confidence score c_t from the policy. We will interpret c_t as the policy's estimate of the probability that the task will ultimately succeed given the observation history and the chosen action, i.e., $\mathbb{P}(Y = 1 \mid o_t, a_t^\star)$.

Next, we propose a baseline method for extracting c_t for a broad class of token-based VLAs. We focus on token-based VLAs because they allow for a familiar probabilistic interpretation; we leave it to future work to establish baselines for other families. However, our subsequent analysis treats c_t as a black-box number, so the measures and methods introduced also apply naturally to flow-based, diffusion-based, or other controllers (e.g., those that output action chunks) as long as they can emit such a scalar.

3.3 Baseline Confidence Estimation for Token-Based VLAs

Many VLAs represent actions using discrete tokens, in close analogy with the token-based outputs of the multimodal language models upon which they are often built. For example, OpenVLA (Kim et al., 2025) and RT-2 (Zitkovich et al., 2023) represent an action with D discrete tokens, where each token corresponds to one dimension of the robot's action space. For each dimension $d \in \{1, \ldots, D\}$, the policy outputs logits $z_t^{(d)}$ and probabilities $p_t^{(d)} = \mathrm{softmax}\big(z_t^{(d)}\big)$, $p_{t,k}^{(d)} = \pi_\theta\big(A_t^{(d)} = k \mid o_t\big)$ for action tokens $k \in \{1, \ldots, K\}$ in an action vocabulary of size K. At time t, the policy selects the highest-probability token $a_t^{(d)} = \arg\max_k p_{t,k}^{(d)}$ for each action dimension d, and decodes these tokens into an action for execution.

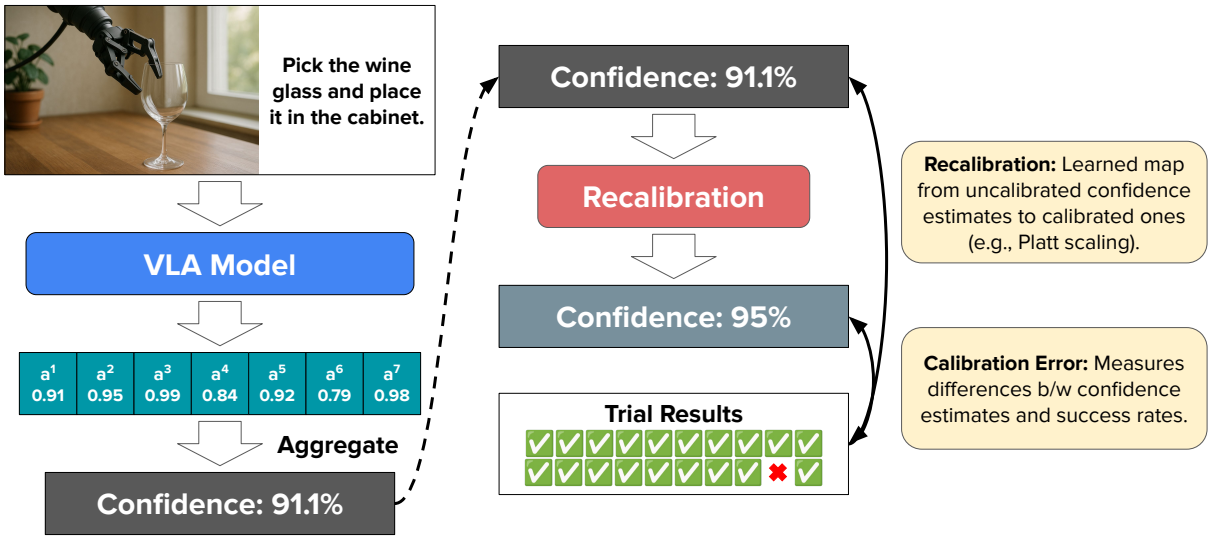

Figure 2: Overview of confidence estimation and calibration for token-based VLAs. Given an input image and text instruction, a token-based VLA generates distributions over discrete action representations, which may correspond to, e.g., action dimensions, latent action codes, or action chunks depending on the model architecture. These distributions can be converted into scalar confidence estimates by aggregating the probability assigned to the selected tokens. Given an uncalibrated confidence estimate, recalibration methods such as Platt scaling use a small calibration dataset to learn a map from uncalibrated confidence estimates to calibrated ones. Calibration error can then be measured by comparing confidence estimates to actual task success rates.

A common heuristic in LLMs is to use the probabilities assigned to generated tokens as a confidence signal for task success, often measured by answer accuracy (Malinin & Gales, 2021; Kadavath et al., 2022). Analogously, we derive a heuristic confidence signal for VLAs using the probability assigned to each selected action-related token. In the dimension-wise case above, this corresponds to using $\max_k p_{t,k}^{(d)}$ for each action dimension d. Given these selected-token probabilities, we can aggregate them into a single scalar confidence estimate c_t, for instance by taking the mean $c_t = \frac{1}{D}\sum_{d=1}^{D} \max_k p_{t,k}^{(d)}$. Other possible aggregation rules over these per-token confidence values include a geometric mean, minimum, maximum, or learned aggregation function. We compare various aggregation rules empirically in Appendix C.1; in our experiments, we use the arithmetic mean.

Not all token-based VLAs expose action probabilities in this exact dimension-wise form. Some models predict multiple discrete tokens per action dimension (Lee et al., 2025), while others predict latent action tokens or tokenized action chunks that are decoded jointly into continuous actions (Bu et al., 2025; Hung et al., 2025). In these cases, the exact form depends on the decoder, but the same principle applies: we use the probabilities assigned to the selected action tokens as the primitive confidence signal.

3.4 Measuring Calibration

Calibration metrics translate deviations from the condition in equation 1 into quantitative measures of *miscalibration*. Given the difficulty of measuring such a condition (i.e., comparing two distributions), we consider a range of metrics. For what follows, let $\{(c_i, y_i)\}_{i=1}^{N}$ denote the reported confidence and binary outcome for each of N robot trials (episodes). Each trial i consists of timesteps $t = 1, \ldots, T_i$ with per-timestep confidences $c_{i,t}$; to obtain a single trial-level value c_i we apply an aggregation function h: $c_i = h(\{c_{i,t}\}_{t=1}^{T_i})$. Possible choices include the confidence before the first action, the mean across timesteps, or the min/max over the trajectory. The following measures are agnostic to choice of h; in our experiments we primarily use the pre-action confidence $c_i = c_{i,1}$, reflecting the high-stakes open-world robotics setting in which early risk assessment is particularly valuable.

One popular measure of miscalibration is **expected calibration error (ECE)** (Guo et al., 2017):

$$\mathrm{ECE}_q \;=\; \big(\mathbb{E}_C\big[|\mathbb{P}(Y=1 \mid C) - C|^q\big]\big)^{1/q}. \tag{2}$$

Put simply, ECE measures the expected difference between confidence and accuracy over the robot's task data distribution. The parameter q is typically set to $q \in \{1, 2\}$. Because we can observe only a finite sample of trial results, the conditional expectation in equation 2 cannot be directly measured. Instead, it is typically approximated with a binning-based estimator. With results from N robot trials, we approximate the population ECE quantity in equation 2 by first ordering predictions according to confidence, and splitting them into M equal-sized bins $B_1, \ldots, B_M$. Then, our empirical estimate $\widehat{\mathrm{ECE}}_q$ is given by $\widehat{\mathrm{ECE}}_q \;=\; \Big(\sum_{m=1}^{M} \frac{|B_m|}{N} \big|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\big|^q\Big)^{1/q}$, where $\mathrm{acc}(B_m)$ and $\mathrm{conf}(B_m)$ are the average accuracy and confidence in the bin.

Brier score (Brier, 1950) is a classic measure of the quality of probabilistic forecasting: $\mathrm{BS} = \frac{1}{N}\sum_{i=1}^{N}(c_i - y_i)^2$. Brier score is an example of a *proper scoring rule* (Gneiting & Raftery, 2007), meaning that it is minimized in expectation when the predicted probabilities match the true conditional success probabilities. Brier score rewards both reliability (confidence matching success) and sharpness (predictions away from the population rate). Another metric used to measure calibration (Guo et al., 2017) is **negative log-likelihood (NLL)**: $\mathrm{NLL} = -\frac{1}{N}\sum_{i=1}^{N}\big[y_i \log c_i + (1 - y_i)\log(1 - c_i)\big]$. Also a proper scoring rule, NLL penalizes very confident failures much more heavily than Brier score.

3.5 Prompt Ensembles

Similar to the VLMs from which they are derived (Zhou et al., 2024), semantically meaningless lexical differences in instructions, e.g., *"pick up the coffee cup"* vs. *"grab the mug"*, might shift visual attention, alter path planning, and thereby change the predictions and confidence scores emitted by VLA models. Although such sensitivity can pose a challenge to reliable task execution, it also creates an opportunity to employ an ensemble-based approach (Lakshminarayanan et al., 2017) to confidence estimation. Specifically, we can treat the particular wording of an instruction as a latent random variable and marginalize over it via Bayesian model averaging. Such averaging over rephrasings reduces variance in the final confidence estimate by canceling out the noise induced by word choice. Implementing this idea as an algorithm, we can then: **(1) Generate rephrasings.** An auxiliary LLM produces r semantically equivalent prompts $\mathcal{L}_{alt} = \{l_{alt}^{(1)}, \ldots, l_{alt}^{(r)}\}$ (see Table 2 for examples). **(2) Estimate confidence with each variant.** Generating an action with $l_{alt}^{(i)}$ yields a confidence $c_t^{(i)}$. **(3) Aggregate.** The final estimate is the ensemble mean $c_t^{ens} = \frac{1}{r}\sum_{i=1}^{r} c_t^{(i)}$. Conceptually, this prompt ensemble technique can play a similar role to full model ensembles (Lakshminarayanan et al., 2017) or inference-time dropout (Gal & Ghahramani, 2016). Also, we stress that this approach is agnostic to the VLA architecture, and can easily be applied to, e.g., a flow-based VLA controller, given the ability to produce c_t. With efficient batching on parallel hardware, this method can add little wall-clock latency in practice.

3.6 Action-Wise Scaling

A standard remedy for miscalibration in classification models is to gather a small validation set from the task distribution and perform post hoc recalibration, learning a function that maps uncalibrated confidence estimates to calibrated ones (Platt, 1999; Naeini et al., 2015; Zadrozny & Elkan, 2001; Guo et al., 2017). Post hoc recalibrators are typically simple functions of the original score function (e.g., logits) or confidence estimate. One popular example of a post hoc recalibration method is Platt scaling (Platt, 1999; Kumar et al., 2019). Given confidence outputs and binary outcomes $\{(c_i, y_i)\}_{i=1}^{N}$ on a held-out validation set, one fits an affine transform $g(c) = \sigma(\alpha c + \beta)$ that minimizes NLL, $\min_{\alpha,\beta} -\sum_i \Big[y_i \log g(c_i) + (1 - y_i)\log\big(1 - g(c_i)\big)\Big]$, where $\sigma(x) = 1/(1 + e^{-x})$. At inference each c_t is replaced with $\tilde{c}_t = g(c_t)$, with the goal of aligning confidence with the probability of task success while leaving the model's actions unchanged.

Unlike classification models that emit one predictive distribution per sample, many VLAs output one distribution per action dimension. Those dimensions can differ dramatically, for example because gripper

open/close appears in nearly 100% of demonstrations, whereas "rotate wrist" is rarer. One global transform therefore may not be able to correct all dimensions simultaneously.

Action-Wise Platt Scaling. We propose to address this heterogeneity by fitting one transform per dimension d:

$$\tilde{c}_t \;=\; \frac{1}{D}\sum_{d=1}^{D} \sigma\big(\alpha_d \max_k p_{t,k}^{(d)} + \beta_d\big), \tag{3}$$

with parameters $\{\alpha_d, \beta_d\}$ learned on the same calibration set. Intuitively, α_d scales dimension d, flattening or sharpening its distribution, while β_d shifts its overall optimism. As in standard Platt scaling, the chosen tokens $a_t^{(d)}$ remain unchanged; we only modify the reported confidence. This dimension-wise perspective can be applied to other post hoc recalibrators as well, for instance action-wise temperature scaling. Beyond the specific methodology, we aim to highlight that VLA calibration will require domain-specific tools rather than a direct transplant of methods designed for standard classifiers.

4 Experiments

Having described the problem of confidence calibration in vision-language-action models, we next conduct an empirical investigation of the following questions: **(1)** Does a higher success rate imply better calibration (Section 4.1)? **(2)** How does calibration evolve over the task time horizon (Section 4.2)? **(3)** Can prompt ensembles consistently improve confidence estimates (Section 4.3)? **(4)** Are action dimensions differentially calibrated, and what are the implications for recalibration (Section 4.4)? Our aim is to highlight key issues and lay the empirical groundwork for future research on calibrating VLAs.

We perform our experiments using 4 different VLA variants: OpenVLA, MolmoAct (Lee et al., 2025), UniVLA (Bu et al., 2025), and NORA (Hung et al., 2025). We choose these models because they represent a variety of VLA design decisions while still all predicting tokens in a manner that allows for a natural probabilistic interpretation.[1] All models are fine-tuned on 4 different task suites from the LIBERO (Liu et al., 2023) benchmark: Spatial, Object, Goal, and 10. LIBERO is a simulation environment for language-conditioned robot manipulation tasks inspired by human activities (see Table 1 for task examples). We deliberately standardize on LIBERO because it is, to our knowledge, the dominant open-source benchmark for modern VLAs and the only one for which a wide range of fine-tuned models are publicly released, enabling comparisons across independently developed VLAs in a common setting; collecting enough real robot data to meaningfully study expected calibration error would be prohibitively expensive (see Limitations Section 6). We also include results for the 8-bit and 4-bit quantized versions of OpenVLA fine-tuned on Spatial, Object, and Goal, for a total of **22 model/task suite combinations**.

Each task suite features 10 different tasks with 50 randomized initializations, for a total of 500 examples. For a given task suite, task success rate represents the proportion of the 500 trials that result in success (while task error rate represents the proportion of trials that result in failure). To calculate ECE, we use 12 equal-mass bins and the Python package released with Kumar et al. (2019).

In Sections 4.1, 4.3, and 4.4, we focus on confidence estimates produced at the first timestep, before any action is taken. This aligns with the need for safety in open-world robot deployments, where robots should signal uncertainty as early as possible in order to avoid costly or dangerous incidents. While alternative heuristics, such as averaging confidence across the trajectory, may also be reasonable, they would often incorporate estimates produced after the robot has already failed or entered an unsafe state. In Section 4.2, we explore how calibration differs across timesteps over the task horizon. To aggregate action-token-level confidence values into a single scalar estimate, we use the arithmetic mean. We compare this aggregation rule with alternatives in Appendix C.1.

4.1 Relationship Between Calibration and Task Success

Much of the early research on calibration in deep learning models was based on perceptions of how task success (usually image classification accuracy) was related to calibration error. In particular, influential work

[1] Descriptions of confidence estimation with MolmoAct, UniVLA, and NORA are found in Appendix B.1.

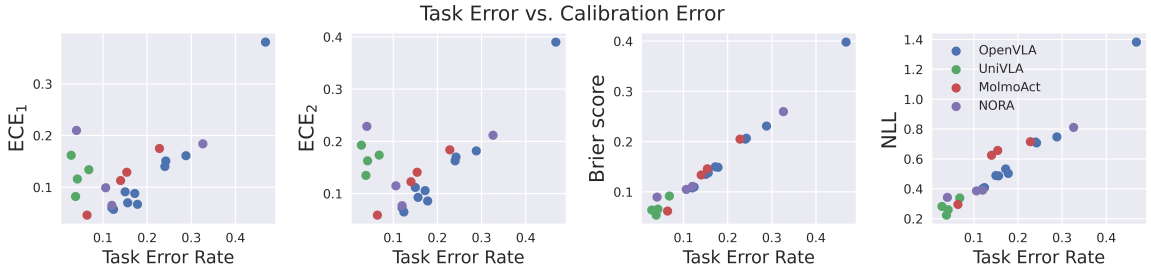

Figure 3: Visualization of task error rates compared against 4 different calibration error measurements for 4 VLA variants (OpenVLA, MolmoAct, UniVLA, and NORA) and 4 LIBERO task suites (Spatial, Object, Goal, 10), as well as OpenVLA 8- and 4-bit versions on Spatial, Object, and Goal. All models exhibit a roughly monotonic relationship between task error and the discriminative measures (Brier score and NLL). ECE shows differences between model families, potentially due to architecture and objective differences.

argued that modern neural networks are poorly calibrated (Guo et al., 2017), suggesting that improved accuracy comes at the expense of calibration. This led researchers to propose training interventions to augment the cross-entropy loss with other objectives seeking improved calibration, potentially at a cost to overall accuracy (Kumar et al., 2018; Mukhoti et al., 2020). However, subsequent research found that, in fact, more accurate networks are generally better calibrated (and easier to recalibrate), and thus modifications to training procedures may not be needed (Minderer et al., 2021). Instead, techniques such as post hoc recalibration (Guo et al., 2017; Zadrozny & Elkan, 2001; Kumar et al., 2019) and ensembling (Lakshminarayanan et al., 2017; Fort & Lakshminarayanan, 2024), applied to models trained for high accuracy, are sufficient to achieve low calibration error (though calibration under domain shift remains a challenge (Ovadia et al., 2019)).

To establish high-level direction for calibration research in VLAs, our first experiment focuses on this important question of how task success relates to pre-execution calibration error (according to ECE_1, ECE_2, Brier score, and NLL) across the 22 model/task suite combinations described above. Confidence estimates are produced using the baseline methods described in Section 3 and Appendix B.1. Results are visualized in Figure 3, and also reported in Table 5. All models exhibit a roughly monotonic relationship between task error and the discriminative metrics (Brier score and NLL), but they differ on ECE: OpenVLA and MolmoAct tend to achieve lower ECE when task error is low, whereas UniVLA and NORA do not show as clean a trend. One possible explanation is that in OpenVLA and MolmoAct, the cross-entropy loss (a proper scoring rule) more directly supervises discretized action dimension tokens, so bin-averaged confidence better tracks per-task success. By contrast, the latent or compressed action representations and auxiliary objectives in UniVLA and NORA may introduce a more significant mismatch between token probabilities and success. These patterns point to architectural complexity in modern VLAs as a potential source of unfavorable calibration behavior. Future work is needed to fully understand which design choices drive this behavior.

4.2 Calibration Over Task Time

The previous experiment evaluates confidence before the first action is executed, a conservative choice for safety-critical deployments. Yet many tasks might allow the robot to collect more information without risk before having to express confidence. For instance, in the wine glass scenario of Figure 1, the gripper can hover above the stem, refine its scene representation, and only then decide whether it is confident enough to proceed. More context should, in principle, yield better calibration.

To test this intuition, we measure calibration across 500 test trials for 100 different levels of task completion ($\{0, 1, \ldots, 99\}\%$), where task completion is calculated as the current timestep index t divided by the total number of timesteps in the task episode (and multiplied by 100). Results for each level of task completion are averaged across the 500 trials, to study whether high-level confidence and calibration trends might occur. In this section, we focus on 6 model/task suite combinations (OpenVLA and its 8-bit version, each applied to Spatial, Object, and Goal suites), to understand how any observations might generalize. Our goal is to

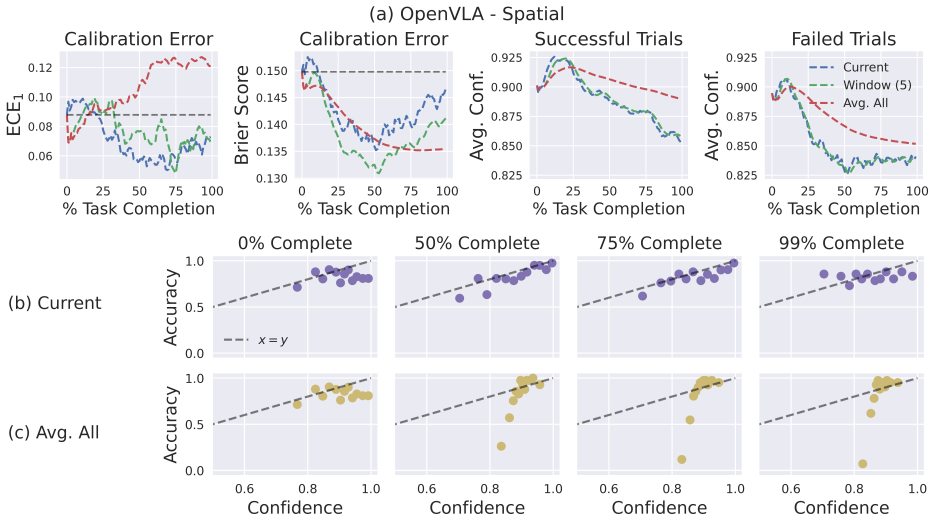

Figure 4: Empirical study of calibration error across the task time. In the top row (a), the left two plots quantify how calibration evolves with task progress, while the right two plots show the average confidence by task time, grouped by successful and failed trials. The bottom rows provide qualitative reliability diagrams for two representative aggregation rules: current-step confidence (b) and averaging all observed confidences so far (c); we omit the sliding-window variant for visual clarity because its behavior is intermediate between these two. Overall, these results illustrate that calibration can improve as the task progresses and more information is gathered, suggesting opportunities for context-aware uncertainty interventions.

examine whether the quality of confidence estimates changes as the robot progresses in its task. At each measured timestep, we compute ECE_1 and Brier score using the baseline method for producing confidence estimates. Because a downstream safety monitor might consider the history of recent estimates (beyond the current one), we evaluate three aggregation rules for reporting a confidence estimate at each timestep: **Current** - confidence from the current timestep only; **Window (5)** - mean over the baseline confidence estimates from the current timestep and the four immediately preceding ones; **Avg. All** - mean over the baseline confidence estimates from all steps seen so far.

Beyond calibration error, we also visualize how average confidence estimates evolve over the task horizon, separated by successful and failed trials. Finally, we plot reliability diagrams for the timesteps corresponding to $\{0, 50, 75, 99\}\%$ completion. Reliability diagrams (Guo et al., 2017) offer a visualization of expected calibration error, using a similar binning strategy. They show confidence against accuracy for each bin, where a well-calibrated system lies on the $x = y$ line. Results for the Spatial task suite with the fine-tuned OpenVLA model are shown in Figure 4. Additional results for Spatial with the Quant-8 model (Figure 10), as well as the Object and Goal suites with the full precision (Figures 11, 13) and Quant-8 models (Figures 12, 14), are presented in Appendix C.5.

Focusing first on results using the confidence estimate from the current step, a clear pattern emerges. As shown in Figure 4 and Figures 10-14, across all 6 settings and both metrics calibration improves sharply from 0% to approximately 50% completion, then plateaus or deteriorates back towards the original level (left-most and left-middle plots in section (a) of the results figures). Beyond lower scores on calibration error, our additional plots characterize the improved probabilistic nature of the confidence estimates towards the middle of the task horizon. In the right-middle and right-most plots in the top section (a) (titled "Successful Trials" and "Failed Trials"), the gap between the average confidence on successful vs. failed trials tends to be greatest around 50% task completion. Additionally, considering the top row of reliability diagrams (section (b)), we can see that the confidence estimates using the current step become far more reliable around this point: error for many bins is large at the beginning, then the difference between confidence and accuracy for most bins becomes smaller until the task is roughly 75% complete. Near task completion, the gap between confidence on successful vs. failed trials closes again, and reliability suffers as a result. These experiments suggest a practical recipe: let the robot execute a certified-safe prefix of the trajectory, and then assess its

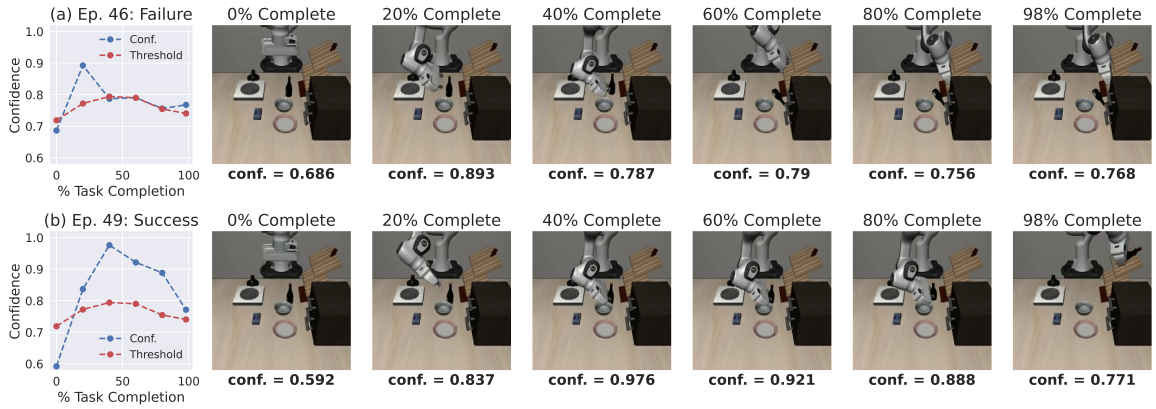

Figure 5: Qualitative examples of a context-aware confidence monitoring strategy applied to a task from the Goal suite. Here, the task is to "put the wine bottle on the rack". The red dashed line represents the 10% quantile of the confidence estimates output by the model across the task time horizon, offering a potential threshold below which the robot may abort execution.

confidence and intervene if necessary. Such horizon-aware monitoring balances safety with the improved reliability in confidence estimates that comes from extra perceptual context.

Next, we observe the effects of other simple strategies for aggregating confidence estimates across timesteps. As described above, alongside the current step confidence, we also consider a sliding window average over 5 single-step estimates and the average of all observed single-step estimates. Across all model/task suite combinations, the sliding window method is particularly effective at improving the Brier score around the middle of the task horizon, although ECE is not always improved. With respect to averaging all single-step estimates so far, we observe that Brier score improves throughout the task, but ECE generally gets worse. The underlying behavior driving these changes can be observed in the bottom row of reliability diagrams (marked (c)): the "Avg. All" method is successful at assigning relatively lower confidence to failed examples, but also reduces variance in the estimates such that they are highly overconfident in these bins (i.e., for the leftmost bin at 99% completion, confidence is still high, but accuracy is near zero). Thus, while this approach fares poorly according to ECE, it is promising in the sense that it enables more effective discrimination between successful and failed trials.

Finally, to probe the generality of these results, we repeat these experiments using the UniVLA model, presented in Appendix Figures 15, 16, and 17. We once again find that calibration improves after making some task progress before deteriorating again, suggesting opportunities for context- and risk-aware applications of confidence quantification.

Qualitative Examples To illustrate how context-aware confidence monitoring could work in practice, we apply it to a representative pick and place task from the Goal suite. The task is to "put the wine bottle on the rack", a case where the robot should be relatively conservative to avoid breaking glass. We consider a naive approach to context-aware monitoring, proposing to halt task performance when both: **(1)** the confidence level falls below a threshold set to the 10% quantile of confidence estimates for that point in the task horizon across all task trials (based on percent completion); **(2)** the robot is within a few inches of contacting an object, or already has contacted an object. Since building a system to detect proximity to objects is beyond the scope of this work, we focus on a qualitative demonstration of this idea. For each example under examination, we plot current confidence for $\{0, 20, 40, 60, 80, 98\}$% completion, as well as the corresponding 10% quantile risk threshold and an image of the robot environment at that time.

Some particularly illustrative examples are shown in Figure 5. In both Episode 46 (marked (a)) and Episode 49 (marked (b)), confidence begins below the threshold. However, given the knowledge that calibration improves throughout the task horizon, we may prefer to allow the robot to proceed with the task as long as it is not too near any objects. Comparing these examples shows the potential of such an approach. In Episode 46, although confidence increases throughout the beginning of the task, it dives sharply after 40%

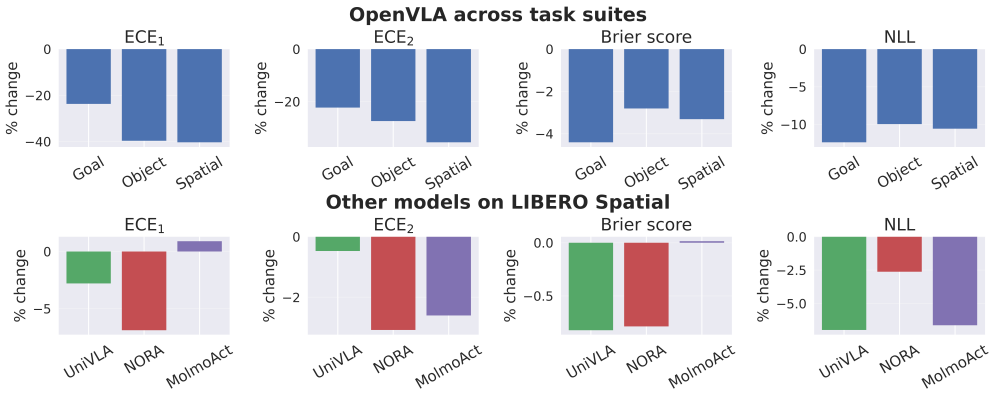

Figure 6: Each bar shows the percent change in a calibration metric after applying Reprompt, relative to the baseline confidence estimate from the original instruction; negative values indicate improved calibration. Results are grouped by OpenVLA task suite and by VLA model on the Spatial suite. ECE_2 and NLL always decrease, while ECE_1 and Brier score decrease in 5 of 6 cases.

task completion and remains low throughout the rest of the task, as the wine bottle is ultimately knocked down and the model is unable to recover. Given our proposed strategy, the task might have been halted before the wine bottle was contacted (at 40% completion), and could, for example, have been deferred to a human. On the other hand, in Episode 49 confidence estimates remain above the threshold for the rest of the trial, and the robot is successful. This episode also highlights the usefulness of an adaptive confidence threshold, given that the confidence estimate at 98% task completion would have fallen below the threshold at other timesteps. More such qualitative examples are provided in Appendix Figure 18. These include both cases where the strategy succeeds and others where it fails (e.g., in Episode 1 confidence falls below the threshold while grasping in an ultimately successful trial).

4.3 Ensembling Confidence Across Prompts

We next study the empirical effectiveness of the prompt ensemble approach described in Section 3. First, for the natural language instruction associated with each task, we create 20 rephrasings using GPT-4o-mini (see Appendix Table 3 for prompts used). During testing, we produce a confidence estimate conditioned on each "reprompt", and average over these confidence scores to obtain the final model confidence. We evaluate our method using OpenVLA across 3 task suites, as well as UniVLA, NORA, and MolmoAct on the Spatial suite. We measure ECE_1, ECE_2, Brier score, and NLL, comparing to the baseline method for producing confidence estimates using the original instruction.

Detailed results (including for quantized OpenVLA) are recorded in Appendix Table 6, while Figure 6 summarizes the impact of Reprompt via percent changes across metrics. Across all model/task suite settings shown, Reprompt always decreases ECE_2 and NLL; ECE_1 and Brier score decrease in 5 of 6 cases, with a small increase for MolmoAct on Spatial. The largest gains occur for OpenVLA across task suites, where ECE reductions can reach roughly 40%. Overall, the improvement is larger for ECE than for the proper scoring rules (Brier score and NLL), suggesting that gains in reliability (i.e., confidence matching marginal success rates) tend to exceed those in sharpness, which is expected from an ensemble technique targeted at reducing variance in confidence estimation. Given its lightweight nature and strong empirical effectiveness, such data augmentation approaches seem promising for enhancing uncertainty quantification in VLAs. To understand the robustness of these results, we perform multiple ablations, confirming that the ensemble approach is robust to different prompts and improves with a larger ensemble; because of space constraints, we defer full details and results to Appendix C.4.

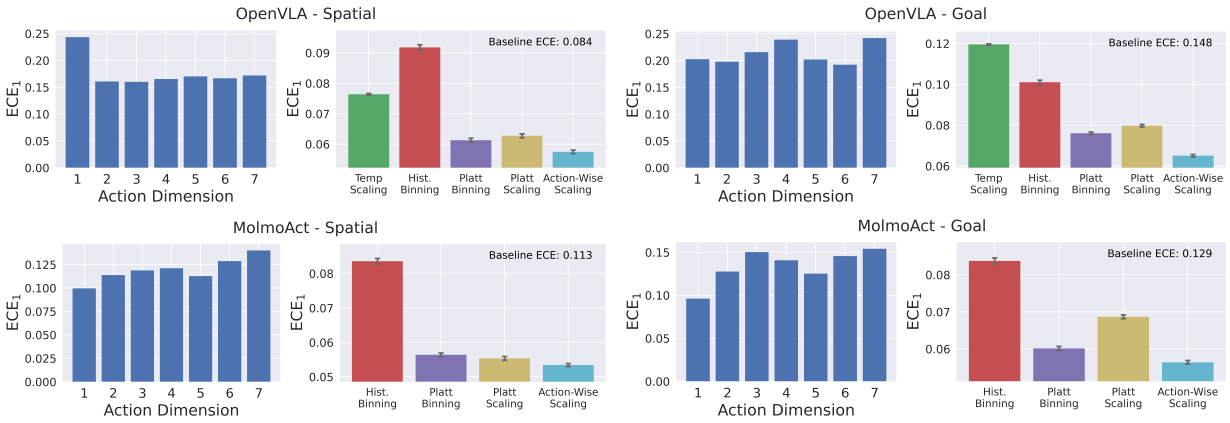

Figure 7: For each model/task setting, the left plot shows ECE for each action dimension using the baseline per-dimension confidence, while the right plot compares recalibration methods (temperature scaling is excluded from MolmoAct results, as the variable-length action encodings preclude a straightforward adaptation). Lower values indicate better calibration in both plots. Calibration varies substantially across action dimensions, and action-wise Platt scaling consistently improves over global recalibration baselines.

4.4 Calibration Across Action Dimensions

Some token-based VLAs such as OpenVLA or MolmoAct decompose a low-level command into tokens explicitly corresponding to their 7 different action dimensions. Our baseline confidence estimate collapses this structure into a single scalar by averaging the top-token probabilities across the dimensions; implicitly, this assumes that every dimension is calibrated to the same degree. That assumption may not hold in practice: a gripper 'open' or 'close' token may appear in nearly every demonstration, whereas a 90° wrist roll could be rare. Such dataset imbalances and other implementation details might skew calibration across dimensions, and relevant algorithms, e.g., for post hoc recalibration, might benefit from explicitly addressing this. To study this question, our final experiment consists of two parts. **(1) Per-dimension calibration audit.** For each dimension d we treat $\max_k p_{t,k}^{(d)}$ as that dimension's confidence and compute ECE_1, to examine whether different degrees of freedom are differentially calibrated. **(2) Targeted recalibration test.** We compare classic Platt scaling to the action-wise variant of Section 3, which learns independent scaling parameters for each action dimension before averaging the transformed confidences across dimensions.

We perform our study using MolmoAct and all 3 OpenVLA variants (full, Quant-8, Quant-4) fine-tuned on the Spatial and Goal task suites, as these models output dimension-wise action predictions. Each experiment is run for 1000 trials, with random 20%/80% calibration-test splits. Given the reduced size of the test set, we measure calibration with 10 equal-mass bins (instead of 12 as in other experiments). Beyond traditional Platt scaling, we also include a range of baselines: temperature scaling (Guo et al., 2017), histogram binning (Zadrozny & Elkan, 2001), and Platt binning (Kumar et al., 2019). (Note: temperature scaling is excluded from MolmoAct results, as the variable-length action encodings preclude a straightforward adaptation.) Figure 7 shows results with the baseline estimates for the full precision OpenVLA and MolmoAct models; Appendix Figure 19 includes the results with the quantized models.

Two key observations arise. First, ECE_1 varies by up to 2 times across dimensions, with no consistent best or worst dimensions across settings. It follows that a single scalar confidence masks significant differences in dimension-wise confidence estimates. Second, we can observe that replacing global Platt scaling with per-action-dimension transforms consistently lowers calibration error (and also improves over the additional baselines). ECE is improved by over 20% in some cases, without changing the selected tokens or adding meaningful runtime overhead. Together, these results indicate that VLA calibration research should treat each degree of freedom as its own concern in order to improve the effectiveness of relevant algorithmic tools. Dimension-aware post hoc methods such as action-wise Platt scaling offer a simple yet effective path toward that goal. More broadly, our findings underscore that calibrating VLAs introduces challenges not

seen in other domains, so meaningful progress will demand substantial domain-specific research rather than wholesale adoption of understanding and techniques from other areas of deep learning.

5 Discussion

Our results suggest that VLAs can already provide informative confidence signals, even when confidence is extracted using simple token-probability baselines. In several settings, these estimates achieve relatively low calibration error, suggesting that the probabilistic structure of current VLA policies is already useful for estimating task success. However, calibration varies across architectures, task suites, timesteps, and action dimensions, so low aggregate ECE in one setting should not be interpreted as a general guarantee of calibration. Rather, these findings motivate evaluating VLA confidence as a property of the full model-task-execution pipeline rather than as a fixed property of the policy alone.

The calibration tools we propose provide initial evidence that this structure can be exploited to improve calibration. Prompt ensembling reduces calibration error by averaging over semantically equivalent instructions, suggesting that some variation in confidence arises from lexical sensitivity rather than task-relevant uncertainty. Action-wise Platt scaling improves over global recalibration by accounting for heterogeneity across control dimensions, while the temporal analysis shows that confidence estimates may become more reliable after partial task progress. Together, these results suggest that effective VLA calibration will benefit from methods that account for the linguistic, temporal, and action-structured nature of embodied decision-making.

6 Limitations

All experiments in this work are conducted in simulation. This choice is not uncommon in related research (Mees et al., 2022; Jiang et al., 2023; Li et al., 2024a;b), and enables the controlled resets and hundreds of rollouts per task suite needed for stable calibration estimates. Replicating the same study on physical robot hardware would require thousands of real executions, which is often prohibitively time-consuming and expensive at the scale needed for reliable ECE measurements. As a result, simulation provides the most practical testbed for initiating calibration analysis in VLAs, even though it cannot fully capture sensor noise, latency, and other physical factors present in the real world. We additionally focus on token-based VLAs, since their discrete predictions admit a natural probabilistic interpretation and enable confidence estimation directly from model likelihoods. Given the novelty of studying calibration in VLAs, these design choices let us establish a reasonable baseline across multiple models, but leave open questions of how these results translate to real settings and other architectures. We view expanding to real-robot evaluations, broader environments, and a wider range of VLA architectures as essential future work for building a more complete picture of VLA calibration.

7 Future Work

Several extensions follow naturally from our work. Confidence calibration should be benchmarked across other environments, robot embodiments, and VLA architectures, including diffusion-based planners, continuous regression heads, and flow-matching controllers. Establishing strong baselines for these architectures will likely require new confidence surrogates and post hoc adjustments. Likewise, perturbation techniques besides instruction paraphrasing, e.g., inserting synthetic lighting changes or random distractors, could expose additional failure modes and inspire new multimodal ensemble techniques. Running such experiments on real robots will be essential for understanding how various physical factors affect VLA calibration in the real world. Uncertainty estimation also holds the potential to guide efficient data gathering and fine-tuning, as in active learning (Wang et al., 2017). Further, while all of our experiments focused on fine-tuned in-distribution settings, future work should consider these phenomena in the zero-shot and out of distribution settings. Finally, our temporal analysis hints at a period mid-trajectory where confidence is most trustworthy; integrating that signal into selective execution, planning, or human-in-the-loop systems is an open challenge for designing adaptive risk-aware robotic pipelines.

References

Masoud Ataei and Vikas Dhiman. Dadee: Well-calibrated uncertainty quantification in neural networks for barriers-based robot safety. *arXiv preprint arXiv:2407.00616*, 2024.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of long-form generations. In *International Conference on Machine Learning*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *CoRR*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1): 1–3, 1950.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, 2024.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 2018.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, 2011.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Adam Fisch, Tommi S. Jaakkola, and Regina Barzilay. Calibrated selective classification. *Transactions on Machine Learning Research*, 2022.

Stanislav Fort and Balaji Lakshminarayanan. Ensemble everything everywhere: Multi-scale aggregation for adversarial robustness. *arXiv preprint arXiv:2408.05446*, 2024.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. Safe: Multitask failure detection for vision-language-action models. *arXiv preprint arXiv:2506.09937*, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U Tan, Navonil Majumder, Soujanya Poria, et al. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv preprint arXiv:2504.19854*, 2025.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts, 2023. URL https://arxiv.org/abs/2210.03094.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.

Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. *Advances in Neural Information Processing Systems*, 2019.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 2017.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

Jiachen Li, Qiaozi Gao, Michael Johnston, Xiaofeng Gao, Xuehai He, Suhaila Shakiah, Hangjie Shi, Reza Ghanadan, and William Yang Wang. Mastering robot manipulation with multimodal prompts through pretraining and multi-task fine-tuning, 2024a. URL https://arxiv.org/abs/2310.09676.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators, 2024b. URL https://arxiv.org/abs/2311.01378.

Justin Lidard, Hang Pham, Ariel Bachman, Bryan Boateng, and Anirudha Majumdar. Risk-calibrated human-robot interaction via set-valued intent prediction. *arXiv preprint arXiv:2403.15959*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 2023.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction, 2021.

Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data, 2022. URL https://arxiv.org/abs/2204.06252.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 2021.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 2020.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI Conference on Artificial Intelligence*, 2015.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 2019.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, 2019.

J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3), 1999.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Pen Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning (CoRL)*, 2023.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *The Conference on Empirical Methods in Natural Language Processing*, 2023.

Pablo Valle, Chengjie Lu, Shaukat Ali, and Aitor Arrieta. Evaluating uncertainty and quality of visual language action-enabled robots, 2025. URL `https://arxiv.org/abs/2507.17049`.

Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, December 2017.

Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. Coplanner: Plan to roll out conservatively but to explore optimistically for model-based rl. In *International Conference on Learning Representations*, 2024.

Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Ambrus, Haruki Nishimura, and Masha Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies, 2025. URL `https://arxiv.org/abs/2503.08558`.

Jessie Yuan, Yilin Wu, and Andrea Bajcsy. When to act, ask, or learn: Uncertainty-aware policy steering, 2026. URL `https://arxiv.org/abs/2602.22474`.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. *International Conference on Machine Learning*, 2001.

Michelle Zhao, Reid Simmons, Henny Admoni, Aaditya Ramdas, and Andrea Bajcsy. Conformalized interactive imitation learning: Handling expert shift and intermittent feedback, 2025. URL `https://arxiv.org/abs/2410.08852`.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. In *International Conference on Learning Representations*, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023.

Thomas P Zollo, Zhun Deng, Jake Snell, Toniann Pitassi, and Richard Zemel. Improving predictor reliability with selective recalibration. In *Transactions on Machine Learning Research*, 2024.

A Additional Related Work

Calibration in LLMs Given that most state-of-the-art VLAs are built on LLM backbones, it is natural to consider how our work relates to existing studies of calibration in LLMs, an area that has received considerable attention recently. Typical approaches to generating confidence estimates with LLMs include expressing calibration as a multiple-choice question for the LLM (Kadavath et al., 2022), directly verbalizing confidence with the model's text output (Lin et al., 2022; Tian et al., 2023; Band et al., 2024), or measuring semantic consistency across many sampled outputs for the same query (Kuhn et al., 2023; Duan et al., 2024; Chen et al., 2024). However, it is difficult to adapt these methods to the VLA setting. For example, while LLMs may produce the entire sequence before measuring confidence, in robotics potential failures must be flagged much earlier in the trajectory to ensure safety and avoid costly accidents. Also, sampling multiple full trajectories in the physical world may be impossible. Finally, current VLAs lack the flexible and robust text-to-text interface of LLMs, and thus cannot be expected to, e.g., answer natural language questions about their confidence in an action prediction.

B Additional Experiment Details

This section contains additional experiment details. All VLA models and benchmarks used in our experiments are open source and publicly available. Our code is included with the submission as supplementary material, and will be released upon publication. Task examples from the LIBERO task suites are shown in Table 1.

B.1 Confidence Estimation with VLA Variants

MolmoAct At each timestep, MolmoAct predicts a sequence of discrete action tokens for every action dimension, with a softmax distribution over the 256 action bins for each token. For a given action dimension, we take the probability assigned to the executed bin for each of its tokens and average these probabilities to obtain a per-dimension confidence. This yields one scalar confidence value per action dimension. We then average these per-dimension confidences across all action dimensions to obtain a single scalar confidence for the action at that timestep.

UniVLA At each timestep, UniVLA predicts a fixed number (4) of discrete latent action tokens, each chosen from a codebook of size 16. For each latent token, we take the softmax probability assigned to the selected code as its token-level confidence. Because these latent tokens are jointly decoded into all continuous control dimensions, the confidence scores do not correspond to particular action dimensions. Finally, we average these per-token confidences to obtain a single scalar confidence for UniVLA at that timestep.

NORA At each timestep, NORA predicts an action chunk encoded as a sequence of discrete tokens, each selected via a softmax over the augmented action vocabulary. For every token, we take the softmax probability of the sampled token as its token-level confidence. Because the NORA tokenizer mixes information from all control dimensions, we do not derive per-dimension confidences. Instead, we compute a single scalar confidence for NORA by averaging these token-level confidences across all tokens in the predicted action chunk for that timestep.

B.2 Rephrasing Prompts

The prompts given to GPT-4o-mini for rephrasing instructions from the LIBERO robot simulation environment are listed in Table 3. Prompt 1 is the main prompt used throughout the experiments, and Prompts 2 and 3 are used to ablate the sensitivity to the rephrasings. Note that the different prompts lead to substantially different rephrasings, as Prompts 2 and 3 mandate the retention of the words "pick" and "place", while as Table 2 shows, Prompt 1 leads to these words being replaced.

Table 1: Task examples from the LIBERO task suites: Spatial, Object, Goal, and 10.

Dataset	Example Image	Example Instruction
Spatial		Pick up the black bowl between the plate and the ramekin and place it on the plate.
Object		Pick up the alphabet soup and place it in the basket.
Goal		Open the middle drawer of the cabinet.
10		Put both the alphabet soup and the tomato sauce in the basket.

Table 2: Example of multiple paraphrases of a single VLA instruction (from our experiments with LIBERO Spatial). The rephrasings carry the same meaning as the original instruction, enabling a *reprompting* approach in which we ensemble over predictions conditioned on lexically different yet semantically equivalent instructions.

Instruction	Pick up the black bowl between the plate and the ramekin and place it on the plate
Rephrasings	(1) Lift the black bowl located between the plate and the ramekin and set it on the plate. (2) Grasp the black bowl found between the plate and the ramekin and move it to the plate. (3) Take the black bowl positioned between the plate and the ramekin and position it on the plate.

Table 3: Prompts for GPT-4o-mini to rephrase LIBERO task instructions. These prompts are used for Spatial and Object. There is a small modification for Goal.

	Prompt
Prompt 1	You are generating alternative phrasings of a robotic task instruction while preserving its exact meaning. ### Task Instruction: '[TASK DESCRIPTION]' ### Instructions: - Generate **20** alternative ways to phrase the task instruction. - Keep each instruction **concise and unambiguous**. - Ensure the instructions remain suitable for a **robot, not a human**. - Only make **semantically meaningless** changes (e.g., word order, synonyms, slight rewording). - Double-check that the new instructions mean the same exact thing for the robot; do not just substitute synonyms without considering context. - Do **not** introduce additional steps, remove essential details, or alter the action. ### Output Format: Each rephrased instruction should be wrapped in '[instruction]' and '[/instruction]' tags, like this: [instruction] Rephrased instruction 1 [/instruction] [instruction] Rephrased instruction 2 [/instruction]
Prompt 2	You are generating alternative phrasings of a robotic task instruction while preserving its exact meaning. ### Task Instruction: '[TASK DESCRIPTION]' ### Instructions: - Generate **20** alternative ways to phrase the task instruction. - Keep each instruction **concise and unambiguous**. - Ensure the instructions remain suitable for a **robot, not a human**. - Only make **semantically meaningless** changes (e.g., word order, synonyms, slight rewording). - Double-check that the new instructions mean the same exact thing for the robot; do not just substitute synonyms without considering context. - Do **not** introduce additional steps, remove essential details, or alter the action. - The first word of the instruction should be 'PICK', and then it should also include the word 'PLACE'. ### Output Format: Each rephrased instruction should be wrapped in '[instruction]' and '[/instruction]' tags, like this: [instruction] Rephrased instruction 1 [/instruction] [instruction] Rephrased instruction 2 [/instruction]
Prompt 3	You are generating alternative phrasings of a robotic task instruction while preserving its exact meaning. ### Task Instruction: '[TASK DESCRIPTION]' ### Instructions: - Generate **20** alternative ways to phrase the task instruction. - Keep each instruction **concise and unambiguous**. - Ensure the instructions remain suitable for a **robot, not a human**. - Only make **semantically meaningless** changes (e.g., word order, synonyms, slight rewording). - Double-check that the new instructions mean the same exact thing for the robot; do not just substitute synonyms without considering context. - Do **not** introduce additional steps, remove essential details, or alter the action. - Make the changes as minor as possible, as the robot's language system is not very robust to rephrasing. - The first word of the instruction should be 'PICK', and then it should also include the word 'PLACE'. ### Output Format: Each rephrased instruction should be wrapped in '[instruction]' and '[/instruction]' tags, like this: [instruction] Rephrased instruction 1 [/instruction] [instruction] Rephrased instruction 2 [/instruction]

Table 4: Comparison of different strategies for aggregating confidence across the set of tokens produced by the VLA models. Results are presented (1) per model, averaged across task suites and (2) averaged across all models and task suites. Using the arithmetic mean performs best on average, but the geometric mean offers comparable performance.

Model	Method	ECE_1	ECE_2	Brier score	NLL
NORA	Arithmetic Mean	0.140	0.158	0.142	0.483
	Geometric Mean	0.162	0.185	0.150	0.501
	Max	0.148	0.152	0.148	2.006
	Min	0.442	0.468	0.354	0.936
MolmoAct	Arithmetic Mean	0.116	0.127	0.137	0.573
	Geometric Mean	0.113	0.124	0.136	0.568
	Max	0.147	0.156	0.147	2.000
	Min	0.083	0.097	0.128	0.460
UniVLA	Arithmetic Mean	0.123	0.166	0.069	0.276
	Geometric Mean	0.144	0.189	0.078	0.298
	Max	0.049	0.059	0.046	0.589
	Min	0.336	0.383	0.192	0.564
OpenVLA	Arithmetic Mean	0.170	0.185	0.216	0.756
	Geometric Mean	0.160	0.175	0.211	0.732
	Max	0.251	0.251	0.251	3.446
	Min	0.220	0.266	0.239	0.699
Average	Arithmetic Mean	0.137	0.159	0.141	0.522
	Geometric Mean	0.145	0.168	0.144	0.525
	Max	0.148	0.155	0.148	2.010
	Min	0.270	0.303	0.228	0.665

C Additional Experiment Results

All task success and calibration results for 22 VLA/task suite combinations are shown in Table 5.

C.1 Comparison of Aggregation Methods

We use the arithmetic mean of selected action-token probabilities as the default confidence score in our experiments. In Table 4, we compare this strategy to taking the geometric mean, maximum, or minimum of the same set of token probabilities. While the geometric mean is often used as a length-normalized confidence score, we find it to be slightly less effective for VLA calibration on average across our testbed. This may reflect the fact that downstream task success is not equivalent to the exact joint likelihood of the tokenized action sequence: action dimensions can differ in entropy, calibration, and task/timestep relevance, and isolated low-probability tokens may not indicate genuine execution risk.

C.2 Relationship Between Task Success and Calibration

Table 5 features all results plotted in Figure 3.

C.3 Prompt Ensembling

Table 6 has prompt ensemble results for all models.

Table 5: All task success and calibration results for 22 VLA/task suite combinations.

Dataset	Model	Quant	ECE_1	ECE_2	Brier	NLL	Succ. Rate
Spatial	NORA	-	0.099	0.115	0.105	0.386	0.894
Object	NORA	-	0.210	0.229	0.090	0.343	0.960
Goal	NORA	-	0.065	0.077	0.111	0.391	0.880
10	NORA	-	0.184	0.212	0.260	0.811	0.674
Spatial	MolmoAct	-	0.113	0.123	0.134	0.625	0.860
Object	MolmoAct	-	0.046	0.059	0.062	0.296	0.936
Goal	MolmoAct	-	0.129	0.141	0.146	0.657	0.846
10	MolmoAct	-	0.175	0.184	0.205	0.716	0.772
Spatial	UniVLA	-	0.162	0.193	0.064	0.282	0.972
Object	UniVLA	-	0.082	0.135	0.054	0.224	0.962
Goal	UniVLA	-	0.116	0.163	0.066	0.261	0.958
10	UniVLA	-	0.134	0.174	0.092	0.339	0.932
Spatial	OpenVLA	-	0.088	0.106	0.150	0.533	0.828
Object	OpenVLA	-	0.060	0.073	0.108	0.401	0.880
Goal	OpenVLA	-	0.151	0.170	0.207	0.707	0.758
10	OpenVLA	-	0.381	0.390	0.398	1.382	0.532
Spatial	OpenVLA	Quant-8	0.070	0.093	0.138	0.488	0.844
Object	OpenVLA	Quant-8	0.057	0.065	0.110	0.409	0.876
Goal	OpenVLA	Quant-8	0.140	0.163	0.205	0.713	0.760
Spatial	OpenVLA	Quant-4	0.067	0.086	0.149	0.503	0.822
Object	OpenVLA	Quant-4	0.091	0.112	0.135	0.489	0.850
Goal	OpenVLA	Quant-4	0.161	0.182	0.231	0.748	0.712

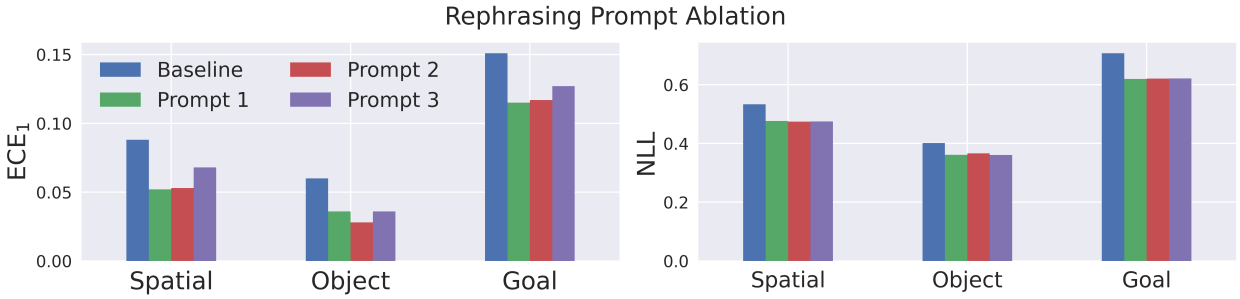

Figure 8: Ablation results for the prompt ensemble method, where different prompts are given to GPT-4o-mini for producing the rephrasings of the original instruction. Prompt 1 is used in the main experiments, while Prompts 2 and 3 are variants. Results show that improvements in calibration error from Reprompt are robust to different rephrasing prompts.

C.4 Ablations for Prompt Ensembles

To understand the robustness of our prompt ensemble results, we perform multiple ablations. First, we consider the effect of changing the prompt given to GPT-4o-mini for producing the 20 instruction rephrasings. In addition to the prompt used to produce the results in Table 6, we try two additional prompts, recorded in Appendix Table 3; "Prompt 1" is the original prompt, and "Prompt 2" and "Prompt 3" represent new prompts for the ablation. Note that the different prompts lead to substantially different rephrasings, as Prompts 2 and 3 mandate the retention of the words "pick" and "place", while Table 2 shows that Prompt 1 leads to these words being replaced. The ablation is run across all 3 task suites using fine-tuned OpenVLA,

Table 6: Calibration error measurements using all 4 models (and the OpenVLA 8-bit and 4-bit fine-tuned model versions) and 3 different LIBERO task suites, for 2 different methods of confidence estimation: (1) baseline (2) ensembling over semantically equivalent prompts ("Reprompt"). The prompt ensemble method improves most measures.

Model	Dataset	Method	ECE_1	ECE_2	Brier score	NLL
OpenVLA	Spatial	Baseline	0.0879	0.1062	0.1498	0.5330
		Reprompt	0.0525	0.0685	0.1449	0.4766
	Object	Baseline	0.0602	0.0727	0.1076	0.4010
		Reprompt	0.0363	0.0527	0.1046	0.3610
	Goal	Baseline	0.1513	0.1701	0.2065	0.7072
		Reprompt	0.1155	0.1322	0.1974	0.6195
OpenVLA (Quant 8)	Spatial	Baseline	0.0704	0.0934	0.1378	0.4876
		Reprompt	0.0498	0.0616	0.1311	0.4342
	Object	Baseline	0.0574	0.0651	0.1096	0.4089
		Reprompt	0.0414	0.0522	0.1081	0.3745
	Goal	Baseline	0.1396	0.1627	0.2052	0.7134
		Reprompt	0.1170	0.1520	0.1938	0.6081
OpenVLA (Quant 4)	Spatial	Baseline	0.0673	0.0861	0.1493	0.5029
		Reprompt	0.0546	0.0790	0.1482	0.4825
	Object	Baseline	0.0911	0.1125	0.1354	0.4889
		Reprompt	0.0663	0.0942	0.1318	0.4540
	Goal	Baseline	0.1615	0.1821	0.2307	0.7475
		Reprompt	0.1456	0.1535	0.2239	0.6720
NORA	Spatial	Baseline	0.0986	0.1153	0.1054	0.3859
		Reprompt	0.0918	0.1117	0.1046	0.3758
UniVLA	Spatial	Baseline	0.1618	0.1933	0.0644	0.2821
		Reprompt	0.1573	0.1924	0.0638	0.2624
MolmoAct	Spatial	Baseline	0.1126	0.1231	0.1337	0.6248
		Reprompt	0.1137	0.1199	0.1338	0.5834

and we measure ECE_1 and NLL. Results are shown in Figure 8. Improvements in calibration error from the Reprompt method are robust to different rephrasing prompts, with all ensemble variants producing lower error than the baseline method across both metrics and all 3 task suites.

Second, we consider the effect of the number of prompts used in the prompt ensemble. Given a sound ensembling approach, we would expect to see improvement as more prompts are added to the ensemble (up to some point). To study this question, we record results using $k \in \{1, 5, 10, 20\}$ different rephrasings (randomly chosen for 1000 trials when $k < 20$). Results for the Spatial, Goal and Object task suites are in Figure 9. We see that the algorithm behaves favorably, where calibration error generally decreases as more instructions are included in the ensemble.

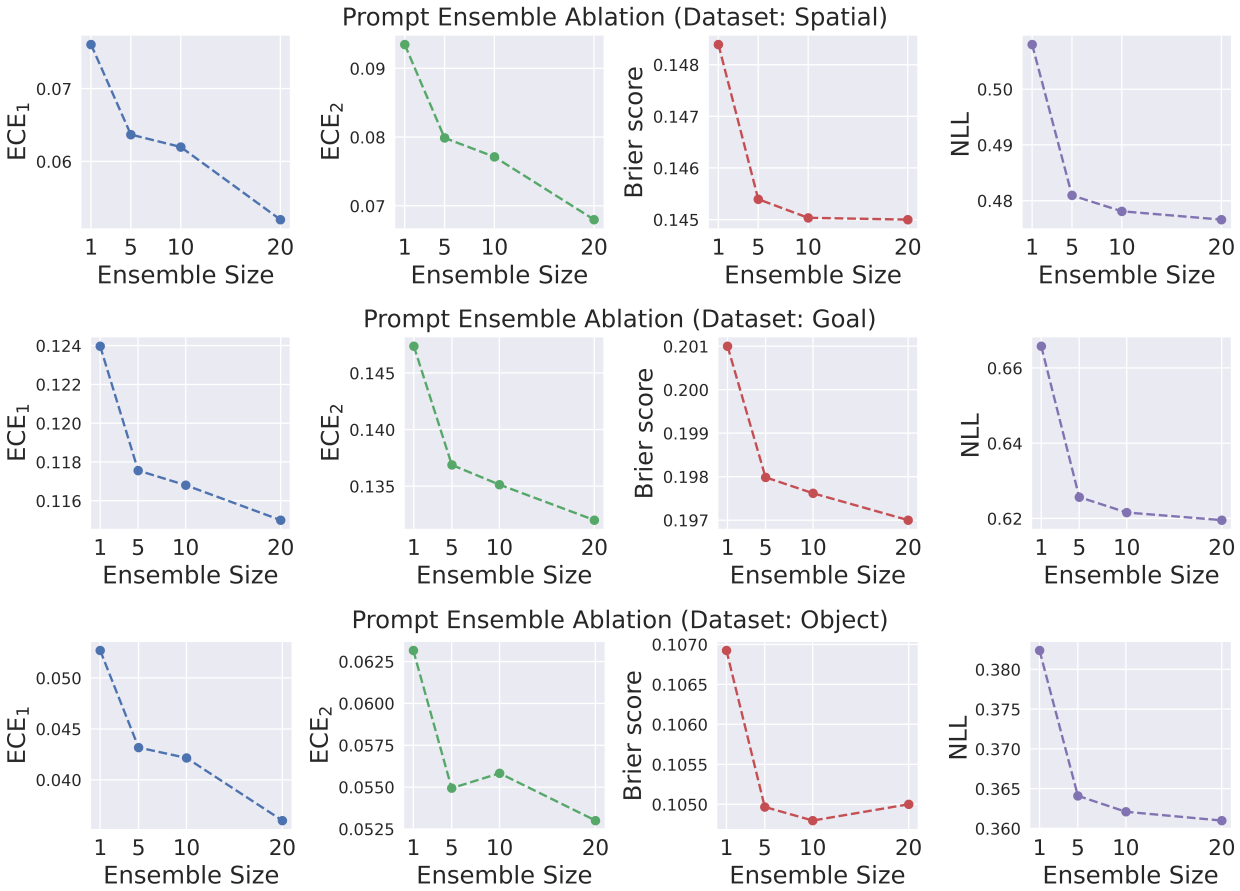

Figure 9: Ablating the number of prompts in the prompt ensemble for the Spatial, Goal, and Object task suites. Increasing the number of prompts generally improves calibration error.

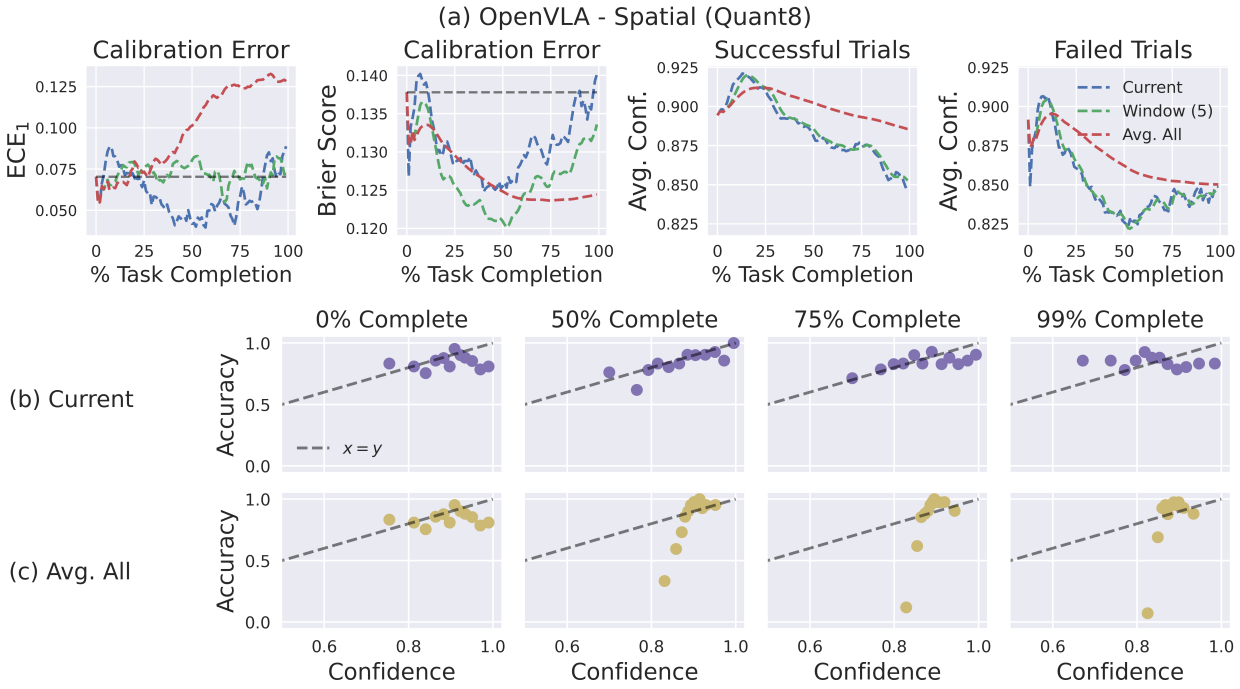

Figure 10: Empirical study of calibration error across task time horizon for the Spatial task suite and the Quant-8 OpenVLA model.

C.5 Calibration Across Task Time

Figures 10, 11, 12, 13, and 14 offer additional results on calibration across the task time horizon using the OpenVLA model. Figures 15, 16, and 17 offer results for the same experiment on UniVLA. Figure 18 shows further qualitative examples from the context-aware monitoring experiment.

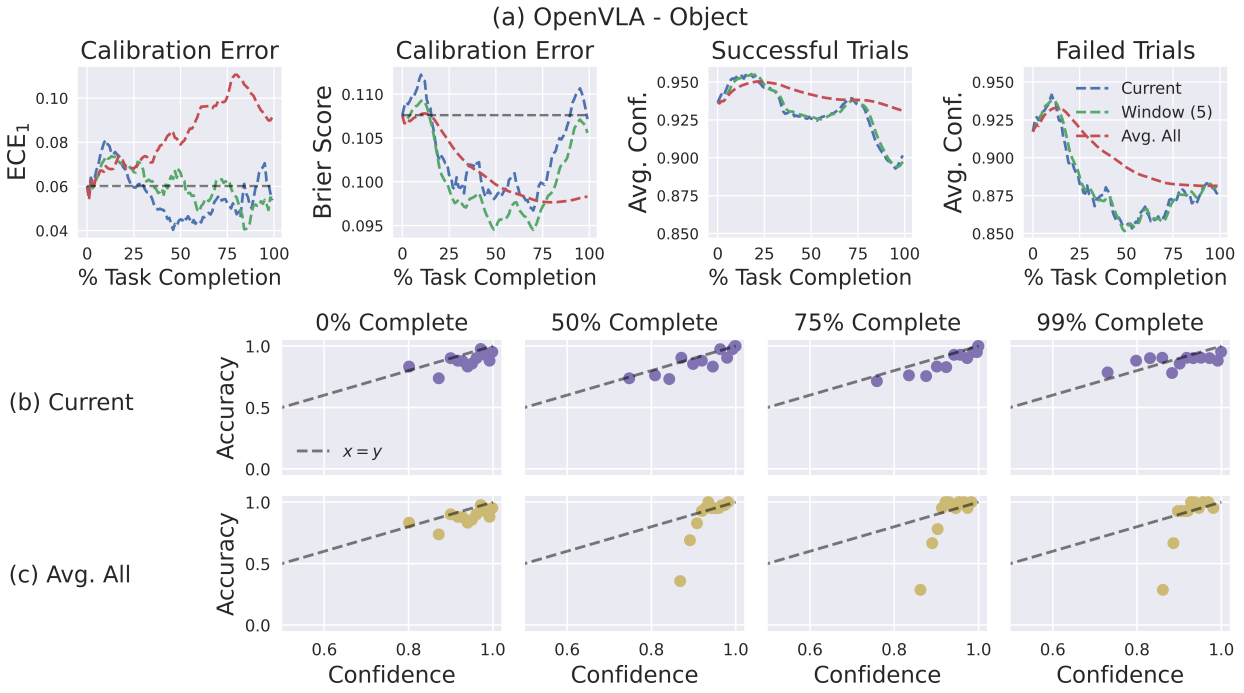

Figure 11: Empirical study of calibration error across task time horizon for the Object task suite and the full precision OpenVLA model.

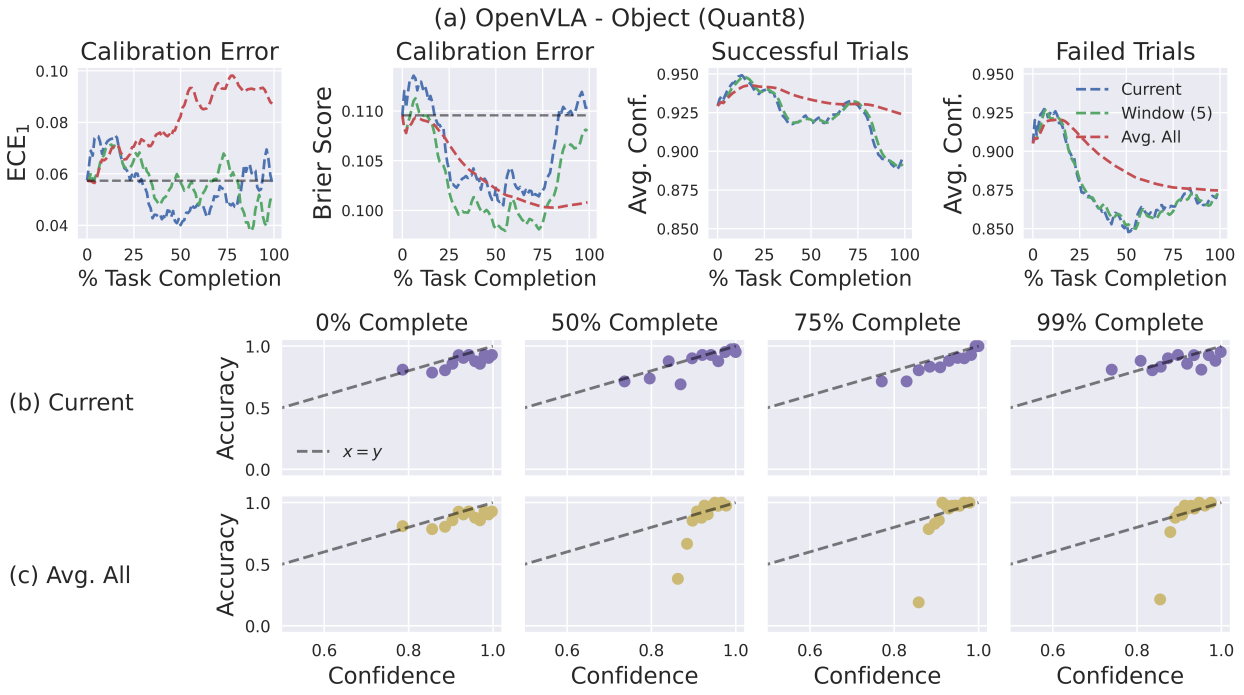

Figure 12: Empirical study of calibration error across task time horizon for the Object task suite and the Quant-8 OpenVLA model.

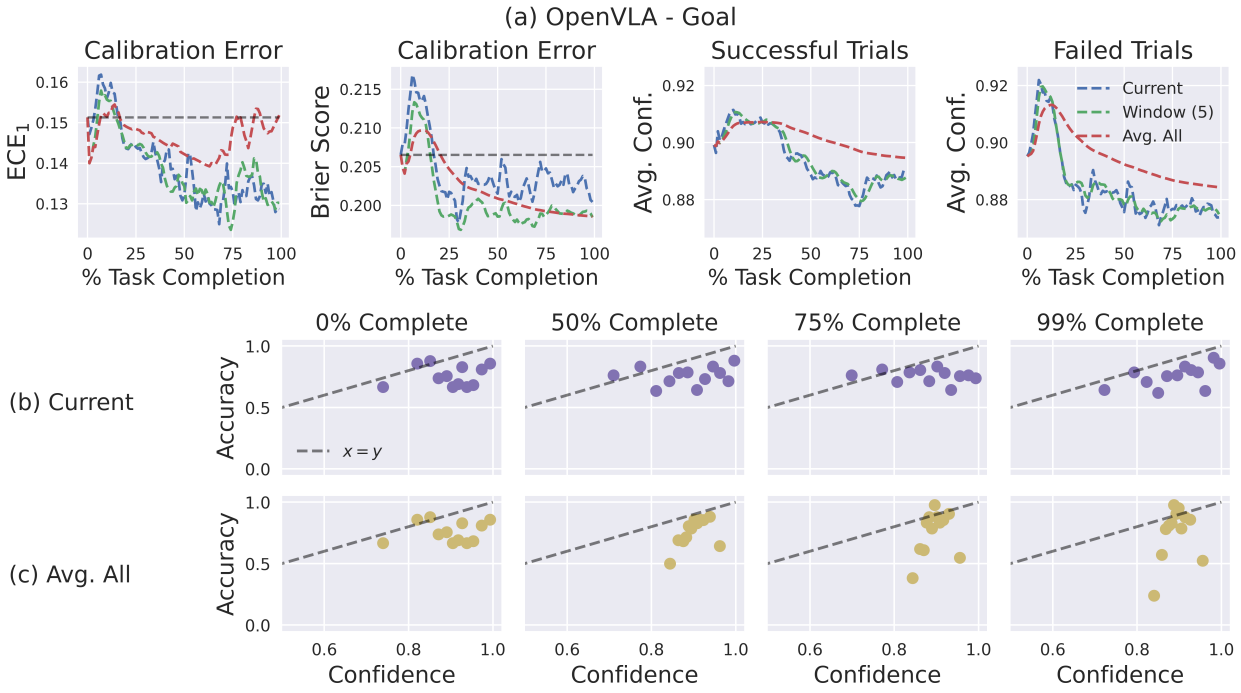

Figure 13: Empirical study of calibration error across task time horizon for the Goal task suite and the full precision OpenVLA model.

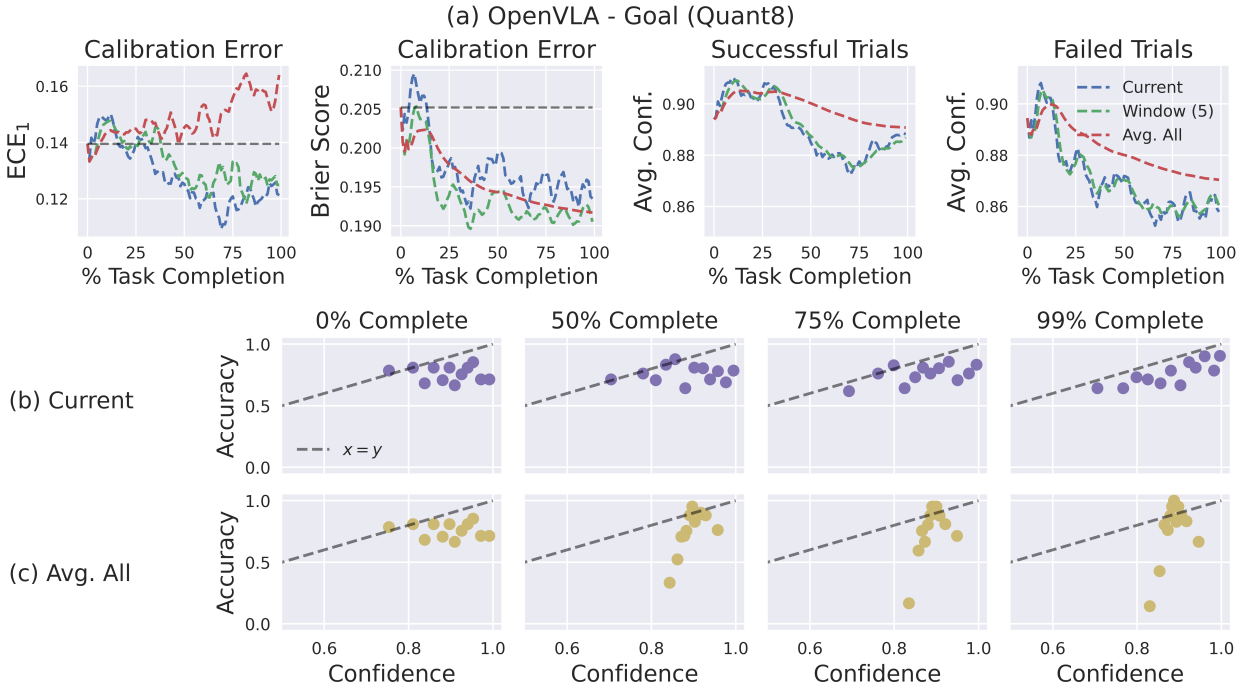

Figure 14: Empirical study of calibration error across task time horizon for the Goal task suite and the Quant-8 OpenVLA model.

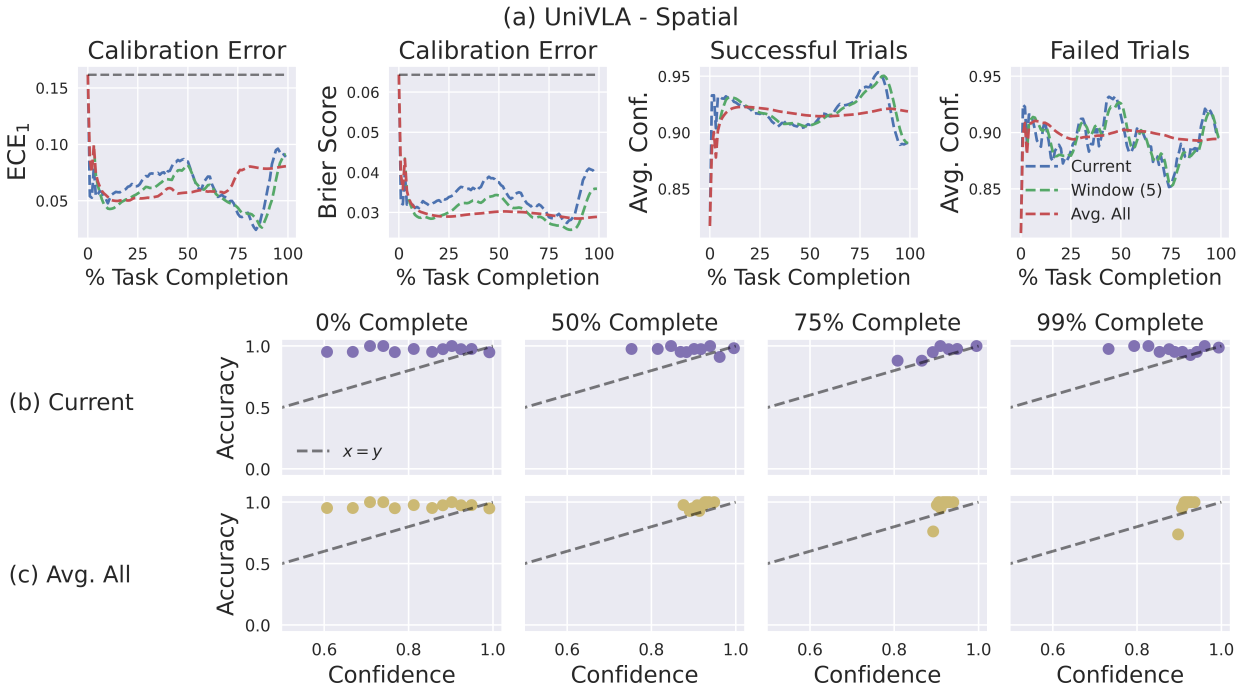

Figure 15: Empirical study of calibration error across task time horizon for the Spatial task suite and the UniVLA model.

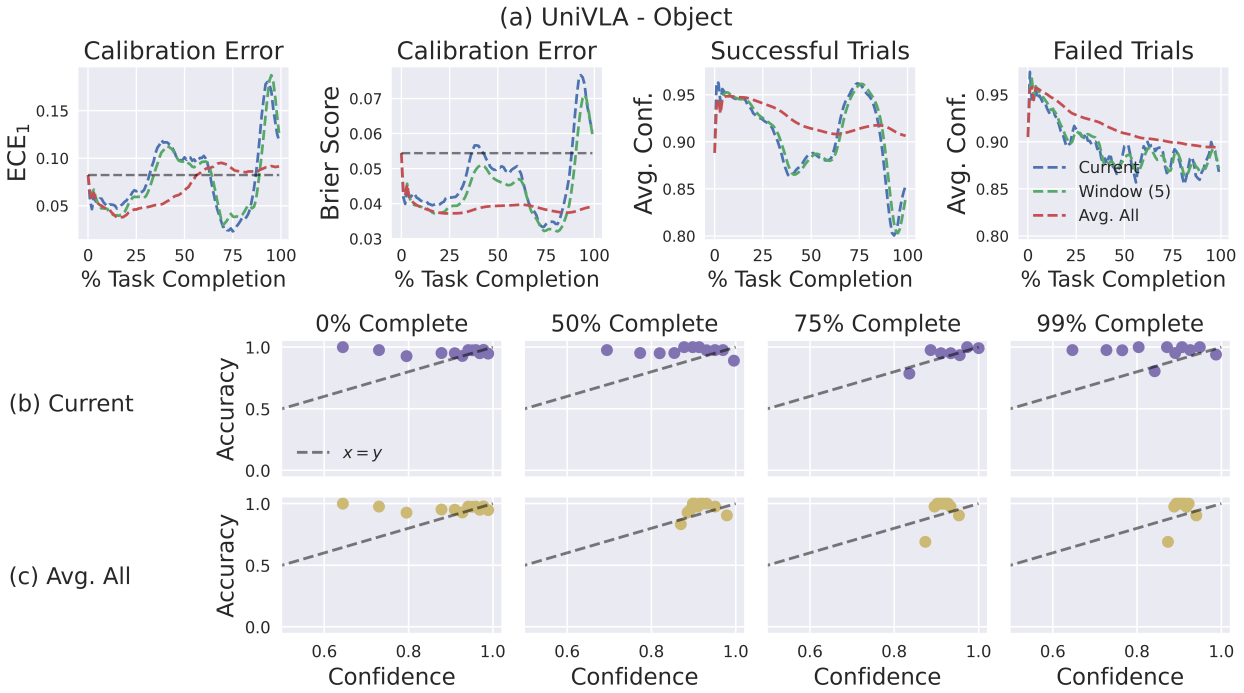

Figure 16: Empirical study of calibration error across task time horizon for the Object task suite and the UniVLA model.

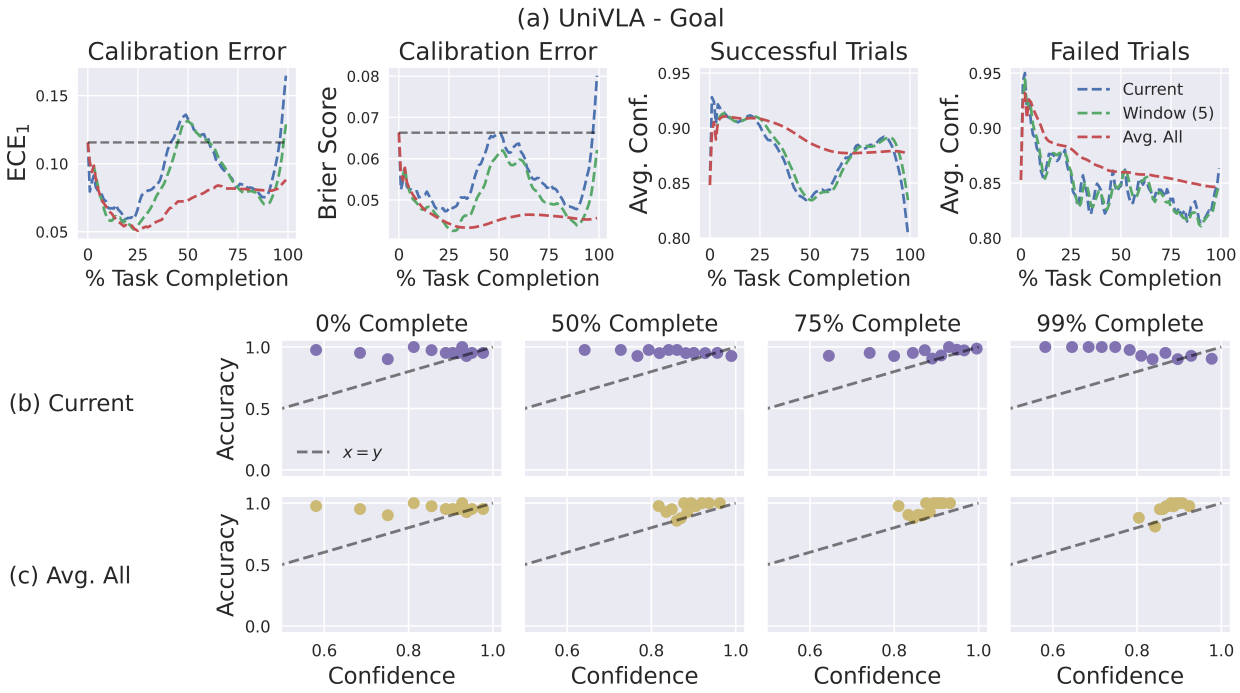

Figure 17: Empirical study of calibration error across task time horizon for the Goal task suite and the UniVLA model.

C.5.1 Qualitative Examples of Context-Aware Confidence Monitoring

Additional qualitative examples of our confidence-aware monitoring demonstration are provided in Figure 18. These include cases where the strategy succeeds and others where it fails (e.g., confidence falls below the threshold while grasping in an ultimately successful trial).

Consider the episode shown across the top row (marked (a), Episode 29 of the 50 in the Goal task suite). Here, confidence begins high, but falls until it is slightly below the halting threshold at 20% task completion, coinciding with the difficult subtask of gripping the rounded glass bottle. Allowed to continue, the model is able to grip the bottle, and confidence rebounds as it approaches the wine rack. However, confidence once again falls quickly when it begins to set the bottle down, possibly because the difficult grip put the bottle in an unfavorable position for placement. In the end, the robot fails to place the bottle securely on the rack, with potentially negative real-world consequences. Given the opportunity, a safety intervention could have been performed at multiple points before this incident, either to avoid grasping the bottle in the first place or to reset the bottle safely on the table instead of trying a failed placement.

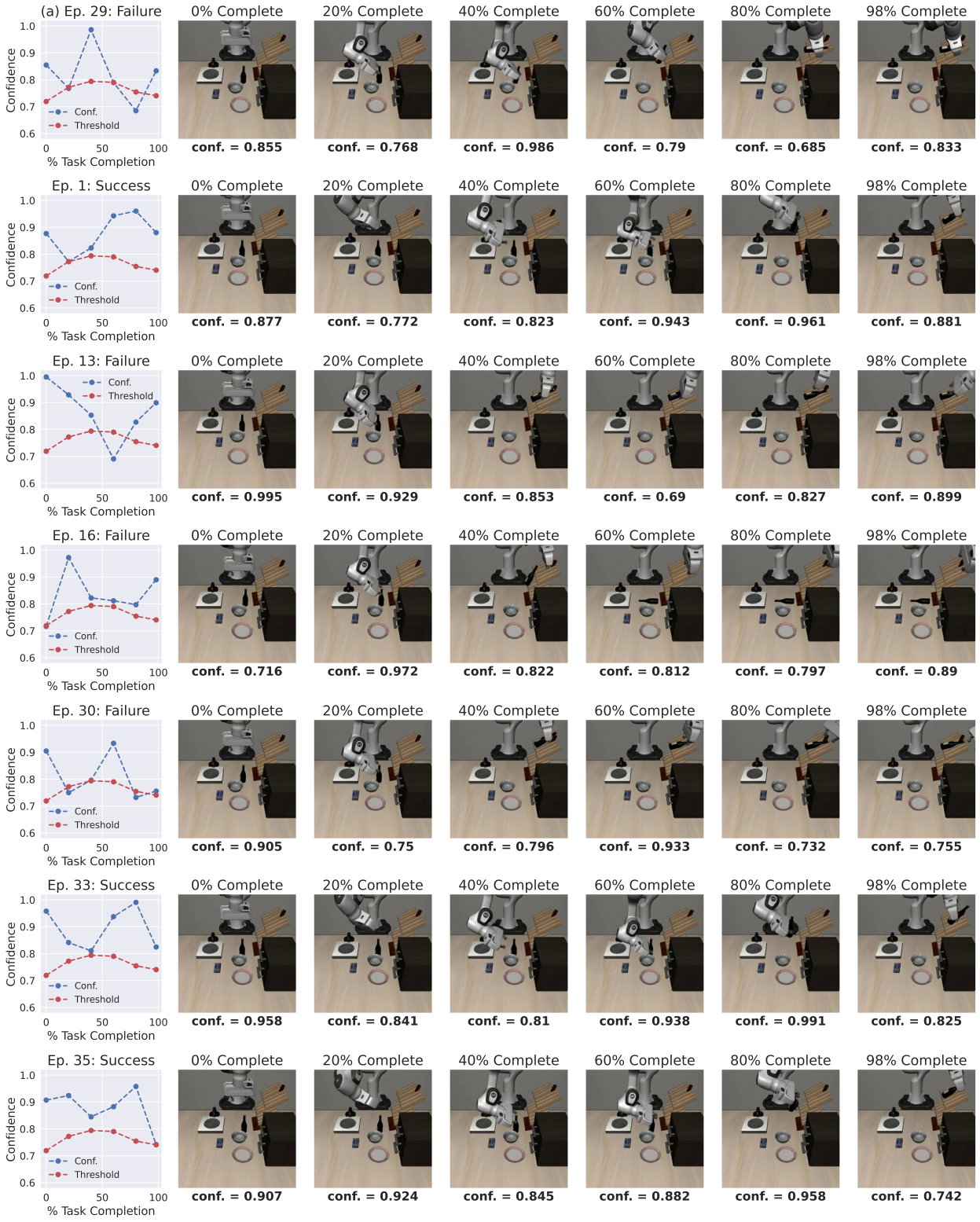

Figure 18: Qualitative examples of a context-aware confidence calibration strategy for the task "put the wine bottle on the rack". The red dashed line represents the 10% quantile of the confidence estimates output by the model across the task horizon, representing a potential threshold below which the robot may abstain from performing the task.

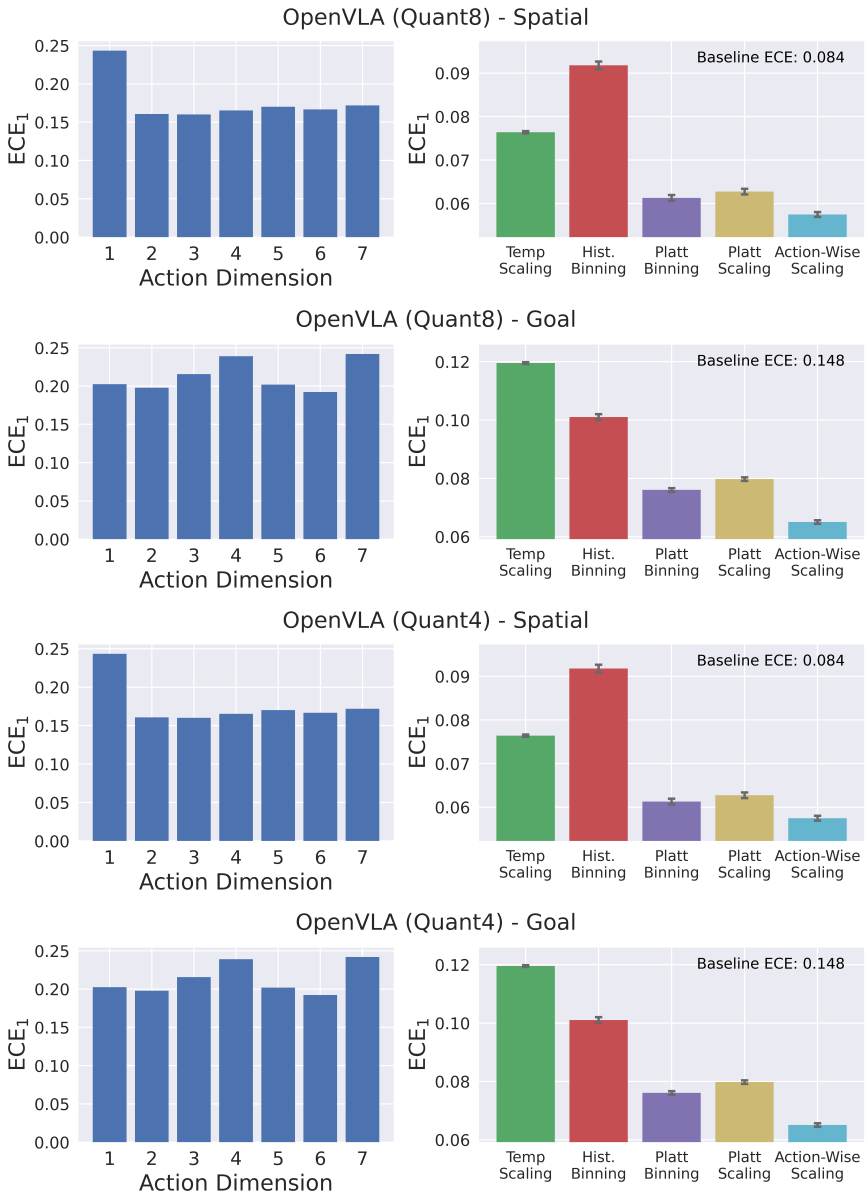

Figure 19: On the left of each pair of plots, we compare miscalibration across action dimensions. On the right, we compare the performance of typical Platt scaling, temperature scaling, histogram binning, and Platt binning to action-wise Platt scaling.

C.6 Calibration Across Action Dimensions

Figure 19 shows additional results for the action-wise recalibration experiments.