World Models That Know When They Don't Know:
Controllable Video Generation with Calibrated Uncertainty

Zhiting Mei[1*], **Tenny Yin**[1], **Micah Baker**[1], **Ola Shorinwa**[1*], **Anirudha Majumdar**[1]

[1]Princeton University
[*]Equal contribution.

Recent advances in generative video models have led to significant breakthroughs in high-fidelity video synthesis, specifically in controllable video generation where the generated video is conditioned on text and action inputs, e.g., in instruction-guided video editing and world modeling in robotics. Despite these exceptional capabilities, controllable video models often *hallucinate*—generating future video frames that are misaligned with physical reality—which raises serious concerns in many tasks such as robot policy evaluation and planning. However, state-of-the-art video models lack the ability to assess and express their confidence, impeding hallucination mitigation. To rigorously address this challenge, we propose $\mathbf{C}^3$, an uncertainty quantification (UQ) method for training *continuous-scale calibrated controllable* video models for dense confidence estimation at the subpatch level, precisely localizing the uncertainty in each generated video frame. Our UQ method introduces three core innovations to empower video models to estimate their uncertainty. First, our method develops a novel framework that trains video models for *correctness* and *calibration* via strictly proper scoring rules. Second, we estimate the video model's uncertainty in latent space, avoiding training instability and prohibitive training costs associated with pixel-space approaches. Third, we map the dense latent-space uncertainty to *interpretable* pixel-level uncertainty in the RGB space for intuitive visualization, providing high-resolution uncertainty heatmaps that identify untrustworthy regions. Through extensive experiments on large-scale robot learning datasets (Bridge and DROID) and real-world evaluations, we demonstrate that our method not only provides calibrated uncertainty estimates within the training distribution, but also enables effective out-of-distribution detection.

Keywords: Controllable Video Models, Uncertainty Quantification, Trustworthy Video Synthesis.

Website: c-cubed-uq.github.io

Code: github.com/irom-princeton/c-cubed

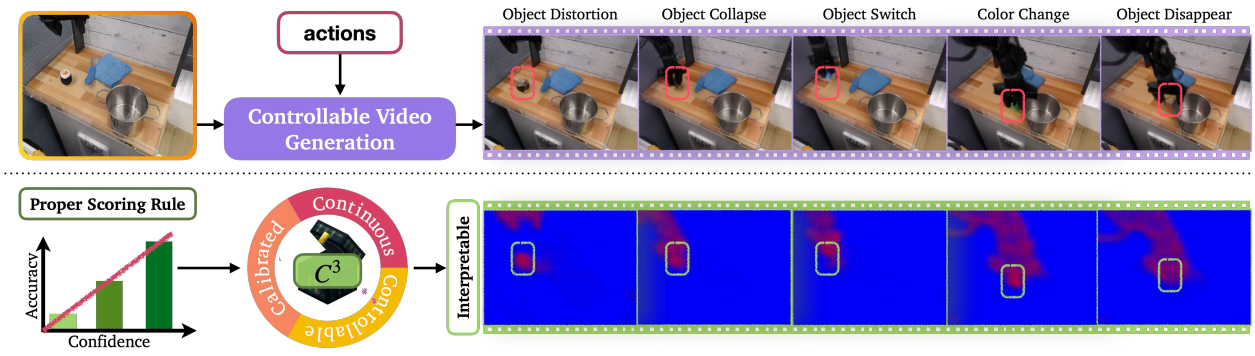

Figure 1 We present C^3, the first method for training video models that know when they don't know. Using proper scoring rules, C^3 generates dense confidence predictions at the subpatch (channel) level that are physically interpretable and aligned with observations.

1 Introduction

Conditioned on text or action inputs, state-of-the-art (SOTA) controllable generative video models [1–4] are capable of synthesizing high-fidelity videos with rich visual content across diverse task settings. However, these models have a high propensity to hallucinate, i.e., to generate new video frames that are physically inconsistent, posing a significant hurdle in applications that demand trustworthy video generation. For example, such hallucinations prevent them from reliable integration in scalable evaluation of generalist robot policies and visual planning [5–7]. Despite their tendency to hallucinate, video generation models lack the fundamental capacity to express their uncertainty, which hinders their trustworthiness. To the best of our knowledge, only one existing work attempts to quantify the uncertainty of video models [8]. However, the resulting estimates only capture task-level uncertainty, failing to resolve the model's uncertainty spatially and temporally at the frame-level, which is essential for safe decision-making.

To address this critical challenge, we present $\mathbf{C}^3$, an uncertainty quantification (UQ) method for *calibrated controllable* video synthesis, enabling subpatch-level confidence prediction at any resolution in video generation accuracy, i.e., at *continuous* scales. We make three central contributions to derive continuous-scale calibrated controllable video generation models. First, we introduce a novel framework for training video generation models for both *accuracy* and *calibration*, founded on proper scoring rules as loss functions, effectively teaching video models to quantify their uncertainty during the video generation process. We demonstrate that the resulting uncertainty estimates are *well-calibrated* (i.e., neither underconfident nor overconfident) using benchmark robot learning datasets, including the Bridge [9] and DROID [10] datasets.

Second, we derive our UQ method directly in the *latent space* of the video model. This key design choice circumvents the high computation costs associated with video generation in the (higher-dimensional) pixel space. Further, operating in the latent space streamlines applicability of our proposed method to a wide range of SOTA latent-space video model architectures [1–3], without requiring specialized knowledge or adaptation for implementation. Moreover, we compute *dense* uncertainty estimates at the subpatch level for high-resolution uncertainty quantification, with more fine-grained detail compared to patch-level UQ representations.

Third, we decode latent-space uncertainty into *interpretable* pixel-space confidence estimates via temporal RGB heatmaps for intuitive visualization. We show that the uncertainty heatmaps are well-aligned with physical intuition, with areas of greater uncertainty associated with dynamic interactions between the robot and its environment. More importantly, the resulting heatmaps localize hallucinations in each video frame, highlighting untrustworthy areas of the video. Further, we show that the model's confidence estimates are negatively correlated with the error between the generated video and the ground-truth video, which is also consistent with intuition.

Finally, we demonstrate the effectiveness of C^3 in detecting *out-of-distribution* (OOD) inputs (i.e., environment conditions and actions) in controllable video generation through real-world experiments on a WidowX 250 robot. In these settings, we show that C^3 is able to provide calibrated uncertainty estimates, even when the quality of the generated video is significantly compromised given the distribution shift at test time.

2 Related Work

Video Generation Models. Research breakthroughs in video generation have led to significant advances in the capabilities of generative video models in recent years. While early video generation models were limited to generating short-duration videos (consisting of only a few frames) with small temporal changes, SOTA models can generate seconds-long videos (consisting of hundreds of frames) with impressive photorealistic detail. Early methods [11–13] synthesize novel videos by applying local (pixel-level) transformations to input images, composing these transformations to capture more complex temporal changes. However, these methods are limited to localized scene updates generally centered around a target object in the scene video, with little to no changes in the background. Moreover, these methods lack sufficient expressivity to generate photorealistic videos. To address these limitations, subsequent work [14–16] employs generative adversarial networks (GANs) [17], training a generator and a discriminator to generate higher-quality videos. Although more effective than earlier work, these methods often fail to capture the diversity inherent in video generation

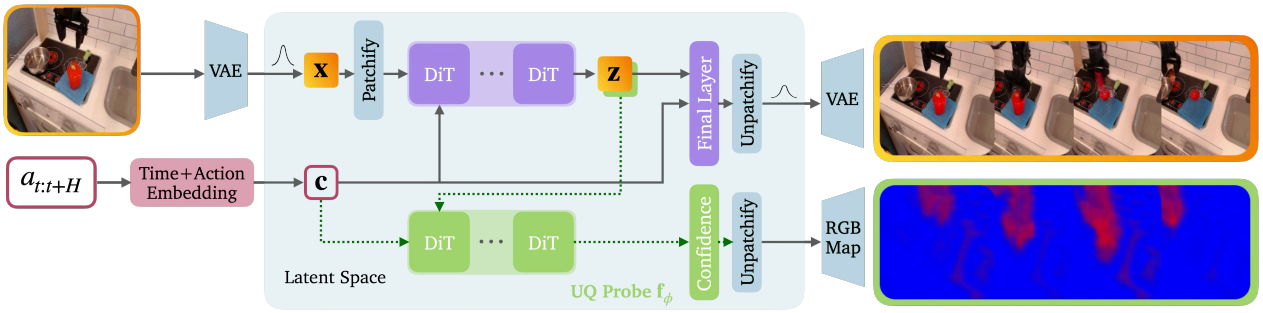

Figure 2 Model Architecture. C^3 enables simultaneous video generation and uncertainty quantification (visualized as a heatmap), quantifying the model's confidence in its accuracy using a UQ probe acting on the video latents.

[18], a challenge referred to as mode collapse. In addition, they require carefully designed model architectures to prevent training instability [19, 20]. Other approaches [21–23] generate videos within a learned latent space using variational inference with variational autoencoders (VAEs) [24] to overcome mode collapse. In contrast to these methods, SOTA methods [1–3, 25] leverage diffusion-based or flow-based modeling [26–28] for high-fidelity video generation, sampling future frames in a discrete latent space learned using vector-quantized autoencoders (VQ-VAE) [29] or vector-quantized generative adversarial networks (VQ-GANs) [30]. Although these methods support text-to-video and image-to-video generation, they do not support (robot) action-conditioned generation, which is essential in many robotics applications, such as robot policy learning and evaluation. Hence, more recent work [1, 5, 6] has explored finetuning text- or image-conditioned video generation models with action-labeled data.

Uncertainty Quantification of Video Models. Uncertainty quantification of large language models (LLMs) has been been extensively studied (see [31] for a review of UQ methods for LLMs); however, only a few papers have explored uncertainty quantification of image or video generation models. Like traditional methods for uncertainty quantification of deep neural networks [32], prior work on uncertainty quantification of generative image models applies Bayesian methods to image diffusion models to estimate epistemic and aleatoric uncertainty using a variance-decomposition-based approach [33]. Another approach [34] takes an ensemble-based UQ perspective, estimating the uncertainty of image diffusion models using the mutual information over the distribution of the weights of an ensemble of the diffusion models. Some other methods [35] extract language descriptions from the generated images, facilitating uncertainty quantification of image generation models using established UQ methods for language models. Extending these methods to uncertainty quantification of video generation models is not trivial, given the spatio-temporal form of videos and the significant computation costs of these methods. One existing method [8] directly considers uncertainty quantification of video models. However, this method only provides a single confidence estimate for each generated video and thus fails to provide more informative *dense* confidence estimates at the frame-level or pixel-level, which we explore in this work.

3 Uncertainty Quantification of Action-Conditioned Video Models

For simplicity, we limit the discussion of our uncertainty quantification method to video diffusion or flow-based models, given their SOTA performance. We provide a brief review of these models in Appendix A. However, we note that our proposed method readily applies to other video generation model architectures, such as GAN-based/RNN-based video models, with relatively straightforward adaptations.

For video generation, we adopt the latent diffusion transformer (DiT) architecture, described by:

$$\begin{aligned} \mathbf{x} &= \text{Encode}\ (\mathbf{v}, \mathbf{g}), \\ \hat{\mathbf{x}} &\sim \text{DiT}(\mathbf{x}, \mathbf{a}), \\ \hat{\mathbf{o}} &= \text{Decode}(\hat{\mathbf{x}}), \end{aligned} \tag{1}$$

where $\mathbf{v}$ denotes the input video frames, $\mathbf{g}$ denotes other conditioning inputs (e.g., text or action), $\mathbf{x} \in \mathcal{U}$

denotes the encoded conditioning inputs in the latent video space $\mathcal{U}$, and $\mathbf{a} \in \mathcal{A}$ represents the sequence of actions carried out by the agent.

3.1 Confidence Prediction

We introduce C^3, a method for uncertainty quantification of controllable (action-conditioned) video generation models that provides dense estimates of the model's confidence in the accuracy of each video frame at the subpatch level, conditioned on input actions. Concretely, we train the video model $\mathcal{V}_\theta : \mathcal{U} \times \mathcal{A} \to \mathcal{U} \times \mathcal{U}$ to generate accurate video frames with corresponding dense confidence estimates, given by:

$$\hat{\mathbf{x}}, \hat{\mathbf{q}} \sim \mathcal{V}_\theta(\mathbf{v}, \mathbf{g}, \mathbf{a}), \tag{2}$$

where $\hat{\mathbf{q}} \in \mathcal{U}$ is the confidence prediction. Each element in $\hat{\mathbf{q}}$ corresponds to the model's confidence in the accuracy of the associated subpatch of the generated latent video $\hat{\mathbf{x}}$.

Traditional UQ methods, such as Monte Carlo-based methods or ensemble-based techniques, generally require multiple forward passes or multiple instances of the model to estimate uncertainty. However, video diffusion models typically have billions of parameters, making these methods too computationally expensive and generally intractable. To overcome these challenges, we take a novel approach to uncertainty quantification of video models. First, we pose uncertainty quantification as a classification problem over the accuracy of the generated video, seeking to assess the model's confidence in the video accuracy. This key choice eliminates the limitations associated with making simplifying modeling assumptions to predict the accuracy of the generated videos. In particular, we avoid inductive biases associated with restricting the predicted accuracy to a specific class of probability distributions, e.g., the Gaussian distribution, which could limit the calibration of the computed uncertainty estimates. We circumvent this bias via the classification problem.

Given the high computational cost of video generation, we design a transformer-based uncertainty quantification probe $\mathbf{f}_\phi : \mathcal{U} \to \mathcal{U}$ to estimate the video model's confidence directly in *latent space*. We integrate $\mathbf{f}_\phi$ within the video generation framework for simultaneous video generation and uncertainty quantification during training and inference. However, we note that both components can also be trained independently. Figure 2 provides an overview of our framework.

For more efficient training, we generate the videos in latent space using a vector-quantized variational autoencoder (VQ-VAE) with spatio-temporal convolution and attention layers to map input video frames to a lower-dimensional latent space. Specifically, we roll-out the forward and reverse diffusion processes in latent space, before mapping the generated latent video to the pixel space using a decoder. In this work, we utilize pre-trained VQ-VAEs [1, 36, 37] trained with a reconstruction objective to compress (high-dimensional) videos into a compact latent space. In addition, we leverage diffusion forcing [38] for independent per-sample noise schedules, in line with prior work.

For action-conditioned video generation, we compute action embeddings from input actions using a multi-layer perceptron and sum the resulting action embeddings with the timestep embedding computed using frequency-space encodings. We feed the resulting embeddings as the conditioning input $\mathbf{c}$ to the DiT. Given the input video frame and action, we extract the internal features $\mathbf{z}$ of the diffusion transformer from the penultimate layer, which is passed into $\mathbf{f}_\phi$, alongside the action and timestep embedding, to predict the subpatch (channel-wise) confidence $\hat{\mathbf{q}}$. This confidence represents the probability that each subpatch of the generated latent video is accurate with respect to an element-wise boolean function $\mathbf{acc}$, which we elaborate in Section 3.2.

3.2 Model Architectures

The accuracy function $\mathbf{acc}$ can be defined to induce different model architectures. To demonstrate our method's amenability to different realizations, we consider three possible architectural instantiations of C^3. We define $\mathbf{acc}$ in terms of a distance function $\mathbf{d}$. In this work, we use the L_1 loss:

$$\mathbf{d}(\hat{\mathbf{x}}, \mathbf{x}^\star) \coloneqq |\hat{\mathbf{x}} - \mathbf{x}^\star|, \tag{3}$$

although other distance metrics can also be used, e.g., the squared deviation.[1] However, we emphasize that in our setting, all p-norms simplify to the L_1 loss in Equation (3) since $\mathbf{d}$ is applied element-wise, making them equivalent. We use the L_1 loss for simplicity. Given $\mathbf{d}$, the accuracy function maps the generated videos to a binary-valued output space of the same dimensions as $\mathcal{U}$, where each element is in $\{0, 1\}$, given by the boolean operator:

$$\mathbf{acc}(\hat{\mathbf{x}}, \mathbf{x}^\star, \varepsilon) \coloneqq \mathbf{d}(\hat{\mathbf{x}}, \mathbf{x}^\star) \leq \varepsilon, \tag{4}$$

based on the errors between the ground-truth and generated videos, given a threshold ε. The technique used in specifying ε induces a range of model architectures, namely: (i) fixed-scale classification models, (ii) multi-class classification models, and (iii) continuous-scale classification models, which we describe in the following subsections. We train all variants of our model with proper scoring rules to ensure their calibration. We provide a brief overview of proper scoring rules in Appendix B.

Fixed-scale classification model (FSC). The FSC model predicts the accuracy of generated videos at a fixed accuracy resolution during training and inference, and thus requires the specification of a single error threshold ε. By requiring only a single value of ε, FSC models are typically faster to train than other models at the cost of generality to a range of resolutions. In practice, we select a value of ε that is appropriate for the task domain. As is standard in classification problems, we train $\mathbf{f}_\phi$ to predict the log-probabilities (logits) and use the sigmoid function σ to map these values to valid confidence estimates $\hat{\mathbf{q}}$ that lie within the interval $[0, 1]$:

$$\hat{\mathbf{q}} = \sigma(\mathbf{f}_\phi(\mathbf{z}, \mathbf{c})), \tag{5}$$

where $\mathbf{f}_\phi$ is the confidence probe, $\mathbf{z}$ is the latent internal feature, and $\mathbf{c}$ is the latent action/time embedding. We optimize the parameters of $\mathbf{f}_\phi$ with the Brier score loss function, given by: $\text{BS} = \mathbb{E}_y(\hat{q} - y)^2$ with ground-truth accuracy y and predicted confidence $\hat{q}$ for the prediction over a single subpatch. We sum over all subpatches in computing the loss for each video. With a slight overload in notation, $y, \hat{q}$ denote each component in $\mathbf{y}, \hat{\mathbf{q}}$, respectively.

Multi-class classification model (MCC). Inspired by the effectiveness of classical UQ methods for large language models [31], we pose video model UQ as a multi-class classification problem by discretizing the output space of predictions into confidence bins, with the corresponding **acc** defined by:

$$\mathbf{acc}(\hat{\mathbf{x}}, \mathbf{x}^\star, O_i) \coloneqq \varepsilon_i \leq \mathbf{d}(\hat{\mathbf{x}}, \mathbf{x}^\star) < \varepsilon^i, \tag{6}$$

where O_i represents the i-th bin with lower-bound ε_i and upper-bound ε^i. For each subpatch of the generated video, the MCC model predicts its confidence that the corresponding subpatch is accurate with respect to the accuracy thresholds associated with the bin. Like the FSC model, the MCC model predicts the logits for each bin, which is subsequently mapped to valid confidence (probability) values $\hat{\mathbf{q}}$ using the softmax. We optimize $\mathbf{f}_\phi$ with the cross-entropy loss function, which is a strictly proper scoring rule given by: $\text{CE} = \mathbb{E}_y\big[-\sum_k y_k \log q_k\big]$, with ground-truth value y and predicted confidence q_k for the k-th bin.

Continuous-scale binary classification model (CS-BC). To demonstrate the expressiveness of our approach, we train a continuous-scale model for any-resolution confidence prediction, conditioning $\mathbf{f}_\phi$ on an accuracy threshold ε specified at inference. During training, we uniformly sample a set of ε_v at each iteration to ensure sufficient coverage of the thresholds. In practice, for faster training, we discretize the space of ε using an adaptive hierarchical technique, by first dividing ε into uniform bins and further adaptively subdividing bins for higher resolution. When domain knowledge is available, more informed discretization or sampling schemes can be used for more efficient training. Like the preceding models, the CS-BC model predicts logits for each subpatch, which is mapped to confidence estimates using the sigmoid function:

$$\hat{\mathbf{q}} = \sigma(\mathbf{f}_\phi(\mathbf{z}, \mathbf{c}, \varepsilon)), \tag{7}$$

taking the conditioning input threshold ε. In our experiments, we explore training the models with both the Brier score and the binary cross entropy, given by: $\text{BCE} = \mathbb{E}_y\big[y \log q + (1 - y) \log(1 - q)\big]$. Note that these loss functions are proper scoring rules.

[1] The distance function in Equation (3) requires executing the reverse diffusion process to generate the latent video $\mathbf{x}$, which is computationally expensive during training. To address this challenge, we express the distance function in terms of the predicted and ground-truth *velocities*, which turns out to be a linear scaling of Equation (3). We refer readers to Appendix A for this implementation detail.

End-to-end training. We train the video generation and uncertainty quantification modules independently end-to-end using the loss function:

$$\mathcal{L}_{\theta,\phi} = \mathcal{L}_\theta + \mathcal{L}_\phi, \tag{8}$$

where θ represents the parameters of the DiTs for video generation and ϕ represents the parameters of the UQ probe. We apply a stop-gradient operator between the video generation DiTs and the UQ probe. In our ablation studies in Appendix G, we explore the effects of backpropagating gradients from the UQ probe $\mathbf{f}_\phi$ to the video generation DiTs. To optimize the loss function in Equation (8), we use stochastic gradient descent with a cosine-annealing decay schedule applied to the learning rate.

Proposition 1 (Uncertainty Decomposition)**.** *Given the input actions and video frames, the predicted confidence* $\hat{\mathbf{q}}$ *provides a calibrated measure of uncertainty of the video diffusion model in the generated video, provided that* ϕ *converges to an optimal solution.*

We provide the proof in Appendix C.

3.3 Decoding Latent Confidence Predictions

Like the latent video $\mathbf{x}$, the predicted confidence $\hat{\mathbf{q}}$ is not immediately interpretable; hence, we decode the predicted confidence from the latent space to the pixel (RGB) space for better visualization. However, simply utilizing pre-trained video tokenizers as decoders would generally yield equally uninterpretable outputs, since these pre-trained decoders are trained to map RGB embeddings from the latent space to the pixel space. As a result, we define a color map in latent space by encoding monochromatic RGB video frames into the latent space. For simplicity, we construct a latent color map from red-only, green-only, and blue-only video frames; however, higher-resolution color maps can also be constructed. We map the confidence estimates to latent RGB video frames by interpolating between the video frames in the latent color map. Subsequently, we map the latent RGB video frames for the predicted confidence to pixel space using the same tokenizer used in decoding the latent video $\mathbf{x}$. In Section 4, we demonstrate that the resulting uncertainty heatmaps are well-aligned with intuition, identifying areas of the generated video that contain hallucinations.

4 Experiments

We evaluate the performance of C^3 in uncertainty quantification of action-conditioned video models, specifically examining its calibration, interpretability, and out-of-distribution detection capabilities via the following questions: (i) Is C^3 *underconfident*, *calibrated*, or *overconfident*? (ii) Are C^3's uncertainty estimates interpretable? (iii) Can C^3 detect *OOD* inputs at inference? We provide additional ablations in Appendix G.

Datasets. We conduct experiments on the Bridge dataset [9], a standard benchmark dataset for robotics-oriented video models. The Bridge dataset consists of real-world robot trajectories collected in 24 environments on a WidowX 250 robot arm with a fixed RGB camera, capturing broad environment variations across different robot manipulation tasks. In addition, we present additional results using the DROID dataset [10] in Section F. The DROID dataset consists of trajectories collected on a Panda robot arm with a Robotiq gripper, featuring greater coverage of tasks with multi-view camera observations collected using a wrist camera and two scene cameras.

Metrics. In order to empirically evaluate the calibration of C^3, we use metrics that capture deviation from perfect calibration, including expected calibration error (ECE) and maximum calibration error (MCE) (see Equation (17)).

4.1 Are C^3's uncertainty estimates calibrated?

We examine the calibration of the uncertainty estimates computed by C^3 in dynamics prediction with controllable video models, specifically in robot manipulation tasks which constitutes a major application domain for these models, e.g., in scalable robot policy evaluation. We train the three model architectures: CS-BC, MCC, and FSC (discussed in Section 3.2) on the train dataset split of the Bridge dataset and evaluate the trained models on the test split. To assess calibration, first, we generate videos and their corresponding

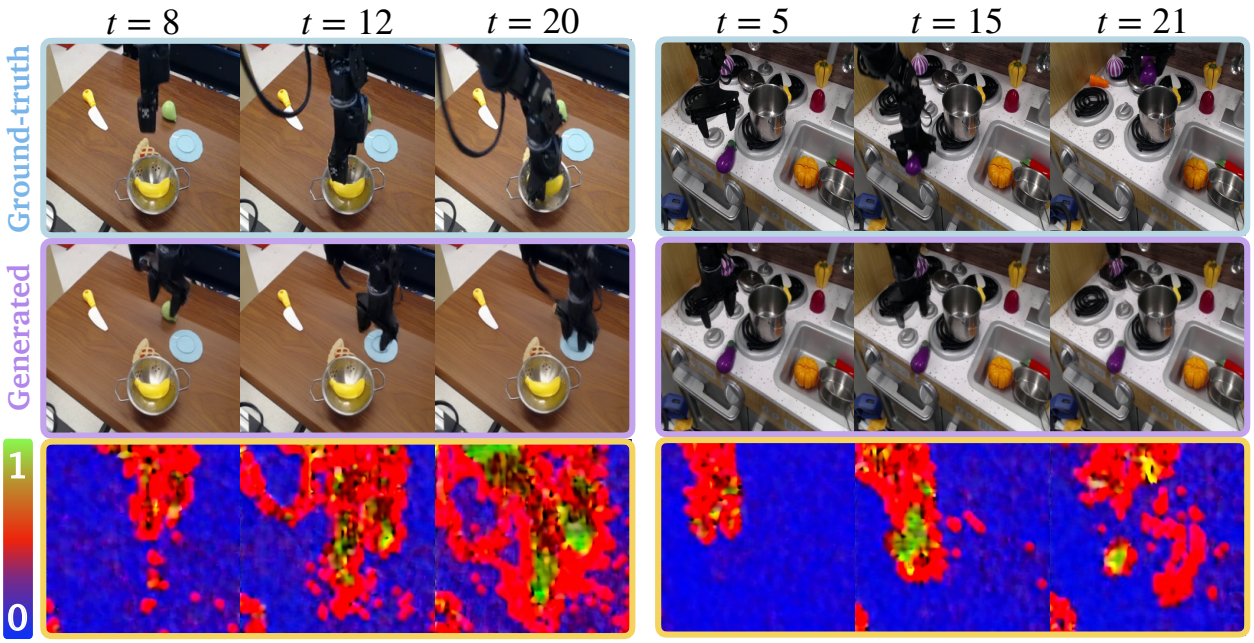

Figure 3 Latent space error. We visualize the latent-space video error in the RGB space, showing the observable range of the errors.

dense confidence predictions conditioned on the initial video frame and the entire action trajectory for each sample trajectory in the test dataset. Subsequently, we compute the expected calibration error (ECE) and the maximum calibration error (MCE) for each model, measuring the deviations from perfect calibration as described in the preceding discussion. We provide additional details on the evaluation procedure in Appendix H.

Visualizing Latent-Space Errors. We visualize the latent-space error between representative generated videos and the corresponding ground-truth videos in the RGB space in Figure 3, to aid understanding of the accuracy resolutions discussed in this work. Intuitively, one would expect a calibrated video model to more confidently identify regions with very low or high errors as accurate or inaccurate, respectively, and to be more uncertain about other regions. For alignment with human intuition, we construct a latent-space color map with three basis colors, with the extreme points of the color map defined by *blue* and *green*, corresponding to the minimum point (low-error region) and the maximum point (high-error region) of the error span, respectively. We use the color *red* to represent the middle region of the error span, which induces an interpretable confidence heatmap, discussed later in this section. Notably, Figure 3 shows that an error span over the range $[0, 1]$ is sufficient to capture essentially all observable error values. Specifically, the resulting color maps contain almost no green region, associated with the maximum end of the error span. We find that most of the error values lie within the selected error span, further justifying the accuracy resolutions utilized in our experiments.

Calibration Errors. In Figure 4, we show the average ECE and MCE of each model across all the test videos. For the continuous-scale model CS-BC, we compute the average errors across ten equally-spaced error thresholds ε_v, spanning the observable latent-space prediction error domain as visualized in Figure 3.[2] In contrast, the FSC model does not take in an accuracy scale for conditioning; as a result, we compute the ECE and MCE at the fixed-scale used in training the model. (For more informative evaluation and more comparative results, we select the fixed scale to lie within the range of ε_v used by the CS-BC model.) Similarly, we compute the ECE and MCE for all classes (accuracy scales) in the MCC model and report the average values in Figure 4. The results indicate that C^3 produces *well-calibrated* uncertainty estimates across all models. Although all models achieve relatively the same MCE, their performance on the ECE differs. This small difference can be explained by the tradeoff between continuous-scale calibration and fixed-scale

[2]As described in Footnote 1, ε_v is the linearly scaled version of ε specifying the deviation between predicted and ground-truth deviations. See Appendix A for the full mathematical derivation of the scaling factor.

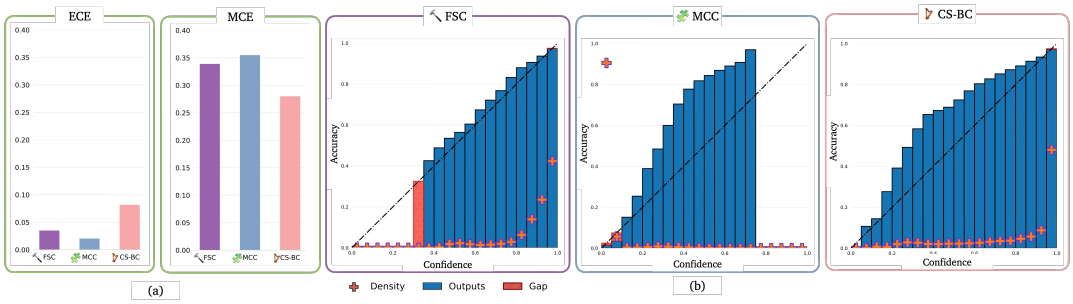

Figure 4 (a) **Average calibration error.** All three architectures have very low ECE, and relatively low MCE. (b) **Aggregated reliability diagrams.** All methods are well-calibrated, closely tracking the line of perfect calibration.

calibration. The improved expressiveness and flexibility of continuous-scale calibration might come at the cost of some reduction in calibration at a single (specific) scale. The converse holds for the fixed-scale model, which is less expressive and flexible. The multi-class classification variant lies between these extremes on the tradeoff curve. We emphasize that the superior calibration performance of C^3 arises from the use of proper scoring rules.

Underconfidence vs. Overconfidence. Next, we assess underconfidence vs. overconfidence of our proposed UQ method using reliability diagrams, which visualize the calibration error associated with the uncertainty estimates across different confidence bins. In Figure 4(b), we show the reliability diagram for each model averaged across all thresholds. The dashed line in each plot traces the path of perfect calibration, while the cross-shaped markers indicate the density of samples in each bin. Across all models, we observe that with C^3, the video models are well-calibrated, i.e., neither underconfident nor overconfident, closely tracking the dashed line in the reliability diagrams across all confidence bins. Notably, the models tend to be more conservative when unsure about the accuracy of the generated video, as visualized by the bars in the $[0.3, 0.7]$ confidence bins exceeding the dashed line. This emergent behavior aligns well with trustworthiness in safety-critical applications, with a greater propensity for the model to express doubt when unsure about the accuracy of the generated video. Further, in Figure 5, we compare calibration of the CS-BC and FSC models at the fixed scale ($\varepsilon_v = 0.5$) used in training the FSC model. We find that both models are well-calibrated, producing relatively the same reliability diagrams.

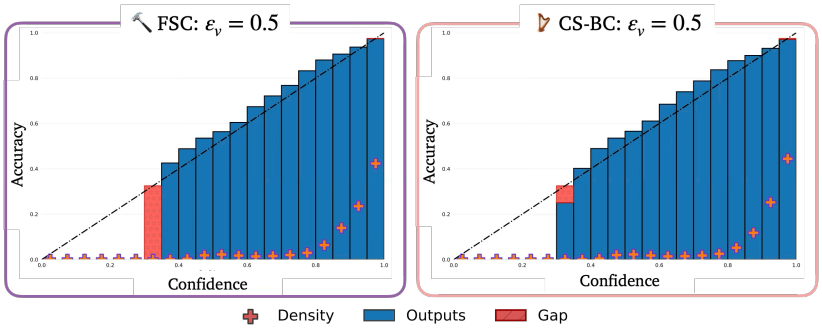

Figure 5 Reliability diagrams for FSC and CS-BC. At the same threshold ε_v (0.5), both models achieve the similarly well-calibrated confidence predictions.

In Figure 6, we provide more detailed results for the CS-BC model, showing calibration across different accuracy threshold levels. Overall, we see that C^3 is well-calibrated across each error threshold ε_v, with the top of the confidence bins tracing the diagonal line. Further, we observe greater uncertainty at lower thresholds (e.g., $\varepsilon_v = 0.2$) which aligns with the intuition that lower accuracy thresholds are generally associated with greater uncertainty in the accuracy of the prediction given the tightness of the threshold. Conversely, as the threshold increases, the sample densities gradually shift right toward the higher confidence region, aligning with the intuition that larger thresholds afford greater confidence in the accuracy of the generated videos. Moreover, we observe that at extremely low values of ε_v ($\varepsilon_v \leq 0.3$), C^3 tends to be underconfident, signified by the histograms going above the line of perfect calibration. Further, we highlight that the model's degree

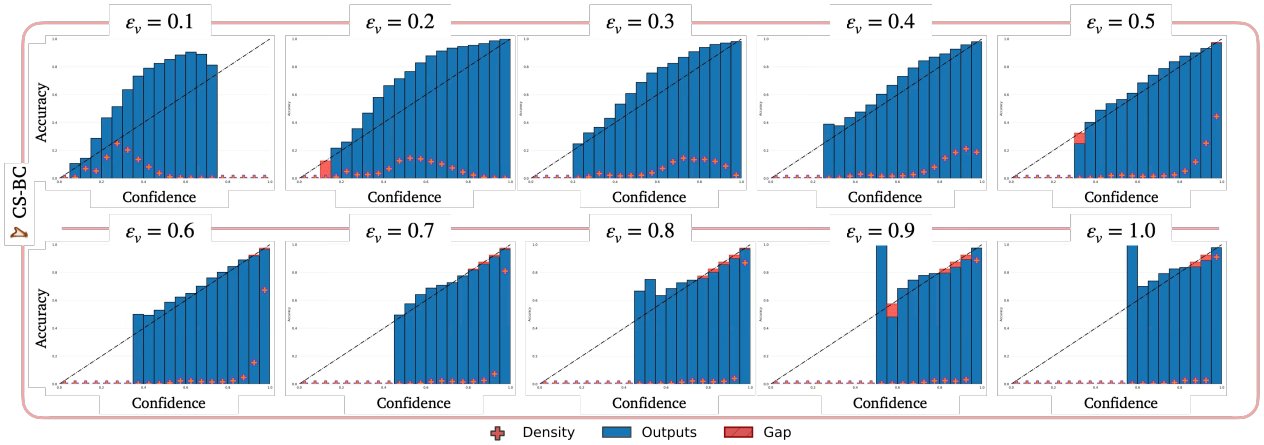

Figure 6 Reliability diagram for each threshold. C^3 is well-calibrated across all accuracy thresholds, with some degree of conservativeness at very low thresholds.

of underconfidence decreases as the accuracy threshold increases. Overall, our observations are well-aligned with safety. Essentially, with C^3, the video model tends to be more conservative at very low accuracy thresholds, mitigating against false negatives—i.e., inaccurate patches that are identified as highly confident regions—which could otherwise lead to potentially harmful consequences.

4.2 Are C^3's uncertainty estimates interpretable?

Here, we examine the interpretability of the video model's uncertainty estimates in video trajectory prediction in robot manipulation.

Qualitative Results. In Figure 7, we show the ground-truth and generated videos from the video model, along with a visualization of the video model's confidence using a confidence heatmap, which transforms the model's confidence predictions to the RGB color space using a color map. The heatmap contains three prominent regions: (i) *blue* regions which signify areas of *high confidence* in the accuracy of the predicted pixel, corresponding to locations where the model is relatively certain that the generated video is correct; (ii) *red* regions which represent areas of *high uncertainty*, corresponding to locations where the model is unsure if the generated video matches the ground-truth video; and (iii) *green* regions which highlight areas where the model is highly confident about the *inaccuracy* of the generated video. The heatmap applies a linear transformation to the raw confidence values, smoothly interpolating between the basis colors in the map.

In Figure 7, we observe that the video model is very confident about the accuracy of the background in the generated video but more uncertain about the robot location and interaction in the video. This finding is consistent with intuition. The video background closely matches the ground-truth video, and is thus sufficiently far away from accuracy-defining margin. On the other hand, predicting the robot motion is more challenging with unobserved dynamics effects, increasing the video model's uncertainty. These results underscore the interpretability of our proposed UQ method. We provide additional visualizations at lower error thresholds in Appendix D.

Quantitative Results. We assess the correlation between the estimated confidence of the video model and the error between the ground-truth and generated latent videos using the *Shepherd's Pi* correlation [39], which is a robust correlation method that uses bootstrapping to identify outliers that would otherwise skew the correlation coefficient. For calibrated models, one would expect a negative correlation between the estimated confidence and the error between the ground-truth and generated videos, generally indicating an increase in the uncertainty of the video model as the video error increases. As expected, for the FSC and CS-BC models, we observe a statistically significant *negative* correlation of -0.373 and -0.172 between the confidence estimates and the absolute errors in the generated video at a 99% significance level, respectively. However, for the MCC model, we obtain a positive correlation coefficient due to the inadequate supervision of rightmost bins which correspond to greater latent error values, given that most of the generated video patches have a

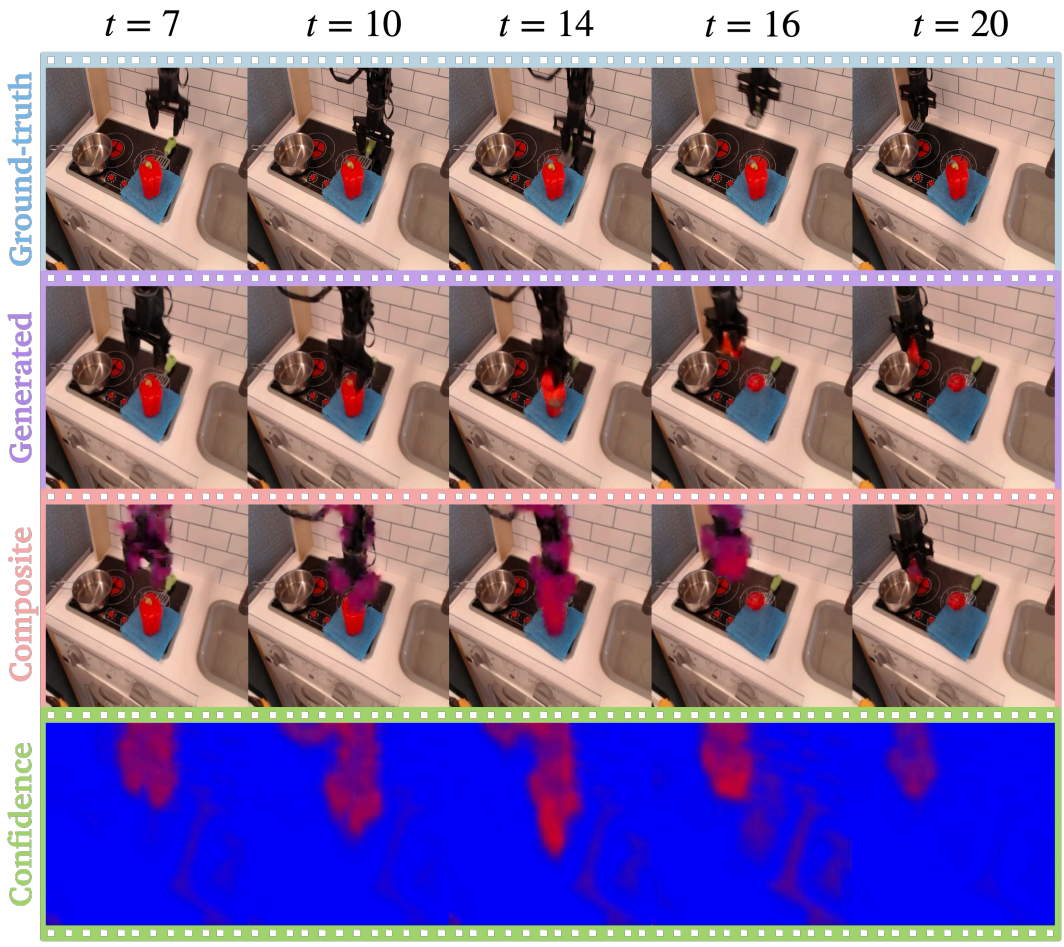

Figure 7 Confidence heatmap visualization. We show the ground-truth and generated video frames, along with the composited confidence map and the full confidence map at higher error thresholds. The video model is more uncertain about the robot's motion compared to the video background at these error thresholds, which is well-aligned with human intuition.

notably small latent error (see Figure 3). For a more informed analysis, we examine the correlation of the confidence estimates of the MCC model with the maximum bin edge set at 0.2. We find that the confidence error is negatively correlated with the video error at the 99% significance level, with a coefficient of -0.130.

From the qualitative results in Figure 7, we note that action-conditioned video models are more likely to be unsure about regions associated with robot end-effector motion and object interaction compared to background regions, which aligns with human intuition. Essentially, motion and dynamic interactions increase uncertainty in the generated video, since these dynamic events depend on many physical parameters, such as the object mass, coefficient of restitution, friction, and gravity, that are *unobservable* from a single conditioning frame. We find that the high-resolution confidence estimates produced by C^3 capture hallucinations, localizing regions of the generated video where the model inserts artifacts such as previously non-existent objects or morphed objects. For example, in Figure 8, we visualize areas with an hallucinated object in the robot's gripper across all video streams. Moreover, the training dataset consists of both rigid and deformable objects, which introduces uncertainty in the object's response to the grasp force during video prediction. The model's confidence estimates reflect this uncertainty, which is indicated by the concentration of the red (high-uncertainty) regions around the grasped object in Figure 9. Some of this uncertainty can be reduced by increasing the history length used as the conditioning input to the video model. We leave an exhaustive exploration of the influence of history conditioning on confidence prediction to future work. Further, our proposed method is able to capture uncertainty from occlusions, which is shown in Figure 10. This observation is also in line with human intuition, further demonstrating the interpretability of our approach.

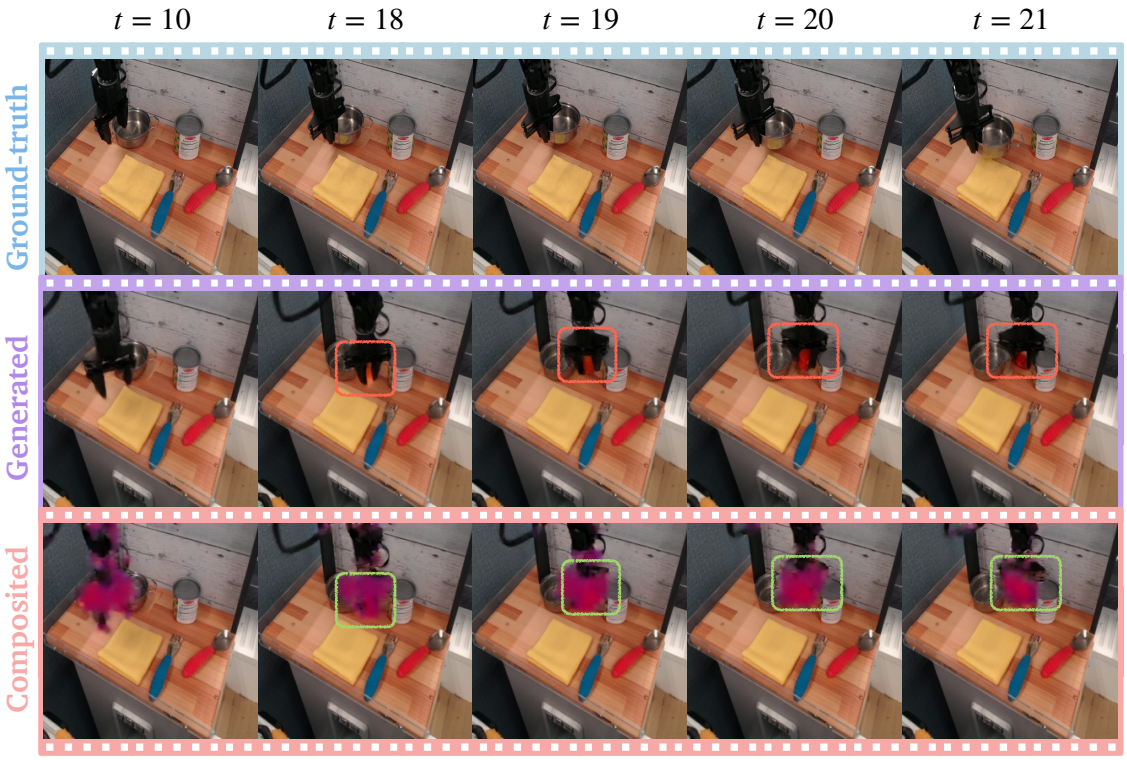

Figure 8 Capturing Hallucinations. The boxed areas highlight severe hallucination of the video generation model, where C^3 is able to capture the sub-patch level high uncertainty content.

4.3 Detecting OOD Inputs at Inference

Here, we explore the performance of C^3 in out-of-distribution (OOD) detection at inference time, noting the importance of calibrated uncertainty estimates in reliable OOD detection. Concretely, a trustworthy video model would express higher uncertainty when given a task that lies outside of the training distribution, reflecting its lack of knowledge of the scene and object dynamics. We examine the calibration of the uncertainty estimates computed by our method in these settings through real-world experiments on a WidowX 250 robot in the Bridge setup within a toy kitchen environment.

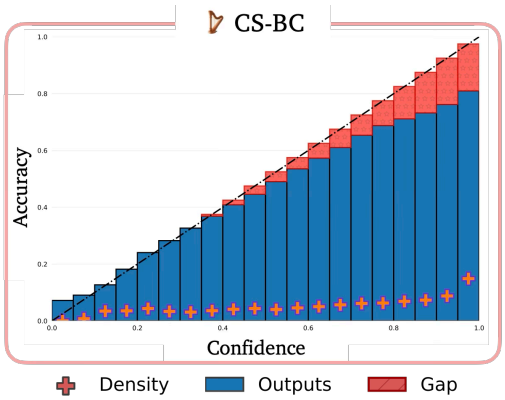

Figure 11 Reliability diagram in OOD conditions. C^3 provides calibrated uncertainty estimates in OOD scenes.

We consider OOD conditions across five axes: background, lighting, environment clutter, target object (task), and robot end-effector, creating environment settings that are noticeably different from those seen in the Bridge dataset. For the *background* axis, we introduce novel background objects into the scene, e.g., computer accessories and sport equipment. For *lighting*, we vary the RGB value of the environment lighting. We add more objects to the scene in the *environment clutter* setting and introduce novel target objects or objects in unseen configurations for grasping in the *target object* test setup. Along the *end-effector* axis, we create OOD conditions by modifying the appearance of the end-effector by attaching lightweight objects (e.g., a towel, plushy toys, etc.) to the robot without noticeably altering the robot dynamics. In these settings, we collect 50 ground-truth trajectories (10 trajectories per setting) and generate videos from the video model using the associated robot actions.

In Figure 12, we show the ground-truth and generated videos and confidence estimate visualizations for one trajectory in each of the five OOD categories. For unfamiliar background objects (e.g., a skeleton), we observe that the model becomes uncertain about the dynamics between the robot and the background object as it

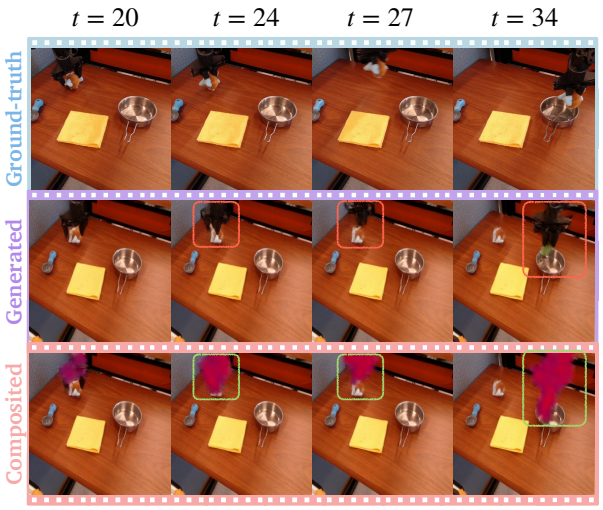

Figure 9 Uncertainty in Object Interactions. The video model is uncertain about the object dynamics during interaction, resulting in high uncertainty.

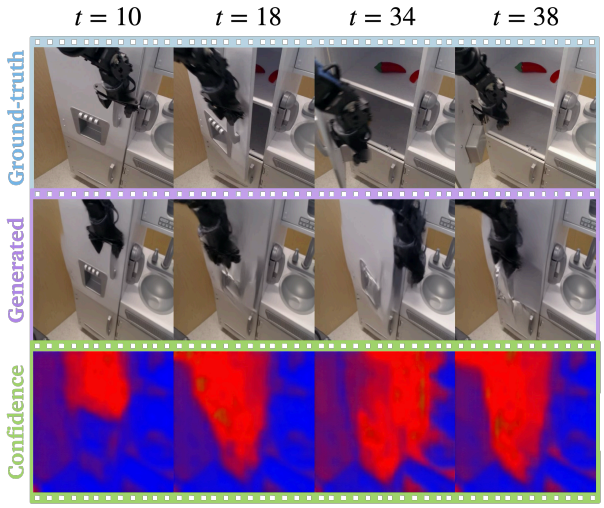

Figure 10 Occlusions. The internal content of the cabinet is occluded, resulting in high uncertainty of the interior area when the cabinet door is opened.

approaches the background object, which can be seen in the generated video. C^3 localizes this uncertainty, accurately delineating more confident video patches from less confident ones. Likewise, we see that the video model struggles to generate accurate videos under unseen lighting conditions, with an observable degradation in the video quality over time. Our method again captures the increasing uncertainty of the video model, both spatially and temporally. We provide visualizations of the results under other OOD conditions in Appendix E.

In Figure 11, we provide the reliability diagram of C^3 in OOD environments. We observe that our method remains well-calibrated with only a very small drop in calibration compared to its performance under nominal conditions. As stated in Section 4.2, C^3 achieves low calibration errors, with an ECE and MCE of 9.98e^{-2} and 1.71e^{-1}, respectively.

4.4 Evaluations on the DROID Dataset

We conduct additional experiments verifying the effectiveness of C^3 on the DROID dataset [10], which covers a much wider range of tasks and environments compared to the Bridge dataset. We train the CS-BC model on this dataset and evaluate the calibration and interpretability of its confidence estimates.

Calibration. In Figure 13, we show the reliability diagram of C^3 computed across the videos in the test dataset. Similar to the Bridge dataset, we observe a near-perfect calibration of the confidence estimates, with the top of the confidence bins closely tracing the diagonal, which represents the line of perfect calibration. Further, we compute the calibration errors across the test dataset with ECE and MCE values of 7.28e^{-2} and 1.74e^{-1}, respectively. We emphasize that the ECE is again close to the lower bound of the range of calibration errors, highlighting the well-calibrated nature of C^3. In summary, we find that our method is well-calibrated across a wide range of video prediction robotics problems, with broad amenability to multi-view camera inputs in diverse environments. Moreover, the results indicate that our method remains effective across different robot embodiments, considering the notable difference between the Panda robot used in the DROID dataset and the WidowX robot used in the Bridge dataset.

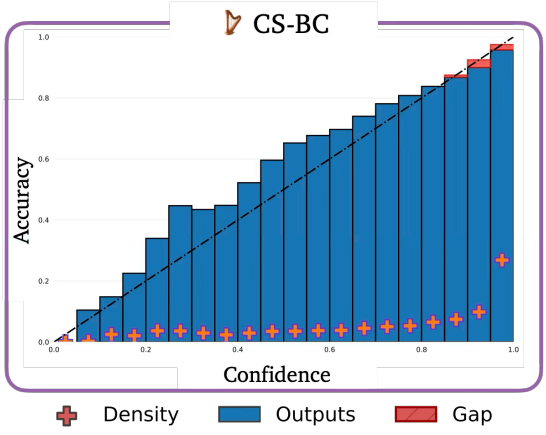

Figure 13 Reliability diagram on the DROID dataset. C^3 remains well-calibrated on the more diverse DROID dataset compared to the Bridge dataset.

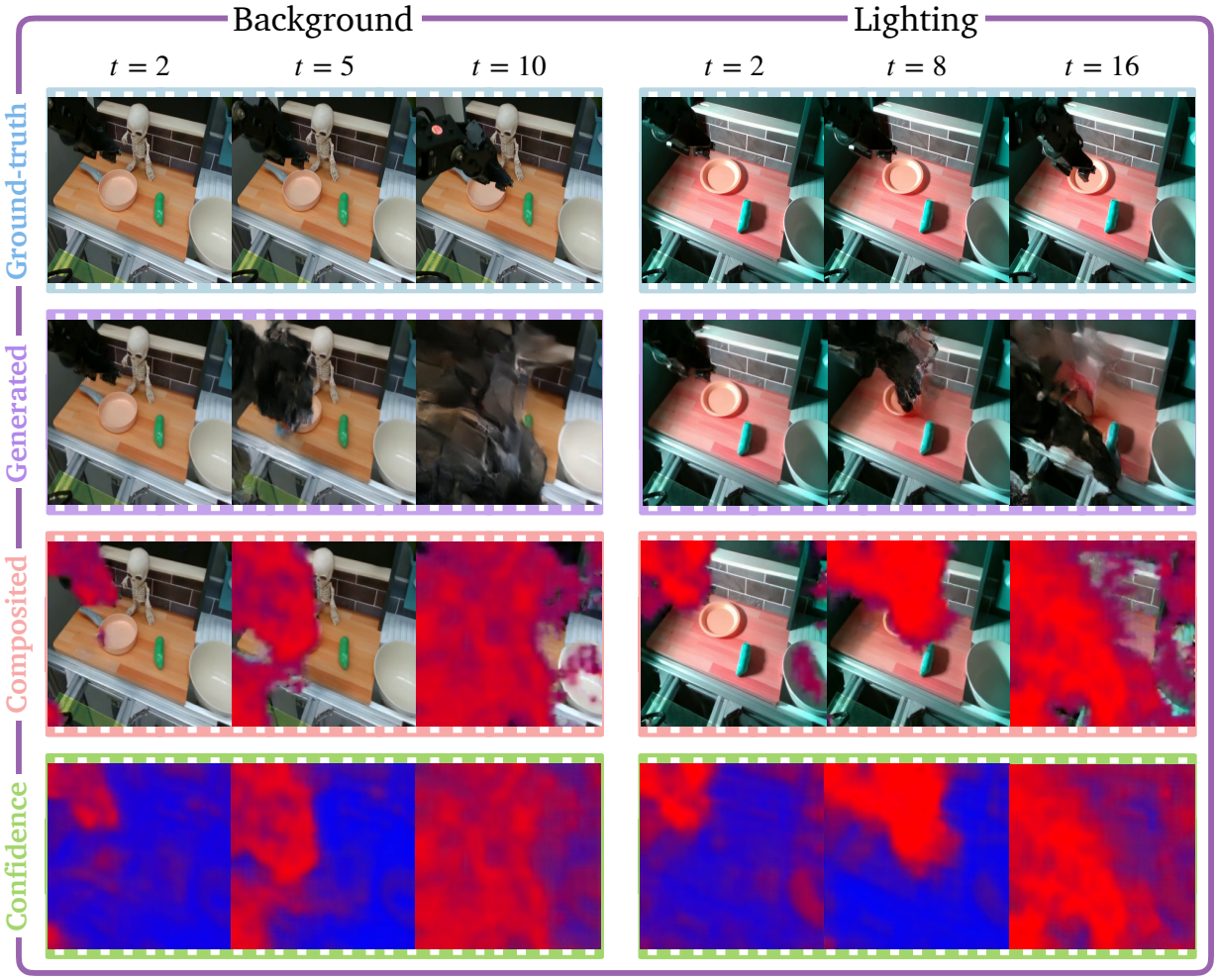

Figure 12 OOD detection. C^3 is able to accurately localize hallucinations (identified as regions of low confidence) in OOD scenarios where the model is presented with inputs outside of its training distribution.

Interpretability. Next, we examine the interpretability of the confidence estimates computed by C^3. First, we compute the correlation between the predicted confidence and the absolute error between ground-truth and generated latent videos. Similar to the Bridge dataset, a negative correlation between both quantities indicates greater interpretability of the uncertainty estimates. On the DROID dataset, we observe a negative correlation coefficient of -0.149 with a significance level greater than 99%, showing a desirable alignment between the model's estimated confidence and the observed accuracy of the generated videos. In other words, with C^3, the video model tends to be more uncertain about the generated videos when it's more likely wrong. Our results can be explained by the use of proper scoring rules in the model's loss function to achieve both calibration and accuracy.

We provide visualizations of the ground-truth and generated videos, along with the estimated confidence, highlighting the calibration and interpretability of C^3's uncertainty predictions.

From Figure 14, we observe that the video model is able to successfully generate multi-view videos, capturing the evolution of the task from two side-camera views (shown in the first-two rows) in addition to a wrist-camera view (shown in the third row). However, we also see a degradation in the relative quality of the generated videos compared to the Bridge dataset, reflecting the increased difficulty associated with multi-view video generation compounded with the greater diversity of the DROID dataset. This observation is particularly conspicuous in the wrist-camera view generated by the video model where the details of the scene quickly fade into a blurry background. Despite this degradation in video quality, C^3 is still able to produce interpretable,

calibrated uncertainty estimates at a fine-grained level, localizing non-confident regions of the video in each camera view. In particular, we see that in the right-camera view of the generated video (second row), our method captures hallucinations of the robot's gripper that appear in the video—the gripper morphs and elongates. Likewise, C^3 correctly identifies the inaccurate blurry background in the wrist-camera view (third row) as a region of high uncertainty. We provide additional results on the DROID dataset in Appendix F, showing C^3's ability to capture uncertainty due to object interaction and occlusions.

5 Conclusion

We present a method for calibrated controllable video synthesis that trains video models to know when they don't know. We use proper scoring rules as loss functions to achieve both accuracy and calibration in video generation. By quantifying uncertainty in the latent space, our proposed method overcomes computation challenges and training instability associated with pixel-space approaches. Furthermore, we map latent-space uncertainty to interpretable pixel-space confidence estimates that are well-aligned with human intuition. We show that our method is able to precisely localize hallucinations in generated videos. Likewise, we demonstrate the calibration of the confidence estimates computed by our method across different robot embodiments and tasks, and further show the effectiveness of C^3 in detecting out-of-distribution inputs at inference time.

6 Limitations and Future Work

Calibration in OOD Settings. Although we demonstrate the calibration of C^3 in out-of-distribution scenarios, its theoretical calibration guarantees only hold within the training data distribution through the use of proper scoring rules. Specifically, the diversity of the training data and the presence of a distribution shift at inference influences the observed calibration of the confidence estimates. We emphasize that this limitation is not unique to our approach. Nevertheless, we reiterate that our results show that our method produces calibrated uncertainty estimates even in OOD settings. Future work will explore training strategies to ensure broad coverage of the test distribution to ultimately minimize the effects of distribution shifts.

Long-Duration Video Generation. Long-duration temporal consistency of the confidence estimates computed by our method is limited by the history length of the conditioning inputs of the video model. With smaller historical contexts, our method may lose track of uncertain video patches over time. Long-duration video generation remains an open research problem, which will be explored in future work.

Efficient Training. Although our approach does not significantly increase training overhead, video models are generally expensive to train, especially state-of-the-art models. Future work will examine strategies to improve training efficiency without compromising video generation quality, for example, through lossless spatial compression.

Looking forward, we believe that rigorous uncertainty quantification is a key missing component of video generation models, and we expect that progress on this topic will enable trustworthy policy evaluation, data generation, and planning.

Acknowledgments

The authors were partially supported by the NSF CAREER Award #2044149, the Office of Naval Research (N00014-23-1-2148), and a Sloan Fellowship.

Figure 14 Hallucination. C^3 identifies hallucinations in the generated videos from the DROID dataset as areas of high uncertainty.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[2] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[3] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, et al. Open-sora 2.0: Training a commercial-level video generation model in 200 k. *arXiv preprint arXiv:2503.09642*, 2025.

[4] DeepMind. Veo-3: A text-to-video generation system with audio. Technical Report Tech Report, DeepMind / Google, 2025. Accessed: YYYY-MM-DD.

[5] Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation, 2025. URL https://arxiv.org/abs/2506.00613.

[6] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation, 2025. URL https://arxiv.org/abs/2510.10125.

[7] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

[8] Zhiting Mei, Ola Shorinwa, and Anirudha Majumdar. How confident are video models? empowering video models to express their uncertainty. *arXiv preprint arXiv:2510.02571*, 2025.

[9] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[10] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[11] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

[12] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016.

[13] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE international conference on computer vision*, pages 4463–4471, 2017.

[14] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[15] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016.

[16] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[17] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[18] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.

[19] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[20] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge?, 2018. URL https://arxiv.org/abs/1801.04406.

[21] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[22] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

[23] Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2318–2328, 2021.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[26] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[28] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[29] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[30] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[31] Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*, 2025.

[32] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

[33] Matthew Chan, Maria Molina, and Chris Metzler. Estimating epistemic and aleatoric uncertainty with a single model. *Advances in Neural Information Processing Systems*, 37:109845–109870, 2024.

[34] Lucas Berry, Axel Brando, and David Meger. Shedding light on large generative networks: Estimating epistemic uncertainty in diffusion models. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

[35] Gianni Franchi, Nacim Belkhir, Dat Nguyen Trong, Guoxuan Xia, and Andrea Pilzer. Towards understanding and quantifying uncertainty for text-to-image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8062–8072, 2025.

[36] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.

[37] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[38] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[39] Guillaume A Rousselet and Cyril R Pernet. Improving standards in brain-behavior correlation analyses. *Frontiers in human neuroscience*, 6:119, 2012.

[40] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[41] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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

[44] W Brier Glenn et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1): 1–3, 1950.

[45] X Chen, M Hong, S Liu, and R Sun. On the convergence of a class of adam-type algorithms for non-convex optimization. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[46] Meixuan He, Yuqing Liang, Jinlan Liu, and Dongpo Xu. Convergence of adam for non-convex objectives: Relaxed hyperparameters and non-ergodic case. *arXiv preprint arXiv:2307.11782*, 2023.

[47] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Appendix

A Video Diffusion/Flow-Based Models

Video diffusion models (and more generally flow-based video generation models) [1–3, 25, 40] have emerged as the dominant model architecture for controllable, high-fidelity video generation, capturing fine-grained scene detail and longer video durations compared to alternative architectures. Video diffusion models learn a data distribution $p_\theta(\mathbf{x})$ over video samples $\mathbf{x} \in \mathcal{U}$ by first destroying the underlying structure in the training data through a *forward* diffusion process and subsequently restoring the structure through a *reverse* diffusion process [27, 41], where $\mathcal{U}$ represents the space of videos and $\mathbf{x}$ consists of a sequence of video frames.

In denoising diffusion probabilistic models (DDPM) [27], the forward diffusion process adds Gaussian noise to the training data following a Markov chain, while the reverse diffusion process recovers the target data by denoising pure noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ through the procedure:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) \sim \mathcal{N}(\mathbf{x_t}; \sqrt{\bar{\alpha}_t}\mathbf{x_0}, (1 - \bar{\alpha}_t)\mathbf{I}), \tag{9}$$

where $\alpha_t \coloneqq 1 - \beta_t$ and $\bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s$, with $\Sigma_\theta(\mathbf{x}_t, t) = \beta_t \boldsymbol{I}$. For more stable training, the learning problem is reformulated as a noise prediction problem, where the learned model predicts $\boldsymbol{\epsilon}_t = (\sqrt{1 - \bar{\alpha}_t})^{-1}\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}$ and optimizes the loss function:

$$\mathcal{L}_\theta = \mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{x}_0}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta,t}(\mathbf{x}_t, t)\|_2^2\right], \tag{10}$$

via stochastic gradient descent.

In practice, we use denoising diffusion implicit models (DDIMs) [42], which generalize DDPMs to a non-Markovian forward diffusion process. In effect, DDIMs decouple the forward and reverse diffusion timesteps and leverage this feature to accelerate the video generation process during inference. Concretely, with DDIMs, we can train the diffusion model using longer forward diffusion timesteps and generate new videos using shorter reverse diffusion timesteps, significantly reducing the generation time and computation overhead. Rather than predicting the noise $\boldsymbol{\epsilon}$, we predict the velocity $\boldsymbol{v}$ with a diffusion transformer.

We train the video model using a robot dataset $\mathcal{D} = \{((I_{j,t}, I_{j,t+1}, a_{j,t}),\ \forall t \in [T_j]),\ j = 1, \ldots, N\}$ consisting of N trajectories. Each data sample consists of the current observation $I_{j,t} \in \mathbb{R}^{H \times W \times C}$, the next observation $I_{j,t+1} \in \mathbb{R}^{H \times W \times C}$, and the corresponding action $a_{j,t} \in \mathbb{R}^m$, for the j'th trajectory of length T_j.

At inference, we sample new video frames using:

$$\mathbf{x}_{t-1} \coloneqq \sqrt{\bar{\alpha}_{t-1}}\ \tilde{\mathbf{x}}_0(t) + \sqrt{1 - \bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\tilde{\mathbf{x}}_0(t)}{\sqrt{1 - \bar{\alpha}_t}}\right), \tag{11}$$

where $\tilde{\mathbf{x}}_0(t) = \sqrt{\bar{\alpha}_t}\ \mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\ \boldsymbol{v}_\theta(\mathbf{x}_t, \mathbf{c})$ denotes the value of $\mathbf{x}_0$ predicted at timestep t and $\mathbf{c}$ denotes the action and timestep embeddings.

Remark 1 (Velocity-space Accuracy)**.** *The distance function in Equation (3) requires executing the reverse diffusion process to generate the latent video* $\mathbf{x}$*, which is computationally expensive during training. To address this challenge, we express the distance function in terms of the predicted and ground-truth velocities,* $\mathbf{v}_\theta$ *and* $\mathbf{v}^\star$*, respectively. By manipulating Equation (11) algebraically, we derive the relation:*

$$\mathbf{d}(\mathbf{x}, \mathbf{x}^\star) = \left|\sqrt{\bar{\alpha}_t(1 - \bar{\alpha}_{t-1})} - \sqrt{\bar{\alpha}_{t-1}(1 - \bar{\alpha}_t)}\ \right| \mathbf{d}(\mathbf{v}, \mathbf{v}^\star), \tag{12}$$

with the corresponding boolean function **acc** *given by:*

$$\mathbf{acc}(\mathbf{v}, \mathbf{v}^\star) \coloneqq \mathbf{d}(\mathbf{v}, \mathbf{v}^\star) \leq \varepsilon_v, \tag{13}$$

where $\varepsilon_v = \frac{\varepsilon}{\left|\sqrt{\bar{\alpha}_t(1-\bar{\alpha}_{t-1})} - \sqrt{\bar{\alpha}_{t-1}(1-\bar{\alpha}_t)}\ \right|}$.

Notably, quantifying accuracy in the velocity-space only requires a simple linear transformation. Given **acc** *in Equation (13), we train* ϵ_θ *and* $\mathbf{f}_\phi$ *without the reverse diffusion process for greater training efficiency.*

B Proper Scoring Rule

For a random variable Y following the distribution $P(Y)$, a scoring rule S evaluates a prediction q of the probability distribution of Y by assigning a real-valued score, which could be interpreted as a penalty or reward, providing a measure of the quality of the predicted distribution. A scoring rule is proper if:

$$\mathop{\mathbb{E}}_{y\sim p(Y)} S(p(Y), y) \leq \mathop{\mathbb{E}}_{y\sim p(Y)} S(q, y), \tag{14}$$

for all q [43]. Note that a proper scoring rule assesses a larger penalty for all predictions that are not equal to the underlying probability distribution of Y. Intuitively, the proper score is minimized when the predicted distribution q matches the true probability of Y. The scoring rule is strictly proper if equality holds in Equation (14) if and only if $p(Y) = q$. Some examples of proper scoring rules include the Brier Score (BS) [44], Cross Entropy (CE), and Binary Cross Entropy (BCE), discussed in the paper.

In addition, we note that the expected calibration error (ECE) and maximum calibration error (MCE) are the standard metrics used to measure deviation from perfect calibration:

$$\text{ECE} \coloneqq \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \tag{15}$$

$$\text{MCE} \coloneqq \max_{m\in\{1,\ldots,M\}} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \tag{16}$$

where B_m represents bin m with cardinality $|B_m|$, and n represents the total number of samples across all bins.

C Proofs

Proposition 1 (Uncertainty Decomposition)**.** *Given the input actions and video frames, the predicted confidence* $\hat{\mathbf{q}}$ *provides a calibrated measure of uncertainty of the video diffusion model in the generated video, provided that* ϕ *converges to an optimal solution.*

Proof. The proof of this proposition follows immediately from the definition of a proper scoring rule in Equation (14). Hence, from Equation (14), the minimum of the right-hand side (RHS) of Equation (14) is unique and in particular, is attained when $\mathbf{q} = p(\mathbf{y})$. By optimizing ϕ to minimize the RHS of Equation (14), we have that $\hat{\mathbf{q}} \to p(\mathbf{y})$, assuming convergence, which is guaranteed under relatively weak conditions [45, 46]. Further, we have that:

$$\mathbb{P}[\mathbf{Y} = \mathbf{1} \mid \mathbf{Q} = \hat{\mathbf{q}}] = \mathbb{E}[\mathbf{Y} \mid \mathbf{Q} = \hat{\mathbf{q}}] = \hat{\mathbf{q}}, \tag{17}$$

where the last equality follows from the fact that $\hat{\mathbf{q}} = p(\mathbf{y})$, upon convergence. The result in Equation (17) indicates calibration of the predicted confidence $\hat{\mathbf{q}}$. □

D Additional Results on Interpretability on the Bridge Dataset

We provide additional results showing that C^3's confidence estimates are interpretable at low error thresholds. In Figure 15, we see that the video model is uncertain about the accuracy of the locations corresponding to the background (indicated by the red regions), but is notably confident that the pixels showing the robot free-space motion and object interaction are inaccurate. This finding aligns with human intuition: the video background is more consistent with the ground-truth and thus is more likely to lie close to the accuracy-defining margin, leading to doubts about its ultimate accuracy. In contrast, the video model often hallucinates the motion of the robot, resulting in pixels that are further away from the ground-truth pixels and ultimately leading to greater confidence in the inaccuracy of these pixels.

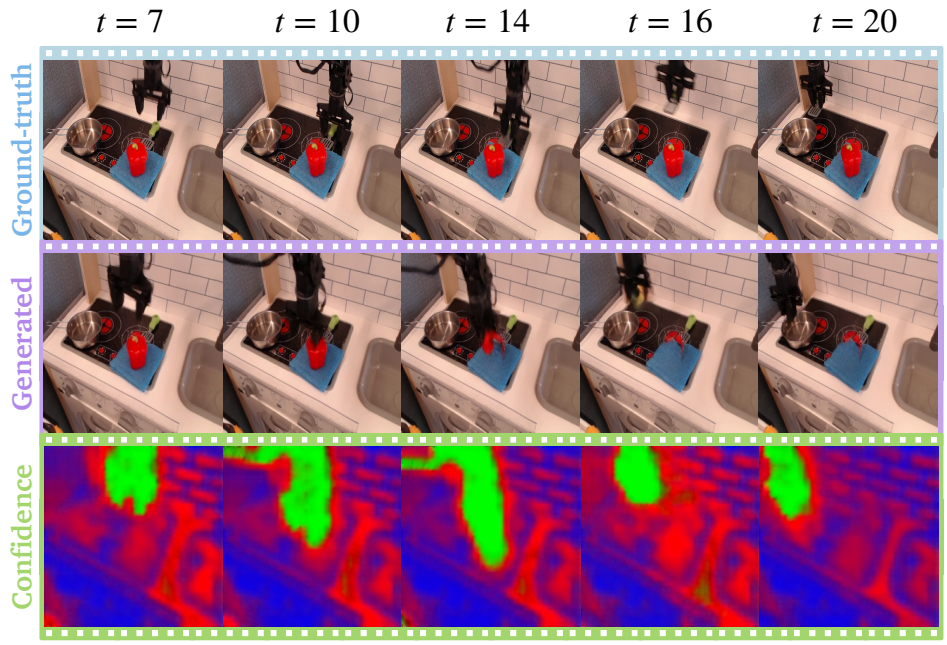

Figure 15 Confidence heatmap visualization. We show the ground-truth and generated video frames, along with the composited confidence map and the full confidence map at lower error thresholds ε_v. At low error thresholds, the video model is uncertain about the *accuracy* of the background but confident in the *inaccuracy* of the robot's motion.

E Detecting OOD Inputs at Inference

We provide additional visualizations of the ground-truth and generated videos and confidence predictions under different OOD conditions in Figures 16 and 17. In cluttered environments, the video model fails to accurately predict the interaction between the robot and the objects in the scene, which C^3 correctly identifies. Lastly, when an unfamiliar object is attached to the robot end-effector, the video model becomes uncertain about the dynamics of the robot, leading to hallucinations in the generated video. Our method identifies these hallucinations as regions of low confidence, while marking other unaffected areas as regions of high confidence. Under extreme lighting conditions, the video model hallucinates a recoloring of the scene in an attempt to match the training data distribution. C^3 detects the low confidence of the model, identifying regions with edited colors as areas of high uncertainty.

F Experiment Results on the DROID Dataset

We provide additional visualizations of the ground-truth and generated videos, along with the estimated confidence, showing the interpretability of the our method's uncertainty estimates in the DROID dataset.

In Figure 18, we show that our proposed method is able to capture uncertainty associated with object interactions. In the ground-truth video, the robot attempts to pick up a spoon; however, the video model is unable to correctly predict the robot and object dynamics, resulting in hallucinations which are visible in the wrist-camera view. Nonetheless, C^3 is able to highlight the different sources of uncertainty in the generation task, spanning uncertainty from the robot's motion, object interaction, and scene background.

In Figure 19, we visualize a scenario where much of the robot arm is originally occluded from the two side-camera views. As the ground-truth actions bring the arm into view, the video model is unable to accurately predict the previously invisible part of the robot arm in the generated videos, resulting in blurry predictions. Our method accurately identifies these regions as highly uncertain. Moreover, in the wrist-camera view, the video model fails to predict the relatively complex layout of the stovetop. Again, C^3 highlights these regions in the video as areas of high uncertainty.

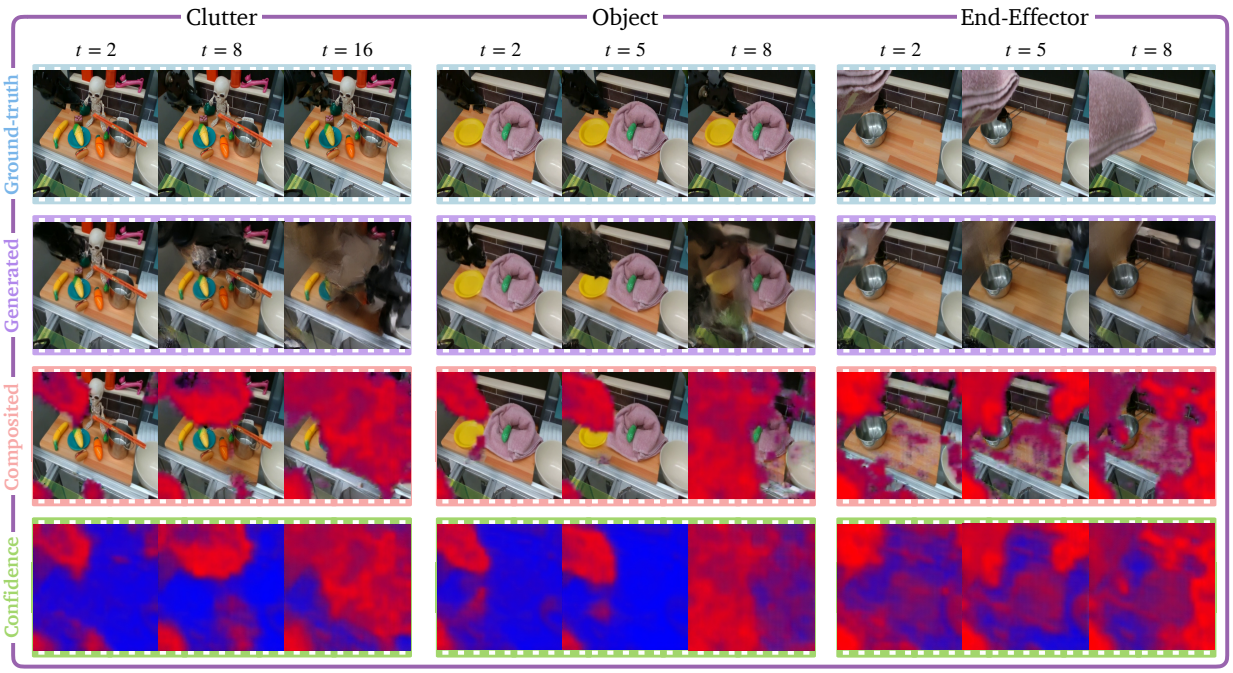

Figure 16 OOD detection. We show additional OOD settings, demonstrating the interpretability of C^3's confidence estimates.

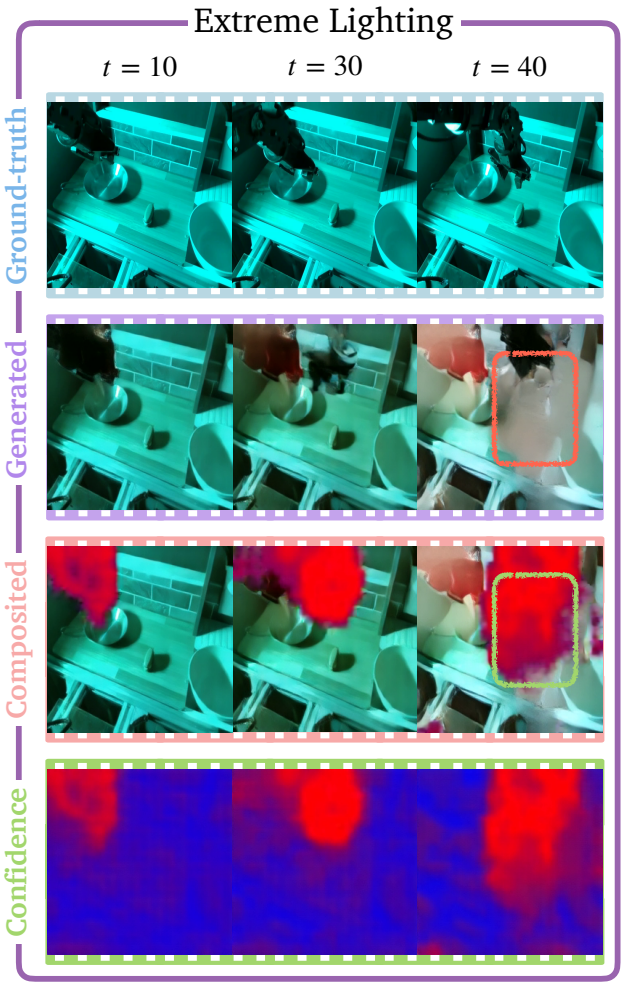

Figure 17 Extreme Lighting condition. C^3 identifies artificially edited pixels where the video model attempts to reset the lighting of the scene to match the training data distribution.

Figure 18 Object Interaction. C^3 localizes uncertainty from multiple sources, e.g., from the robot's motion, object dynamics, and scene background.

Figure 19 Occlusions. Our method captures the uncertainty associated with the robot parts that were occluded in the first frame.

G Additional Ablations

We perform additional ablations to study the effects of design choices on the scoring rule, diffusion forcing, and end-to-end training without the stop-gradient operation, with respect to calibration of the confidence estimates computed by C^3.

Proper Scoring Rules. We examine the calibration of video models trained with the binary cross-entropy loss function and the Brier loss function. Since both loss functions are proper scoring rules, one would expect both models to achieve similar calibration performance. We train two variants of the CS-BC model using these loss functions and observe similar calibration levels between both models, in line with the preceding expectation. The results indicate a negligible difference in the ECE of about $3e^{-4}$ and a similarly negligible difference in the MCE of about $6e^{-3}$.

Further, we visualize the reliability diagrams associated with each model in Figure 20. We see very similar trends in the reliability diagrams across all confidence bins. Specifically, the models are well-calibrated but more conservative when unsure about the accuracy of the generated video. These findings underscore the general applicability of our proposed framework to strictly proper scoring rules.

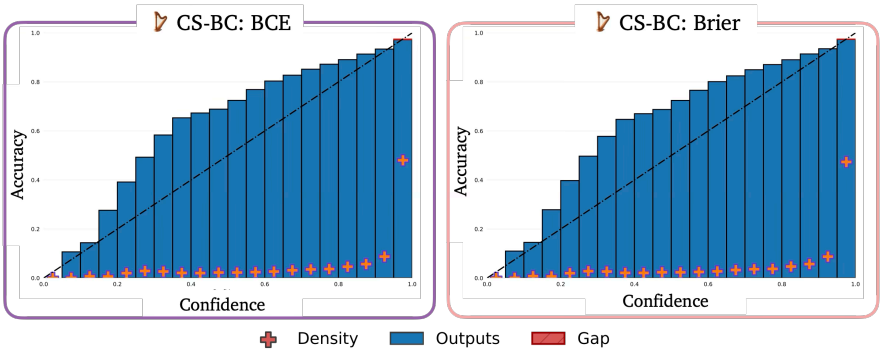

Figure 20 Reliability diagram for ablation on proper scoring rules. C^3 remains well-calibrated with the BCE and Brier scores.

Diffusion forcing. We explore diffusion forcing in generating the confidence maps during video prediction and compute the ECE and MCE to evaluate calibration. We visualize the reliability diagrams of the CS-BC model with and without diffusion forcing in Figure 21, showing that diffusion forcing degrades the calibration of C^3. We hypothesize that this observation could be due to the effects of the recurrence in diffusion forcing, which leads to a notable increase in the conservatism of the confidence estimates. With diffusion forcing, the ECE rises to about $3.3e^{-1}$ with an associated increase in the MCE to about $5.54e^{-1}$. A comprehensive analysis of the effects of diffusion forcing on confidence prediction lies beyond the scope of this paper; hence, we leave it to future work.

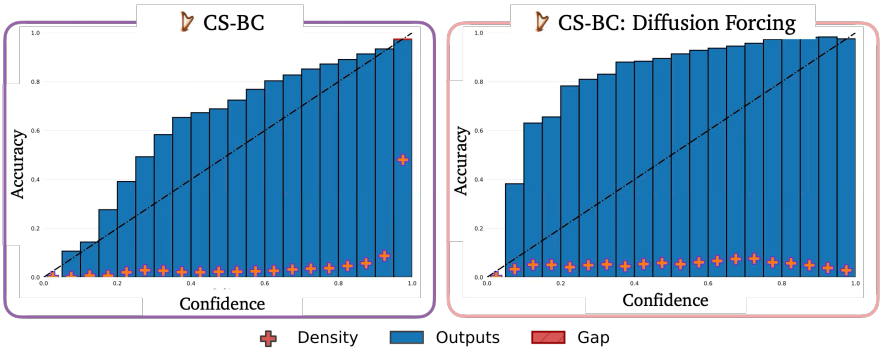

Figure 21 Reliability diagram for ablation on diffusion forcing. Diffusion forcing increases underconfidence of C^3, degrading calibration.

End-to-End Training without Stop-Gradient. We examine the calibration of C^3 when training the video model and the UQ probe $\mathbf{f}_\phi$ end-to-end without a stop-gradient operator between both models. In other words, we update the parameters of the video model with the gradient of $\mathbf{f}_\phi$, assessing the existence of any training synergies from joint training. We train the CS-BC model with and without the stop-gradient operator and find no significant difference in the calibration of the resulting confidence estimates. We observe a difference of about $5e^{-3}$ and $3e^{-3}$ in the ECE and MCE, respectively, highlighting that both variants achieve the same calibration level. However, we note that backpropagating the gradient of $\mathbf{f}_\phi$ through the video model incurs additional computation overhead. As a result, the stop-gradient operation provides a computational edge, especially in large video models with billions of parameters.

H Additional Implementation Details

We implement the video generation model using a diffusion transformer architecture with 49 transformer layers, each with 4 heads and an embedding dimension of 512. We use the Stable Diffusion VAE [47] for encoding the videos into the latent space, extracting video patches with no temporal compression. Using a learning rate of $1e^{-5}$ with a cosine decay scheduler, we train the video model for 50k iterations with a batch size of 4 for the Bridge dataset and a batch size of 2 for the DROID dataset, on 8 NVIDIA L40 GPUs. We use an input video resolution of 256×256 for both datasets. We stack the multi-view camera inputs in the DROID dataset to construct a single video frame as input.

We train on the entire train split of the Bridge dataset; however, given the large size of the DROID dataset, we only train on a subset (TRI), covering both success and failure videos, across a broad range of tasks and environments. When training the CS-BC model, we randomly sample thresholds from a discrete set of 28 threshold values constructed linearly from 0.1 to 1 with adaptive (denser) spacing at lower thresholds between 0.1 and 0.3 to more effectively capture the fine-grained signal existent in this subrange. We define the output bins of the MCC model using the same set of threshold values. We evaluate the video models on 110 and 83 trajectories in the test split of the Bridge and DROID datasets, respectively.

When evaluating the CS-BC model at inference time, we use 10 linearly-spaced values of ε_v ranging from 0.1 to 1.0. Except otherwise noted in our qualitative evaluations, we visualize the confidence predictions at the midpoint of the range corresponding to threshold values of 0.5 or 0.6. We compute the aggregate calibration errors for each model using standard implementations over 20 bins and compute the video accuracy using the ℓ_1 loss.