Zero-shot Safety Prediction for Autonomous Robots with Foundation World Models

Zhenjiang Mao[1], Siqi Dai[1], Yuang Geng[1], and Ivan Ruchkin[1]

***Abstract*— A world model creates a surrogate world to train a controller and predict safety violations by learning the internal dynamic model of systems. However, the existing world models rely solely on statistical learning of how observations change in response to actions, lacking precise quantification of how accurate the surrogate dynamics are, which poses a significant challenge in safety-critical systems. To address this challenge, we propose foundation world models that embed observations into meaningful and interpretable latent representations. This enables the surrogate dynamics to directly predict interpretable future states by leveraging a training-free large language model. In two common benchmarks, this novel model outperforms standard world models in the safety prediction task and has a performance comparable to supervised learning despite not using any data. We evaluate its performance with a more specialized and system-relevant metric by comparing estimated states instead of aggregating observation-wide error.**

I. Introduction

A world model represents an understanding of how a robotic system works by learning how observations change with corresponding actions [1]. For example, it can describe how the image seen by a legged robot changes after taking several steps. Originally, world models were introduced to address data insufficiency when training reinforcement-learning controllers, which were limited by scarce real-world data [2], [3]. This training typically requires the agent to operate in a real environment to gather extensive data, which makes it difficult and costly to explore diverse scenarios. As shown in Fig. 1, a world model can learn the behavior of both the true dynamical model and observation models and thus become a surrogate of the real world. Some world models also learn rewards to support controller training. Not only do world models offer a new way for constructing better controllers by building a new generative *world*, the surrogate dynamics also support quantifying the competence [4] and safety predictions [5] for highly critical autonomous robots.

In the past, researchers primarily focused on training better controllers with world models, with insufficient attention given to how accurate the *world* of world models is — a crucial aspect for safety-critical robots. Typically, a world model does not directly predict observations, such as images from cameras or other sensor data, due to the constraints on time and computational resources. Instead, as shown on the left of Fig. 2, it extracts useful features into latent representations with an encoder, which is part of a Variational Autoencoder (VAE) [6], and then predicts based on these representations.

[1]Zhenjiang Mao, Siqi Dai, Yuang Geng, and Ivan Ruchkin are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611, USA, {z.mao, dais, yuang.geng, iruchkin}@ufl.edu

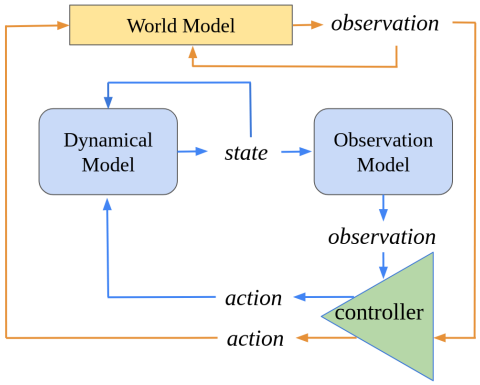

Fig. 1. A dynamical model (blue flow) vs. a world model (orange flow).

Usually, the performance of the world model is evaluated with the Mean Square Error (MSE) between the predicted observation and ground truth; this metric considers only the aggregate performance of the reconstruction — and lacks the examination of the fine-grained and meaningful details of the prediction. This coarse checking does not distinguish slight but critical differences (e.g., if a car's position shifts a few pixels, it may lead to a collision) that may cause devastating outcomes in safety predictions. Also, the latent representations do not have a physical meaning in standard models and cannot be used to evaluate whether a predicted latent state is safe or has other important characteristics. As a result, it becomes necessary to develop an additional classifier to check such critical aspects, akin to a safety check as discussed in our earlier work [5], which requires more data, adds noise, and may suffer from distribution shift in predicted observations (compared to the real sensor data).

The rise of foundation models provides an opportunity to create meaningful representations with a zero-shot segmentation of observed images. Not only the interpretable representations can simplify the prediction and also the whole world model can also be implemented with a training-free architecture using Large Language Models, eliminating the need to collect and label training data.

This paper proposes *foundation world models* that further reduce data requirements by using foundation models in two key elements of world models. First, to obtain interpretable latent representations, we use the *Segment Anything Model* (SAM) [7] to get the pixel positions of all objects in the observation. The important characteristics of latent states, in particular their safety (e.g., whether a collision has occurred or not), can be calculated based on these representations. Second, we predict the future position of these objects with a *Large Language Model* (LLM). In addition, we introduce a

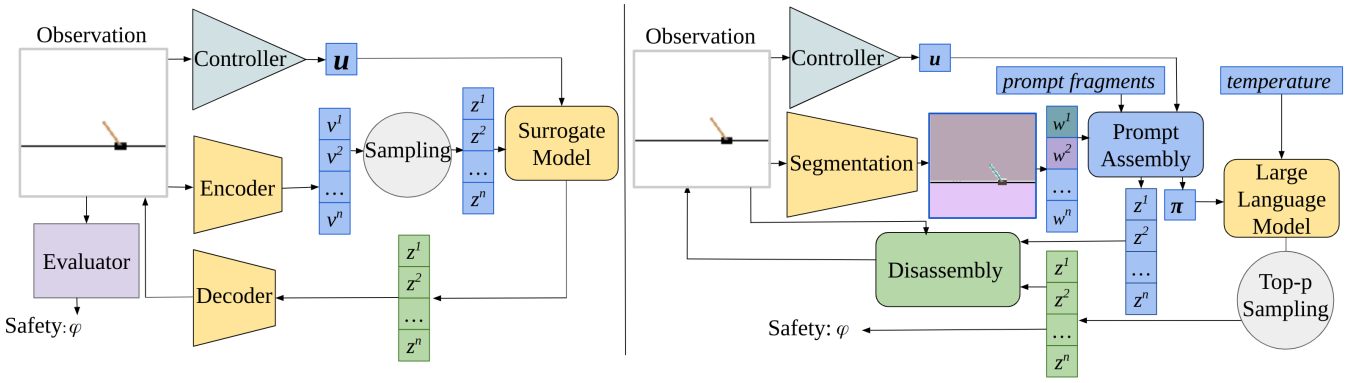

Fig. 2. The structure of an existing world model (left) and the proposed foundation world model (right).

more focused metric for the accuracy evaluation of predicted state. This metric helps us evaluate the quality of surrogate dynamics in a system-specific way.

Our evaluation on two common simulated benchmarks demonstrates the value of foundation world models. Our foundation model not only demonstrates superior state prediction based on the newly proposed metric — but also excels in safety predictions. We experiment with several baseline approaches, evaluation metrics, and different Large Language Models for latent prediction.

In summary, this paper makes three contributions:

1) A training-free world model that combines foundation models with interpretable embedding and overcomes the distribution shift of the predicted observation, which occurs in standard world models.
2) A segmentation-based metric for the accuracy of the surrogate dynamic prediction by quantifying the deviations of each object in the observation.
3) An experimental study of safety prediction where foundation world models show better performance despite not using any training data, compared to the existing world model and supervised learning methods.

Sec. II introduces the details of world models and our problem description. Sec. III and Sec. IV describe the foundation world models and the results of our experiments. In Sec. V and Sec. VI, we will review the related work and discuss the conclusion and future work.

II. Preliminaries and Problem Statement

A. Observation Prediction with World Models

Consider a system s with dynamical model f in state $\boldsymbol{x}_t$, where action $\boldsymbol{u}_t$ is generated by an image-based controller h: $h(\boldsymbol{y}_t) = \boldsymbol{u}_t$. Each time step, the dynamics generates the state for the next time step: $f(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{x}_{t+1}$. The system also includes an observation model g that converts the states $\boldsymbol{x}$ to observations $\boldsymbol{y}$: $g(\boldsymbol{x}_t) = \boldsymbol{y}_t$.

The left side of Fig. 2 illustrates the architecture and operational workflow of the standard world model, composed of a decoder and an encoder [6]. The *encoder* $\boldsymbol{E}$ extracts useful features from images or sensor data $\boldsymbol{y}_t$ at time t to build a distribution over latent vectors: $\boldsymbol{E}(\boldsymbol{y}_t) = \boldsymbol{v}_t = [\boldsymbol{v}_t^1, \boldsymbol{v}_t^2, ..., \boldsymbol{v}_t^n]$, where each p^i is a one-dimensional Gaussian distribution: $N(\mu_t^i, \sigma_t^i)$. After sampling a latent vector $\boldsymbol{z}_t \sim \boldsymbol{v}_t$, the *surrogate dynamical model* $\boldsymbol{P}$ (or simply surrogate model) predicts the future latent representation based on the past data: $\boldsymbol{P}(\boldsymbol{z}_{t:t+m}, \boldsymbol{u}_{t:t+m}) = \boldsymbol{z}_{t+m+1}$. The *decoder* $\boldsymbol{D}$ reconstructs a predicted observation $\hat{\boldsymbol{y}}$ from the latent representation: $\boldsymbol{D}(\boldsymbol{z}_{t+1}) = \hat{\boldsymbol{y}}_{t+1}$. The VAE and the surrogate model are trained sequentially. The optimization target of the VAE is to minimize the reconstruction error between the true image $\boldsymbol{y}_t$ and the reconstructed image $\hat{\boldsymbol{y}}_t$, as well as the KL divergence between the prior distribution and the latent distribution [6]. In the surrogate model training, the mean squared error (MSE) is commonly employed as the loss function to minimize the error between the predicted image and the true image. In summary, the goal of a world model is to learn the following distribution:

$$p(\boldsymbol{y}_{t+m+1} \mid \boldsymbol{y}_{t:t+m}; \boldsymbol{u}_{t:t+m}),$$

where m is the input length of known observations. Thus, the world model $\boldsymbol{W}$ can be expressed as follows:

$$\boldsymbol{W}(\boldsymbol{y}_{t:t+m}, \boldsymbol{u}_{t:t+m}, \epsilon) = \hat{\boldsymbol{y}}_{t+m+1},$$

where ϵ is the source of randomness in the latent sampling.

There is no single agreed-upon metric to evaluate the performance of a world model. In various applications of world models, the controller-training tasks are usually judged on how well the resulting controller performs compared with traditional methods [1], [3] in terms of reward. Traditionally, evaluation metrics like MSE are commonly adopted as the standard measurement criteria for quantifying the predictive capacity of world models. Accordingly, we compute the MSE of the world model's predicted observation in a pixel-wise manner:

$$\text{MSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{LW} \sum_{i=1}^{L} \sum_{j=1}^{W} (\hat{\boldsymbol{y}}^{(i,j)} - \boldsymbol{y}^{(i,j)})^2,$$

where L and W are the length and width of the image and $\boldsymbol{y}^{(i,j)}$ is the pixel value at the position (i, j).

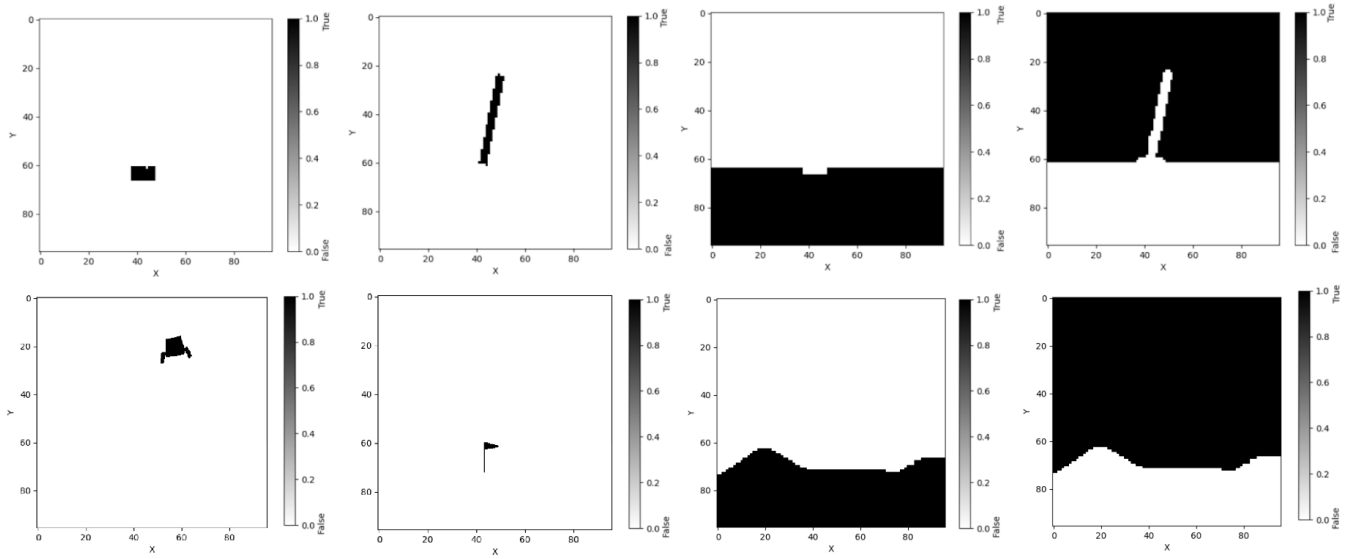

Fig. 3. An example of segmentation matrices. Upper, from left to right: segmentation of the cart, pole, lower background, and upper background. Lower, from left to right: segmentation of the lander, lander point flag, lower background, and upper background.

We also consider another standard metric for visual prediction — the *Structural Similarity Index Measure* (SSIM):

$$\text{SSIM}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{(2\mu_{\hat{\boldsymbol{y}}}\mu_{\boldsymbol{y}} + C_1)(2\sigma_{\boldsymbol{y}\hat{\boldsymbol{y}}} + C_2)}{(\mu_{\boldsymbol{y}}^2 + \mu_{\hat{\boldsymbol{y}}}^2 + C_1)(\sigma_{\boldsymbol{y}}^2 + \sigma_{\hat{\boldsymbol{y}}}^2 + C_2)},$$

where μ and σ^2 are the mean and variance of an image over all pixels, $\sigma_{\boldsymbol{y}\hat{\boldsymbol{y}}}$ is the covariance between images, and C_1 and C_2 are stability constants used to avoid close-to-zero values in the denominator. This metric takes brightness, contrast, and structural differences into account. However, MSE and SSIM are aggregation metrics because they cover and summarize the whole image but ignore the more important objects in the images. The next section will present our improvement over these metrics.

B. Object-based Prediction Metric

Our insight, unused in earlier world models, is that the observation can be split into multiple meaningful objects by some segmentation algorithm Ω: $\Omega(\boldsymbol{y}) = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ...]$. Each $\boldsymbol{\omega}$ is a segmentation matrix with $True$ and $False$ pixels with the same size as $\boldsymbol{y}$, as shown in Fig. 3 for two example systems. The $True$ elements reflect the position of the segmented object. These objects in the observation are semantically important elements of the world that are assumed to have a causal relation to future positions. The prediction of the objects' positions will evaluated separately, unlike e.g. the image-wise MSE that takes all pixels into account and is affected by inconsequential entities like static objects/background that occupy most of the observation frame.

To tailor the evaluation to each object $\boldsymbol{\omega}$, we will calculate its *centroid* $\boldsymbol{\omega}^c = (\boldsymbol{\omega}_x^c, \boldsymbol{\omega}_y^c)$ (i.e., its center of mass) as:

$$\boldsymbol{\omega}_x^c = \frac{1}{LW}\sum_{i=1}^{L}\sum_{j=1}^{W} j \cdot \boldsymbol{\omega}[i][j]$$

$$\boldsymbol{\omega}_y^c = \frac{1}{LW}\sum_{i=1}^{L}\sum_{j=1}^{W} i \cdot \boldsymbol{\omega}[i][j]$$

Each prediction $\hat{\boldsymbol{\omega}}$ of a true object $\boldsymbol{\omega}$ will be quantified using the **centroid distance** (CD) defined with some norm $\mathcal{L}$, which can be for instance an L1 or L2-norm:

$$\text{CD}(\boldsymbol{\omega}, \hat{\boldsymbol{\omega}}) = \|\hat{\boldsymbol{\omega}}^c - \boldsymbol{\omega}^c\|_{\mathcal{L}}$$

For evaluating the image-level error between a prediction $\hat{\boldsymbol{y}}$ and a ground truth $\boldsymbol{y}$, we implement the segmentation algorithm to split the image into several parts: $\Omega(\boldsymbol{y}) = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ..., \boldsymbol{\omega}_n]$, $\Omega(\hat{\boldsymbol{y}}) = [\hat{\boldsymbol{\omega}}_1, \hat{\boldsymbol{\omega}}_2, ..., \hat{\boldsymbol{\omega}}_n]$. Then, we get a vector of CDs $[\text{CD}(\boldsymbol{\omega}_1, \hat{\boldsymbol{\omega}}_1), \text{CD}(\boldsymbol{\omega}_2, \hat{\boldsymbol{\omega}}_2), ..., \text{CD}(\boldsymbol{\omega}_n, \hat{\boldsymbol{\omega}}_n)]$ that precisely reflects how well each object is predicted, rather than aggregating the errors of unrelated pixels. We leave the cases of missing and ghost objects for future work.

C. Safety Prediction Problem

The *safety predicate* φ is a binary function of state $\boldsymbol{x}_t$ that determines the safety of the system at time t: $\varphi(\boldsymbol{x}_t)$. For example, it can indicate if the robot is dangerously close to an obstacle. The safety prediction problem [5] is defined as follows:

> Given horizon $k > 0$, a safety predicate φ, a sequence of m observations and actions from a known a controller h: $[\boldsymbol{y}_t, \boldsymbol{y}_{t+1}, ..., \boldsymbol{y}_{t+m}]$, $[\boldsymbol{u}_t, \boldsymbol{u}_{t+1}, ..., \boldsymbol{u}_{t+m}]$ from an unknown system s, determine whether the system satisfies the following logical formula for the future time moments:
>
> $$\varphi(\boldsymbol{x}_{t+m}) \wedge \ldots \wedge \varphi(\boldsymbol{x}_{t+m+k})$$

Since safety prediction is binary, we use the F1 score to balance the evaluation of precision and recall. False-positive safety predictions are dangerous in safety-critical systems, so the false positive rate (FPR) is also taken into account.

A significant challenge in standard world models is that the safety is not straightforward to infer from high-dimensional observations, so it necessitates an extra learning-based safety evaluator (often a Convolutional Neural Network, CNN) shown on the left of Fig. 2. This CNN safety evaluator (as well as the controller) can suffer from distribution shift due to distorted predictions $\hat{\boldsymbol{y}}$ of standard world models originating in the decoder. Specifically, the conditional distribution of safety is the same for training and test datasets $p_{train}(\varphi(\boldsymbol{x}) \mid \boldsymbol{y}) = p_{test}(\varphi(\boldsymbol{x}) \mid \boldsymbol{y})$, but the distribution of observations is shifted: $p_{train}(\boldsymbol{y}) \neq p_{test}(\boldsymbol{y})$. This kind of distribution shift is covariate shift [8], which influences the accuracy of the CNN evaluator.

Algorithm 1 Prompt assembly for state prediction

Input: A series of observation $\boldsymbol{y}_{t:t+m}$, a segmentation model Ω and corresponding actions $(U = \boldsymbol{u}_t, \boldsymbol{u}_{t+1}, ..., \boldsymbol{u}_{t+m})$, and prompt fragment R: instruction fragment r_1, input assembly fragment r_2, and prediction fragment r_3.
Output: A full prompt π and predicted states $\boldsymbol{z}$.
FUNCTION `Assemble`$(\boldsymbol{y}_{t:t+m}, \Omega, U, R)$:

$\boldsymbol{z} \leftarrow [\,]$ ▷ Initialize an empty list
$\pi \leftarrow r_1$ ▷ Add instructions at the beginning
for i from t to $t+m$ **do**
  $\boldsymbol{\omega}_i \leftarrow \Omega(\boldsymbol{y}_i)$
  for j from 1 to $\text{len}(\boldsymbol{\omega}_i)$ **do**
    $\pi \leftarrow \pi + r_2$ ▷ Add assembly prompt
    $\boldsymbol{z}_i^j \leftarrow (\boldsymbol{\omega}_i^j)^c$ ▷ Get the centroid
    $\pi \leftarrow \pi + \boldsymbol{z}_i^j$ ▷ Add the centroid
    $\pi \leftarrow \pi + \boldsymbol{u}_M$ ▷ Add action
    $\boldsymbol{z} \leftarrow \boldsymbol{z} + \boldsymbol{z}_i^j$
  end for
end for
$\pi \leftarrow \pi + r_3$ ▷ Add prediction order and specify the required output format
return $\pi, \boldsymbol{z}$

III. Foundation World Models

We propose *foundation world models* that incorporate (a) a pre-trained foundation segmentation model into our proposed architecture as an encoder and (b) a large language model as a latent predictor. Specifically, we adopt the Segment Anything model (SAM) [7] as the segmentation model to split the observation into several objects: $\Omega(\boldsymbol{y}) = [\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ...]$. The extracted centroids $\boldsymbol{\omega}^c$ are used as latent representations $\boldsymbol{z}$, which are causally informative as future object positions are caused (in part) by past object positions. Combining the representations $\boldsymbol{z}$ with actions $\boldsymbol{u}$ and *prompt fragments* $R = \{r_1, r_2, r_3\}$, we adopt the Algorithm 1 to form[1] a complete prompt π. A Large Language Model takes prompt π and returns a formatted prediction of $\boldsymbol{P}(\pi) = \hat{\boldsymbol{z}}$. Three types of pre-prompts in R are used to assemble a full prompt: the first r_1 is to describe the whole task, e.g., *"Suppose I have a sequence of actions and states of a system, please predict the next step's state given the following information."* Next, there's a loop to fuse the state description r_2 and the corresponding values *"The time, states and the action for this step are:'*. We will repeat to add r_2 for m times to add all the input information. Finally, we add the state prediction instruction r_3: *"Can you predict the state for the next moment? Please only give me your prediction values as a list"* at the end of the prompt to form the complete π.

To implement LLM-based prediction of latent states, we choose two models: GPT 3.5 provided by OpenAI and Gemma 7B-it based on the Gemini technology [9]. One challenge is that not all segmented objects prove to be useful; for instance, objects like the white background in the cart pole system are uninformative by themselves. These non-essential objects require more tokens in the inference of the Large Language Model, which increases the computational cost and also potentially reduces prediction accuracy. We eliminate this redundant information before the step of prompt assembly by removing objects whose centroids do not change at any time in the past observations. Future work can explore other causal techniques [10] to focus on the prediction-relevant objects.

Algorithm 2 Output disassembly into observation

Input: Predicted state $\boldsymbol{z}_{t+m+1}$, current state $\boldsymbol{z}_{t+m}$, current observation $\boldsymbol{y}_{t+m}$, and segmentation algorithm Ω.
Output: Predicted observation $\boldsymbol{y}_{t+m+1}$
FUNCTION
`Disassemble` $(\boldsymbol{z}_{t+m+1}, \boldsymbol{y}_{t+m}, \boldsymbol{z}_{t+m}, \Omega)$:

$\boldsymbol{y}_{t+m+1} \leftarrow \boldsymbol{y}_{t+m}$ ▷ Copy most recent observation
for i from 1 to $\boldsymbol{len}(\boldsymbol{z}_{t+m+1})$ **do**
  $\boldsymbol{\omega}_i \leftarrow \Omega(\boldsymbol{y}_i)$
  $\boldsymbol{\delta}^i \leftarrow \boldsymbol{z}_{t+m+1}^i - \boldsymbol{z}_{t+m}^i$ ▷ Compute the displacement of each object
  for $pixel = True$ in $\boldsymbol{\omega}_i$ **do**
    $px \leftarrow pixel.x$ ▷ x position of the pixel
    $py \leftarrow pixel.y$ ▷ y position of the pixel
    SWAP($\boldsymbol{y}_{t+m+1}[px][py]$, $\boldsymbol{y}_{t+m+1}[px + \delta_x^N][py + \delta_y^N]$) ▷ Move the object into the predicted position
  end for
end for
return $\boldsymbol{y}_{t+m+1}$

World models perform stochastic prediction to ensure diverse outputs. In existing world models, this randomness is implemented by sampling latent states in the VAE, the diversity of which can be controlled with the prior. In our LLM-based world model, we implement randomness with *top-p sampling* [11], for which temperature and threshold are two hyperparameters to control the diversity of the outputs. The top-p sampling restricts the outputs to combinations of words with a cumulative probability higher than the threshold p.

[1]We use "+" to denote string concatenation and adding to a state vector.

		Horizontal position CD error						Vertical position CD error					
Method	Input length	k=10	k=20	k=30	k=40	k=50	k=60	k=10	k=20	k=30	k=40	k=50	k=60
VAE & MDN-LSTM		7.217	7.096	7.030	7.138	7.046	7.125	3.666	3.417	3.280	3.131	3.058	2.908
SAM & MLP		**.0910**	**.1698**	**.2670**	**.3840**	**.5017**	**.5975**	.4230	.7957	**.1092**	**.3905**	.6713	.8780
SAM & LSTM	m=1	.1305	.2558	.3594	.4983	.6096	.7356	.2586	**.5135**	.7338	.9390	**.1772**	**.3371**
SAM & GPT 3.5		3.122	1.175	5.985	3.738	4.793	3.905	2.969	1.195	2.147	4.572	4.794	6.292
SAM & Gemma		.4955	.3124	.4827	.7379	.7316	.7568	**.2385**	.6611	8.739	5.199	5.191	5.736
VAE & MDN-LSTM		3.666	3.417	3.280	3.131	3.058	2.908	3.605	3.359	3.277	3.087	3.057	2.884
SAM & MLP		**.0799**	**.1444**	**.2075**	**.2689**	**.3363**	**.4206**	**.3107**	**.5970**	**.8939**	**.1624**	**.4233**	**.7062**
SAM & LSTM	m=2	.1262	.2461	.3435	.4253	.5065	.5829	.3536	.6680	.9888	.2648	.5389	.8033
SAM & GPT 3.5		.1495	.2244	1.088	1.057	3.014	2.730	1.853	4.176	2.256	3.359	3.401	4.404
SAM & Gemma		.6474	6.449	3.146	.3081	5.564	3.261	.8410	7.952	3.193	.4780	4.426	8.634
VAE & MDN-LSTM		3.605	3.359	3.277	3.087	3.057	2.884	3.612	3.323	3.327	3.116	2.894	2.826
SAM & MLP		**.0684**	**.1333**	**.2039**	**.2795**	**.3313**	.4124	**.2481**	**.4197**	**.5750**	.6773	.7743	.8921
SAM & LSTM	m=4	.1396	.2654	.3928	.5002	.5980	.6748	.3145	.5833	.9137	**.1572**	.3273	**.5452**
SAM & GPT 3.5		.1011	.1654	1.438	1.012	3.704	2.924	1.167	.9396	5.967	2.723	2.396	6.000
SAM & Gemma		.2000	.5749	.5357	.4636	.8839	**.1608**	.6410	.9549	1.9403	.1722	**.2840**	.9854
VAE & MDN-LSTM		3.612	3.323	3.327	3.116	2.894	2.826	3.622	3.390	3.216	2.973	2.880	2.866
SAM & MLP		4.938	7.279	9.578	27.14	45.94	29.42	2.290	75.62	51.35	30.91	54.48	93.20
SAM & LSTM	m=8	.1892	.3761	**.5760**	.7560	.9665	**.2009**	**.3077**	.5708	.8250	.9926	**.1637**	.2818
SAM & GPT 3.5		**.1001**	**.1374**	2.412	1.180	4.247	5.231	.3391	.6566	3.298	2.180	3.143	7.534
SAM & Gemma		.8003	.1471	.9705	**.2189**	**.5253**	.5808	.3354	**.5219**	**.2950**	**.1237**	.9888	**.0901**

TABLE I

STATE PREDICTION PERFORMANCE FOR THE LUNAR LANDER: THE HORIZONTAL AND VERTICAL CENTROID DISTANCE (CD) ERRORS. GRAY ROWS INDICATE THE USE OF ADDITIONAL DATA. THE RANGE OF POSITION IS [0, 20].

		Position CD error (pixel range: [-48, 48])						Angle (degree range [-180°, 180°])					
Method	Input length	Upright			Falling			Upright			Falling		
		k=10	k=20	k=30	k=10	k=20	k=30	k=10	k=20	k=30	k=10	k=20	k=30
VAE & MDN-LSTM		3.767	3.628	4.807	9.083	9.046	11.93	25.48	24.16	25.88	36.76	37.26	37.72
SAM & MLP		1.301	2.750	4.304	**3.248**	5.501	9.101	4.638	8.784	14.54	16.05	21.89	29.40
SAM & LSTM	m=1	**1.276**	**2.541**	4.141	4.391	**4.891**	**3.976**	**4.557**	**7.836**	9.555	19.31	**20.38**	**8.489**
SAM & GPT 3.5		1.983	3.008	**3.604**	5.401	8.537	10.02	6.187	8.270	**9.022**	20.97	30.18	31.47
SAM & Gemma		1.935	3.993	4.582	3.745	8.411	10.236	4.882	10.63	12.58	**8.110**	34.66	36.69
VAE & MDN-LSTM		3.627	3.932	4.747	9.052	9.045	11.87	26.29	25.43	24.50	37.44	36.39	36.77
SAM & MLP		2.264	3.767	6.527	3.662	**4.074**	**7.453**	8.276	17.64	36.89	19.16	27.03	50.79
SAM & LSTM	m=2	2.571	4.815	6.490	5.777	8.626	9.332	8.226	16.61	29.53	22.51	33.27	45.33
SAM & GPT 3.5		**1.685**	3.704	5.778	**2.790**	5.455	10.40	6.330	13.27	18.33	**13.14**	23.49	36.25
SAM & Gemma		1.711	**2.872**	**3.731**	4.251	6.588	9.545	**5.309**	**7.765**	**9.932**	16.11	**20.31**	**28.63**
VAE & MDN-LSTM		2.807	3.713	4.368	9.054	9.031	11.90	26.58	24.69	25.86	37.34	37.42	37.58
SAM & MLP		2.366	3.435	**4.321**	5.491	8.426	10.46	8.344	13.26	23.39	20.69	28.30	37.27
SAM & LSTM	m=4	2.835	4.876	6.700	5.545	9.418	14.42	9.004	14.11	24.51	21.98	30.71	51.05
SAM & GPT 3.5		**1.683**	3.769	6.398	**2.498**	**6.307**	30.50	**5.970**	12.10	19.37	**10.99**	23.84	39.76
SAM & Gemma		1.813	**2.954**	4.453	4.602	6.871	**10.06**	6.153	**7.818**	**12.34**	18.44	**22.30**	**31.00**
VAE & MDN-LSTM		3.682	3.584	4.893	9.000	9.074	12.00	24.94	25.36	25.01	37.30	37.47	36.75
SAM & MLP		2.208	3.610	5.012	5.307	7.989	11.32	19.87	11.77	18.82	14.73	27.05	36.12
SAM & LSTM	m=8	2.485	5.008	7.701	5.134	10.42	16.16	8.853	17.69	31.12	21.04	34.97	59.24
SAM & GPT 3.5		1.826	4.150	7.904	**2.842**	**6.451**	11.34	6.010	13.08	16.55	**11.38**	25.77	35.57
SAM & Gemma		**1.787**	**3.034**	**4.352**	4.798	7.153	**10.32**	**5.803**	**8.685**	**11.46**	19.71	**24.92**	**30.16**

TABLE II

STATE PREDICTION PERFORMANCE FOR THE CART POLE: THE MEAN CENTROID DISTANCE (CD) AND THE MEAN ABSOLUTE ERROR (MAE) FOR THE POLE'S ANGLE. GRAY ROWS INDICATE THE USE OF ADDITIONAL DATA.

After inputting our prompt π, we will receive a formatted state prediction output $\hat{\boldsymbol{z}}$ based on which we evaluate the safety. To continue the prediction sequence into the next step, we need to generate an observation to feed into the image-based controller h. This "output disassembly" step is implemented with Algorithm 2, and describes the process of rebuilding the observation for the controller. This kind of rebuilding won't cause object duplication and loss as shown in Fig 4, which is a common distribution shift that occurs in standard world models. This problem is brought about by the inability of the decoder to construct an observation for an out-of-distribution latent representation predicted by the surrogate model. Our proposed models use meaningful latent states, which are comparable based the CD metric, on which safety predicates can be encoded directly and evaluated without learning. This issue does not exist in the foundation world models because safety can be interpreted by the estimated states and the image reconstruction is precise pixel movement.

		F1 score ↑						FPR ↓					
Method	Input length	k=10	k=20	k=30	k=40	k=50	k=60	k=10	k=20	k=30	k=40	k=50	k=60
VAE & MDN-LSTM		.8610	.8280	.8159	**.8099**	**.7912**	**.7963**	1.000	1.000	1.000	1.000	1.000	1.000
SAM & MLP		.0000	.0000	.0000	.0000	.0000	.0000	.3914	.2845	.2074	.1002	.0535	**.0000**
SAM & LSTM	m=1	.0000	.0000	.0000	.0000	.0000	.0000	**.0676**	**.0784**	**.0579**	**.0579**	**.0289**	**.0000**
SAM & GPT 3.5		**.9566**	**.9324**	**.8622**	.7464	.5328	.6821	.0863	.2293	.3000	.2533	.1006	.2367
SAM & Gemma		.9159	.6667	.7143	.3670	.5233	.6471	1.000	1.000	1.000	.2381	.7480	.6561
VAE & MDN-LSTM		.8558	.8345	.8106	.8067	.7955	.7771	1.000	1.000	1.000	1.000	1.000	1.000
SAM & MLP		.0000	.0000	.0000	.0000	.0000	.0000	.4456	.2692	.2161	.1320	.0400	**.0000**
SAM & LSTM	m=2	.0000	.0000	.0000	.0000	.0000	.0000	.0685	.0802	.0536	.0606	**.0000**	**.0000**
SAM & GPT 3.5		**.9280**	**.9094**	**.8371**	.8624	.6479	.6695	**.0078**	.0443	.0338	.0248	.0353	.0206
SAM & Gemma		.8850	.9038	.3000	**1.000**	**.8651**	**.8329**	.1441	**.0000**	**.0000**	**.0000**	.0221	.0375
VAE & MDN-LSTM		.8535	.8466	.8013	.7995	.8088	.7897	1.000	1.000	1.000	1.000	1.000	1.000
SAM & MLP		.0000	.0000	.0000	.0000	.0000	.0000	.5473	.4475	.3860	.3093	.2871	.2889
SAM & LSTM	m=4	.0000	.0000	.0000	.0000	.0000	.0000	.0825	.0837	.0775	.0597	**.0000**	**.0000**
SAM & GPT 3.5		.9487	.9471	.7580	**.8333**	.4537	.5861	**.0000**	.0075	**.0000**	**.0061**	.0067	**.0000**
SAM & Gemma		**.9759**	**.9815**	**.9367**	.7929	**.8843**	**.8137**	.1304	**.0000**	.1750	.6667	.2889	.6190
VAE & MDN-LSTM		.8270	.8113	.8010	.8045	**.7923**	**.7797**	1.000	1.000	1.000	1.000	1.000	1.000
SAM & MLP		.0000	.0000	.0000	.0000	.0000	.0000	**.0000**	**.0000**	**.0000**	**.0000**	**.0000**	**.0000**
SAM & LSTM	m=8	.0000	.0000	.0000	.0000	.0000	.0000	.0527	.0667	.0742	.0579	.0460	.0638
SAM & GPT 3.5		.9168	**.9710**	.6575	**.8233**	.3260	.6042	.0064	.0068	**.0000**	**.0000**	**.0000**	**.0000**
SAM & Gemma		**.9677**	.7570	**.9582**	.6047	.5596	.7009	.1429	.2391	.1045	.2016	**.0000**	.2639

TABLE III

SAFETY PREDICTION PERFORMANCE FOR THE LUNAR LANDER: F1 SCORE AND FALSE POSITIVE RATE. GRAY ROWS INDICATE THE USE OF ADDITIONAL DATA.

IV. EXPERIMENTS AND RESULTS

A. Experimental Setup

We use simulation environments from OpenAI gym [12] as two cases in this paper: (i) a cart pole with a four-dimensional physical state and (ii) a lunar lander with an eight-dimensional physical state. For the cart pole, the safety threshold θ_{thre} is defined as the angle between the pole and the vertical line θ being less than $\pi/4$: $|\theta| < \pi/4$. For the lunar lander, safety means keeping the horizontal position d_x of the lander within the landing range: $8 < d_x < 12$. For the cart pole, to evaluate safety from our predicted states, we calculate the angle instead of showing the pole's centroid error. For the cart pole, given that the pole is upright most of the time, to get a more detailed evaluation of our models, we split the test experiment of the cart pole into two parts: (i) upright and (ii) falling.

As supervised learning baselines, we train two additional state prediction models: a Multilayer Perceptron (MLP) and a Long Short-Term Memory (LSTM). They are combined with our SAM-based representations. We also compare our approach to the original world model based on VAE representations and an MDN-LSTM state predictor. We test for 10,000 sequences with varied input lengths m and prediction horizons k in each case study. In the results tables, we marked the results of the supervised models in gray because these models take *extra* 30,000 sequences to train and therefore impose a significant data burden compared to our zero-shot foundation world models. This fact relaxes the expectations of the performance of our proposed models.

B. Experimental Results

First, we discuss the state prediction results. For the cart pole, in order to show the safety-related state more intuitively, we calculate the angle by the centroids of the cart and pole to substitute the CD of the pole. As depicted in Table I and II, the errors of all methods are relatively low on short prediction horizons. As the value of k increases, the error generally increases, indicating that predicting states further into the future is more difficult. Specifically, GPT 3.5 and Gemma alternately exhibit relatively low errors in most settings, especially when predicting the falling status. Table I shows the error for the lunar lander, where the foundation models do not perform as well as supervised learning methods on short inputs. Our investigation suggests that this shortcoming is due to insufficiently long input prompts, chosen due to cost constraints; accordingly, the Gemma-based approach beats the rest in most cases for input length 8.

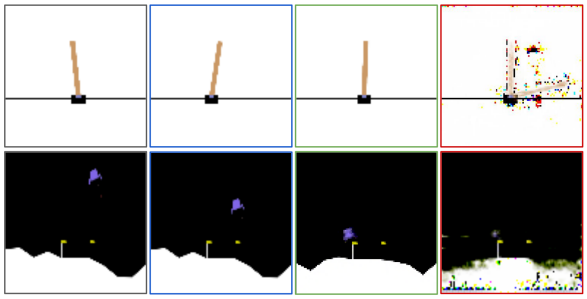

Fig. 4. The observation of cart pole (upper) and lunar lander (lower). From left to right, generated by: the true observation model, a foundation world model, an existing world model without distribution shift, and an existing world model with distribution shift.

Tables III and IV show the results of the safety prediction. The standard world model performs well on short prediction horizons. As the horizon becomes longer, SAM-based mod-

Method	Input length	F1 score ↑						FPR ↓					
		Upright			Falling			Upright			Falling		
		k=10	k=20	k=30	k=10	k=20	k=30	k=10	k=20	k=30	k=10	k=20	k=30
VAE & MDN-LSTM		.9987	**.9915**	.9732	.9905	.9301	.8598	.0676	.4474	1.000	.0610	.4756	**.9980**
SAM & MLP		.9964	.9763	.9317	.9729	.9113	.5915	.0000	**.4328**	1.000	.0000	**.4167**	1.000
SAM & LSTM	m=1	.9964	.9789	.9646	.9729	**.9382**	.8636	.0000	.4328	1.000	.0000	.4167	1.000
SAM & GPT 3.5		.9886	.9909	**.9871**	.8837	.8636	**.8857**	.7857	1.000	1.000	.8407	1.000	1.000
SAM & Gemma		**1.000**	.9531	.9352	**1.000**	.6667	.6021	**.0000**	1.000	1.000	**.0000**	1.000	1.000
VAE & MDN-LSTM		.9977	.9912	.9753	**.9905**	.9318	.8576	.1125	.4545	1.000	.0610	.4634	1.000
SAM & MLP		.9987	**.9941**	.8075	.9870	**.9459**	.6551	.0704	.0666	**.0641**	.0833	.0833	.0833
SAM & LSTM	m=2	**.9994**	.9860	.9670	.9866	.9295	.8529	**.0000**	**.0000**	.0641	**.0000**	**.0000**	**.0833**
SAM & GPT 3.5		.9924	.9768	.9711	.9398	.8981	.8662	.4667	.7333	.7692	.4182	.7167	.8333
SAM & Gemma		.9926	.9906	**.9873**	.9473	.9444	**.9167**	.5625	1.000	1.000	.5000	1.000	1.000
VAE & MDN-LSTM		**.9992**	.9910	.9778	.9861	.9324	.8556	**.0429**	.4730	.9877	.0854	.4593	.9980
SAM & MLP		.9982	**.9979**	**.9933**	**.9870**	**.9867**	**.9743**	.0795	**.0000**	**.1941**	**.0833**	**.0000**	**.1667**
SAM & LSTM	m=4	.9982	.9922	.9346	.9870	.9382	.5901	.0795	.4305	.3980	.0833	.4167	.4167
SAM & GPT 3.5		.9918	.9802	.9625	.9488	.8946	.8487	.3500	.7333	.8182	.3178	.7593	.8890
SAM & Gemma		.9881	.991	.9765	.9062	.9230	.8955	.6440	1.000	1.000	.7500	1.000	1.000
VAE & MDN-LSTM		**.9992**	.9919	.9773	**.9895**	.9324	.8579	**.0484**	.3750	1.000	**.0671**	.4593	.9980
SAM & MLP		.9987	**.9979**	**.9969**	.9870	**.9866**	**.9743**	.0625	**.0000**	.1428	.0833	**.0000**	.1667
SAM & LSTM	m=8	.9987	.9908	.9320	.9870	.9315	.7619	.0625	.1052	**.0833**	.0833	.0833	**.0833**
SAM & GPT 3.5		.9960	.9744	.9650	.9516	.8850	.8535	.2500	.8421	1.000	.2375	.8083	.8958
SAM & Gemma		.9871	.9820	.9829	.9041	.8823	.9062	.6617	1.000	1.000	.7000	1.000	1.000

TABLE IV

SAFETY PREDICTION PERFORMANCE FOR THE CART POLE: F1 SCORE AND FALSE POSITIVE RATE. GRAY ROWS INDICATE THE USE OF ADDITIONAL DATA.

els exhibit higher F1 and lower FPR. The foundation world models have competitive results compared with supervised learning despite being zero-shot. For the lunar lander, the standard world model and supervised learning fail to predict safety accurately. In the standard world model, the FPR is 1 for all predictions, which means that it predicts "safe" at all times. This might be caused by the vulnerability of generative ability [13] to distribution shifts [8] when the model encounters unseen data, with examples shown in Fig. 4. Another consideration is that, since the trajectory of the lunar lander resembles the shape of a sine function, this trajectory may exceed the safety bounds in intermediate states. The supervised learning methods they forecast a smoother trajectory without unsafe intermediate states, which leads to poor safety prediction because safety is defined on the whole sequence, but just on the last state.

The secondary results with SSIM and MSE of predicted observations are shown in Tables V and VI. As mentioned earlier, these metrics are not relevant enough to the dynamics and safety predictions of safety-critical systems. Combining with the metric of the state prediction, it shows that he standard world model can only do well in the MSE and fails in SSIM, which means it has a low quality of predicting and reconstructing the observations.

V. RELATED WORK

A. Foundation Models and Applications

With the rise of large AI models that benefit from the availability of huge computing resources by using a considerable amount of data and model parameters, foundation models achieve high performance on multiple cross-domain tasks. Many well-performing models like Generative Pre-trained Transformer (GPT) [14], Bidirectional Encoder Representations from Transformers (BERT) [15], Contrastive Language-Image Pre-Training (CLIP) [16] and Segment Anything model (SAM) [7] extensively serve as the *foundation* for downstream tasks. We extend the core idea of LLMs from text prediction to complex sequence tasks such as safety prediction, by exploiting the statistical patterns of language learned from large text data.

Furthermore, one significant advantage of LLMs is their extensive pre-training, making them easily adapted to various domains such as robotics decision-making tasks [17]. Another language-guided abstraction [18] can transfer high-level task descriptions into task-relevant state abstractions by using a pre-trained language model. Abstract meaning representation also plays an important role in LLMs and can improve their performance [19]. We believe that LLNs can achieve better performance on complex task sequences because the alignment between human and robot representations is closer than that of human and robot understanding [20].

As the internal structure of LLM is a generative pre-trained transformer, it can be used for any prediction probelms. Some researchers have tested LLM's forecasting ability with humans and it shows that the ability of LLM is approaching the human level [21]. Some new foundation models like Chronos [22] are specialized in non-dynamical prediction tasks like traffic, weather, and energy consumption. Language can also be the input of robotic systems [23] to be trained for end-to-end language-controlled systems like RT2 [24], which can perform basic semantic reasoning to finish its tasks. The trustworthiness of robotic techniques based on foundation models is still an area that has not been widely explored [25]. Besides applying the foundation model

		SSIM ↑						MSE ↓					
Method	Input length	k=10	k=20	k=30	k=40	k=50	k=60	k=10	k=20	k=30	k=40	k=50	k=60
VAE & MDN-LSTM		.6933	.6933	.6931	.6933	.6932	.6933	.1925	.1924	.1925	.1925	.1924	.1924
SAM & MLP		.9922	.9901	.9896	.9892	.9890	.9888	.0013	.0021	.0023	.0024	.0025	.0025
SAM & LSTM	m=1	.9930	**.9914**	**.9905**	.9900	.9897	**.9894**	.0010	**.0016**	**.0020**	**.0021**	.0022	**.0023**
SAM & GPT 3.5		.9923	.9911	**.9905**	**.9903**	**.9909**	.9891	.0015	.0019	.0022	.0022	**.0020**	.0026
SAM & Gemma		**.9950**	.9892	.9896	.9893	.9902	.9889	**.0007**	.0024	.0027	.0024	.0022	.0026
VAE & MDN-LSTM		.6933	.6933	.6932	.6933	.6932	.6932	.1924	.1924	.1924	.1925	.1925	.1924
SAM & MLP		.9929	.9911	.9902	.9897	.9894	.9891	.0010	.0017	.0021	.0022	.0024	.0025
SAM & LSTM	m=2	.9924	.9907	.9899	.9896	.9894	.9891	.0012	.0018	.0021	.0022	.0023	.0024
SAM & GPT 3.5		**.9947**	**.9926**	**.9915**	**.9911**	**.9906**	**.9908**	**.0008**	**.0015**	**.0018**	**.0020**	**.0022**	**.0022**
SAM & Gemma		.9918	.9907	.9897	.9884	.9891	.9834	.0017	.0021	.0021	.0028	.0025	.0027
VAE & MDN-LSTM		.6932	.6933	.6932	.6933	.6933	.6933	.1925	.1925	.1924	.1925	.1925	.1925
SAM & MLP		.9936	.9921	**.9910**	.9904	.9901	**.9898**	.0008	.0014	**.0018**	**.0020**	**.0021**	.0023
SAM & LSTM	m=4	.9927	.9913	.9904	.9900	.9896	.9894	.0011	.0016	.0020	.0021	.0022	.0023
SAM & GPT 3.5		**.9954**	**.9932**	.9908	.9884	.9900	.9862	**.0006**	**.0012**	.0020	.0021	.0023	**.0021**
SAM & Gemma		.9920	.9930	.9909	**.9908**	**.9905**	.9890	.0016	.0013	.0020	**.0020**	**.0021**	.0026
VAE & MDN-LSTM		.6932	.6933	.6932	.6932	.6932	.6932	.1925	.1924	.1924	.1924	.1925	.1925
SAM & MLP		**.9936**	**.9940**	**.9940**	**.9940**	**.9940**	**.9940**	.0015	.0014	**.0014**	**.0014**	**.0014**	**.0014**
SAM & LSTM	m=8	.9929	.9914	.9906	.9900	.9894	.9890	.0011	.0015	.0019	.0021	.0023	.0025
SAM & GPT 3.5		.9847	.9847	.9773	.9565	.9623	.9101	**.0004**	**.0012**	.0022	.0021	.0024	.0020
SAM & Gemma		.9908	.9907	.9900	.9891	.9890	.9894	.0019	.0021	.0023	.0025	.0025	.0024

TABLE V

COMPARISON OF STRUCTURAL SIMILARITY INDEX MEASURE (SSIM) AND MEAN SQUARE ERROR (MSE) FOR LUNAR LANDER'S PREDICTED OBSERVATIONS. GRAY ROWS INDICATE THE USE OF ADDITIONAL DATA.

to control and forecast, novel simulators also adopt LLMs to generate various scenes for traffic simulations [26].

B. Safety Assurance of Autonomous Systems

Trajectory forecasting plays a key role within autonomous systems for a variety of purposes, including safety assurance. One tricky problem with safety assurance based on trajectory prediction is the gap between high-dimensional input and low-dimensional states. For sensors operating in high-dimensional spaces, integrating physics models is crucial for enhancing predictive accuracy. Works such as the Social ODE [27] and Deep Kinematic Models [28] exemplify this approach by merging deep learning techniques with physical models to address issues related to data unreliability. Another popular safety assurance method is reachability analysis. Given the challenge of finding all reachable states in nonlinear dynamics and neural network controllers, specialized verification tools [29], [30] are designed to develop accurate overapproximations of these reachable sets. However, these tools cannot work on high-dimensional controllers such as image-input controllers. One tricky problem with safety assurance based on trajectory prediction is the gap between high-dimensional input and low-dimensional states. High-dimensional verification processes could be tailored, for instance, through the application of generative models [31], and by approximating high-dimensional systems with their lower-dimensional counterparts [32]. Unlike the cases above, we apply SAM to bridge the gap between high-dimensional and low-dimensional space, which requires no additional training like LLNs and is easy to implement.

Finally, conformal prediction is also widely recognized for establishing boundaries around time series forecasts within a defined confidence level [33], [34], which could be used to provide safety bounds for the trajectory with a specific confidence.

VI. DISCUSSION

While our foundation world model demonstrates improved performance in certain aspects and produces physically meaningful states, its dynamic predictions essentially rely on statistical correlations. Also, deploying foundation world models may require human judgment about the relevance of segmented classes; however, for many robotic systems, this effort is dwarfed by the significant data collection required to deploy standard world models or other supervised methods.

In our future work, we plan to delve deeper into the fine-tuning of foundation models to enhance their performance on particular robotic prediction tasks. Additional dynamic states, such as speed and angular velocity, can be derived from the observation sequence and integrated into state representation to develop a more physics-specific surrogate model. In addition, other kinds of multimodal foundation models [16], [35] may also help with building an comprehensive world model.

REFERENCES

[1] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 2451–2463, https://worldmodels.github.io. [Online]. Available: https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution

[2] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," 2022.

[3] V. Micheli, E. Alonso, and F. Fleuret, "Transformers are sample-efficient world models," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=vhFu1Acb0xb

[4] A. Acharya, R. Russell, and N. R. Ahmed, "Competency assessment for autonomous agents using deep generative models," 2022.

Method	Input length	SSIM ↑ Upright			SSIM ↑ Falling			MSE ↓ Upright			MSE ↓ Falling		
		k=10	k=20	k=30	k=10	k=20	k=30	k=10	k=20	k=30	k=10	k=20	k=30
VAE & MDN-LSTM		.8678	.8629	.8567	.8426	.8332	.8297	.0031	.0049	.0058	.0060	.0094	.0103
SAM & MLP		**.9779**	**.9612**	.9494	**.9526**	.9374	.9300	.0023	.0045	.0060	**.0050**	.0074	.0086
SAM & LSTM	m=1	.9757	.9575	.9451	.9487	**.9411**	**.9360**	**.0022**	**.0044**	.0061	.0051	**.0061**	**.0070**
SAM & GPT 3.5		.9691	.9611	**.9564**	.9314	.9225	.9234	.0034	.0048	**.0055**	.0078	.0099	.0098
SAM & Gemma		.9632	.9465	.9322	.9409	.9140	.9076	.0035	.0059	.0066	.0067	.0096	.0121
VAE & MDN-LSTM		.8683	.8629	.8564	.8437	.8327	.8294	.0030	.0049	.0059	.0059	.0094	.0103
SAM & MLP		.9567	.9316	.9116	.9337	.9208	.9045	.0045	.0069	.0092	.0065	**.0076**	.0100
SAM & LSTM	m=2	.9624	.9403	.9170	.9333	.9280	.9137	.0042	.0068	.0089	.0074	.0081	**.0089**
SAM & GPT 3.5		.9600	.9383	.9248	.9400	.9194	.9045	.0038	.0064	.0083	**.0057**	.0086	.0115
SAM & Gemma		**.9724**	**.9623**	**.9575**	**.9418**	**.9335**	**.9231**	**.0030**	**.0046**	**.0053**	.0066	.0086	.0099
VAE & MDN-LSTM		.8682	.8624	.8574	.8432	.8331	.8278	**.0031**	.0049	**.0057**	.0059	.0094	.0103
SAM & MLP		.9596	.9369	.9177	.9292	.9177	.9128	.0042	.0064	.0076	.0079	.0100	**.0101**
SAM & LSTM	m=4	.9539	.9281	.9160	.9207	.9020	.8944	.0050	.0079	.0091	.0086	.0116	.0122
SAM & GPT 3.5		.9594	.9366	.9200	**.9424**	.9157	.9020	.0039	.0066	.0087	**.0054**	.0091	.0115
SAM & Gemma		**.9712**	**.9614**	**.9524**	.9380	**.9294**	**.9213**	.0032	**.0047**	.0060	.0070	**.0089**	.0102
VAE & MDN-LSTM		.8679	.8615	.8566	.8435	.8320	.8288	**.0031**	.0049	**.0057**	**.0058**	.0093	**.0104**
SAM & MLP		.9635	.9441	9266	.9314	.9214	.9139	.0039	.0061	.0078	.0078	.0098	.0108
SAM & LSTM	m=8	.9553	.9295	.9146	.9249	.9006	.8957	.0047	.0079	.0095	.0083	.0122	.0125
SAM & GPT 3.5		.9577	.9354	.9220	.9379	.9140	.9002	.0041	.0068	.0084	.0059	.0092	.0114
SAM & Gemma		**.9712**	**.9611**	**.9522**	**.9342**	**.9288**	**.9170**	.0032	**.0048**	.0061	.0073	**.0091**	.0109

TABLE VI

COMPARISON OF STRUCTURAL SIMILARITY INDEX MEASURE (SSIM) AND MEAN SQUARE ERROR (MSE) FOR CART POLE'S PREDICTED OBSERVATIONS. GRAY ROWS INDICATE THE USE OF ADDITIONAL DATA..

[5] Z. Mao, C. Sobolewski, and I. Ruchkin, "How safe am i given what i see? calibrated prediction of safety chances for image-controlled autonomy," 2024.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[7] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[8] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, no. 1, pp. 521–530, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320311002901

[9] G. Team, "Gemini: A family of highly capable multimodal models," 2023.

[10] M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang, "Causalvae: Structured causal disentanglement in variational autoencoder," 2023.

[11] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," 2020.

[12] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *CoRR*, vol. abs/1606.01540, 2016. [Online]. Available: http://arxiv.org/abs/1606.01540

[13] "Safety and trustworthiness of deep neural networks: A survey," *CoRR*, vol. abs/1812.08342, 2018, withdrawn. [Online]. Available: http://arxiv.org/abs/1812.08342

[14] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *CoRR*, vol. abs/2103.00020, 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[17] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng, "Large language models as general pattern machines," 2023.

[18] A. Peng, I. Sucholutsky, B. Z. Li, T. R. Sumers, T. L. Griffiths, J. Andreas, and J. A. Shah, "Learning with language-guided state abstractions," 2024.

[19] Z. Jin, Y. Chen, F. Gonzalez, J. Liu, J. Zhang, J. Michael, B. Schölkopf, and M. Diab, "Role of semantic representations in an era of large language models."

[20] A. Bobu, A. Peng, P. Agrawal, J. A. Shah, and A. D. Dragan, "Aligning human and robot representations," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '24. ACM, Mar. 2024. [Online]. Available: http://dx.doi.org/10.1145/3610977.3634987

[21] D. Halawi, F. Zhang, C. Yueh-Han, and J. Steinhardt, "Approaching human-level forecasting with language models," 2024.

[22] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, "Chronos: Learning the language of time series," 2024.

[23] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," 2023.

[24] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023.

[25] X. Huang, W. Ruan, W. Huang, G. Jin, Y. Dong, C. Wu, S. Bensalem, R. Mu, Y. Qi, X. Zhao, K. Cai, Y. Zhang, S. Wu, P. Xu, D. Wu, A. Freitas, and M. A. Mustafa, "A survey of safety and trustworthiness of large language models through the lens of verification and validation," 2023.

[26] S. Tan, B. Ivanovic, X. Weng, M. Pavone, and P. Kraehenbuehl, "Language conditioned traffic generation," 2023.

[27] S. Wen, H. Wang, and D. Metaxas, "Social ode: Multi-agent trajectory forecasting with neural ordinary differential equations," in *European Conference on Computer Vision*. Springer, 2022, pp. 217–233.

[28] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider, D. Bradley, and N. Djuric, "Deep kinematic models for kinematically feasible

vehicle trajectory predictions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 563–10 569.

[29] M. Althoff, M. Forets, C. Schilling, and M. Wetzlinger, "Arch-comp22 category report: Continuous and hybrid systems with linear continuous dynamics," in *Proc. of 9th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 2022.

[30] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*. Springer, 2013, pp. 258–263.

[31] S. M. Katz, A. L. Corso, C. A. Strong, and M. J. Kochenderfer, "Verification of image-based neural network controllers using generative models," *CoRR*, vol. abs/2105.07091, 2021. [Online]. Available: https://arxiv.org/abs/2105.07091

[32] Y. Geng, S. Dutta, and I. Ruchkin, "Bridging dimensions: Confident reachability for high-dimensional controllers," 2024.

[33] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, "Adaptive conformal prediction for motion planning among dynamic agents," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 300–314.

[34] X. Qin, Y. Xia, A. Zutshi, C. Fan, and J. V. Deshmukh, "Statistical verification of cyber-physical systems using surrogate models and conformal inference," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2022, pp. 116–126.

[35] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021.