Navigation World Models

Amir Bar[1] **Gaoyue Zhou**[2] **Danny Tran**[3] **Trevor Darrell**[3] **Yann LeCun**[1,2]
[1]FAIR at Meta [2]New York University [3]Berkeley AI Research

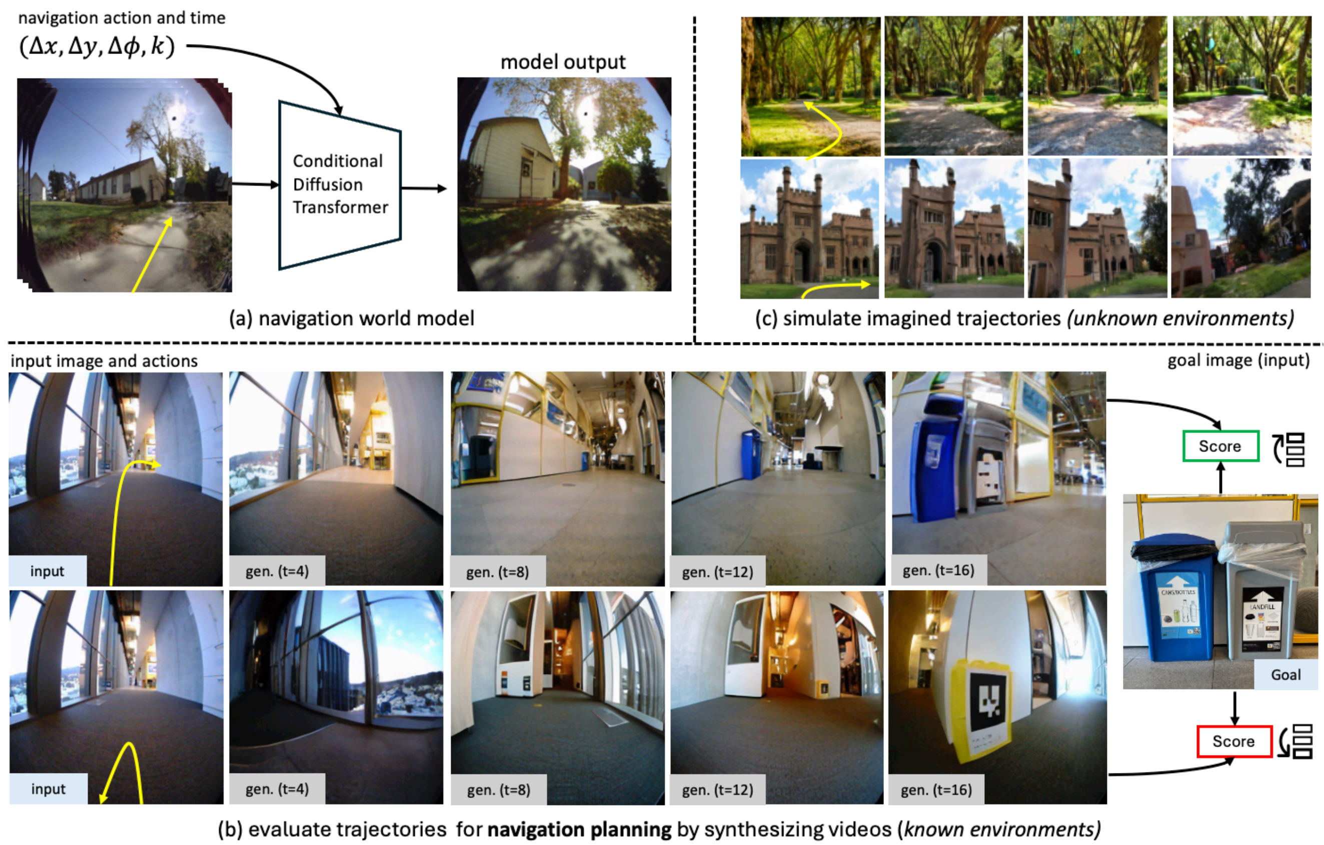

Figure 1. We train a Navigation World Model (NWM) from video footage of robots and their associated navigation actions (a). After training, NWM can evaluate trajectories by synthesizing their videos and scoring the final frame's similarity with the goal (b). We use NWM to plan from scratch or rank experts navigation trajectories, improving downstream visual navigation performance. In *unknown environments*, NWM can simulate imagined trajectories from a single image (c). In all examples above, the input to the model is the first image and actions, then the model auto-regressively synthesizes future observations. **Click on the image to view examples in a browser**.

Abstract

Navigation is a fundamental skill of agents with visual-motor capabilities. We introduce a Navigation World Model (NWM), a controllable video generation model that predicts future visual observations based on past observations and navigation actions. To capture complex environment dynamics, NWM employs a Conditional Diffusion Transformer (CDiT), trained on a diverse collection of egocentric videos of both human and robotic agents, and scaled up to 1 billion parameters. In familiar environments, NWM can plan navigation trajectories by simulating them and evaluating whether they achieve the desired goal. Unlike supervised navigation policies with fixed behavior, NWM can dynamically incorporate constraints during planning. Experiments demonstrate its effectiveness in planning trajectories from scratch or by ranking trajectories sampled from an external policy. Furthermore, NWM leverages its learned visual priors to imagine trajectories in unfamiliar environments from a single input image, making it a flexible and powerful tool for next-generation navigation systems[1].

[1]Project page: https://amirbar.net/nwm

1. Introduction

Navigation is a fundamental skill for any organism with vision, playing a crucial role in survival by allowing agents to locate food, shelter, and avoid predators. In order to successfully navigate environments, smart agents primarily rely on vision, allowing them to construct representations of their surroundings to assess distances and capture landmarks in the environment, all useful for planning a navigation route.

When human agents plan, they often imagine their future trajectories considering constraints and counterfactuals. On the other hand, current state-of-the-art robotics navigation policies [53, 55] are "hard-coded", and after training, new constraints cannot be easily introduced (e.g. "no left turns"). Another limitation of current supervised visual navigation models is that they cannot dynamically allocate more computational resources to address hard problems. We aim to design a new model that can mitigate these issues.

In this work, we propose a Navigation World Model (NWM), trained to predict the future representation of a video frame based on past frame representation(s) and action(s) (see Figure 1(a)). NWM is trained on video footage and navigation actions collected from various robotic agents. After training, NWM is used to plan novel navigation trajectories by simulating potential navigation plans and verifying if they reach a target goal (see Figure 1(b)). To evaluate its navigation skills, we test NWM in *known environments*, assessing its ability to plan novel trajectories either independently or by ranking an external navigation policy. In the planning setup, we use NWM in a Model Predictive Control (MPC) framework, optimizing the action sequence that enables NWM to reach a target goal. In the ranking setup, we assume access to an existing navigation policy, such as NoMaD [55], which allows us to sample trajectories, simulate them using NWM, and select the best ones. Our NWM achieves state-of-the-art standalone performance and competitive results when combined with existing methods.

NWM is conceptually similar to recent diffusion-based world models for offline model-based reinforcement learning, such as DIAMOND [1] and GameNGen [66]. However, unlike these models, NWM is trained across a wide range of environments and embodiments, leveraging the diversity of navigation data from robotic and human agents. This allows us to train a large diffusion transformer model capable of scaling effectively with model size and data to adapt to multiple environments. Our approach also shares similarities with Novel View Synthesis (NVS) methods like NeRF [40], Zero-1-2-3 [38], and GDC [67], from which we draw inspiration. However, unlike NVS approaches, our goal is to train a single model for navigation across diverse environments and model temporal dynamics from natural videos, without relying on 3D priors.

To learn a NWM, we propose a novel Conditional Diffusion Transformer (CDiT), trained to predict the next image state given past image states and actions as context. Unlike a DiT [44], CDiT's computational complexity is linear with respect to the number of context frames, and it scales favorably for models trained up to $1B$ parameters across diverse environments and embodiments, requiring $4\times$ fewer FLOPs compared to a standard DiT while achieving better future prediction results.

In unknown environments, our results show that NWM benefits from training on unlabeled, action- and reward-free video data from Ego4D. Qualitatively, we observe improved video prediction and generation performance on single images (see Figure 1(c)). Quantitatively, with additional unlabeled data, NWM produces more accurate predictions when evaluated on the held-out Stanford Go [24] dataset.

Our contributions are as follows. We introduce a Navigation World Model (NWM) and propose a novel Conditional Diffusion Transformer (CDiT), which scales efficiently up to $1B$ parameters with significantly reduced computational requirements compared to standard DiT. We train CDiT on video footage and navigation actions from diverse robotic agents, enabling planning by simulating navigation plans independently or alongside external navigation policies, achieving state-of-the-art visual navigation performance. Finally, by training NWM on action- and reward-free video data, such as Ego4D, we demonstrate improved video prediction and generation performance in unseen environments.

2. Related Work

Goal conditioned visual navigation is an important task in robotics requiring both perception and planning skills [8, 13, 15, 41, 43, 51, 55]. Given context image(s) and an image specifying the navigation goals, goal-conditioned visual navigation models [51, 55] aim to generate a viable path towards the goal if the environment is known, or to explore it otherwise. Recent visual navigation methods like NoMaD [55] train a diffusion policy via behavior cloning and temporal distance objective to follow goals in the conditional setting or to explore new environments in the unconditional setting. Previous approaches like Active Neural SLAM [8] used neural SLAM together with analytical planners to plan trajectories in the $3D$ environment, while other approaches like [9] learn policies via reinforcement learning. Here we show that world models can use exploratory data to plan or improve existing navigation policies.

Differently than in learning a policy, the goal of a world model [19] is to simulate the environment, e.g. given the current state and action to predict the next state and an associated reward. Previous works have shown that jointly learning a policy and a world model can improve sample efficiency on Atari [1, 20, 21], simulated robotics environments [50], and even when applied to real world robots [71]. More recently, [22] proposed to use a single world model that is shared across tasks by introducing action and task embeddings while [37, 73] proposed to describe actions in language, and [6] proposed to learn latent actions. World models were also explored in the context of game simulation. DIAMOND [1] and GameNGen [66] propose to use diffusion models to learn game engines of computer games like Atari and Doom. Our work is inspired by these works, and we aim to learn a single general diffusion video transformer that can be shared across many environments and different embodiments for navigation.

In computer vision, generating videos has been a long standing challenge [3, 4, 17, 29, 32, 62, 74]. Most recently, there has been tremendous progress with text-to-video synthesis with methods like Sora [5] and MovieGen [45]. Past works proposed to control video synthesis given structured action-object class categories [61] or Action Graphs [2]. Video generation models were previously used in reinforcement learning as rewards [10], pretraining methods [59], for simulating and planning manipulation actions [11, 35] and for generating paths in indoor environments [26, 31]. Interestingly, diffusion models [28, 54] are useful both for video tasks like generation [69] and prediction [36], but also for view synthesis [7, 46, 63]. Differently, we use a conditional diffusion transformer to simulate trajectories for planning without explicit 3D representations or priors.

3. Navigation World Models

3.1. Formulation

Next, we turn to describe our NWM formulation. Intuitively, a NWM is a model that receives the current state of the world (e.g. an image observation) and a navigation action describing where to move and how to rotate. The model then produces the next state of the world with respect to the agent's point of view.

We are given an egocentric video dataset together with agent navigation actions $D = \{(x_0, a_0, ..., x_T, a_T)\}_{i=1}^n$, such that $x_i \in \mathbb{R}^{H \times W \times 3}$ is an image and $a_i = (u, \phi)$ is a navigation command given by translation parameter $u \in \mathbb{R}^2$ that controls the change in forward/backward and right/left motion, as well as $\phi \in \mathbb{R}$ that controls the change in yaw rotation angle.[2]

The navigation actions a_i can be fully observed (as in Habitat [49]), e.g. moving forward towards a wall will trigger a response from the environment based on physics, which will lead to the agent staying in place, whereas in other environments the navigation actions can be approximated based on the change in the agent's location.

Our goal is to learn a world model F, a stochastic mapping from previous latent observation(s) $\mathbf{s}_\tau$ and action a_τ to future latent state representation s_{t+1}:

$$s_i = \text{enc}_\theta(x_i) \qquad s_{\tau+1} \sim F_\theta(s_{\tau+1} \mid \mathbf{s}_\tau, a_\tau) \tag{1}$$

Where $\mathbf{s}_\tau = (s_\tau, ..., s_{\tau-m})$ are the past m visual observations encoded via a pretrained VAE [4]. Using a VAE has the benefit of working with compressed latents, allowing to decode predictions back to pixel space for visualization.

Due to the simplicity of this formulation, it can be naturally shared across environments and easily extended to more complex action spaces, like controlling a robotic arm. Different than [20], we aim to train a single world model across environments and embodiments, without using task or action embeddings like in [22].

The formulation in Equation 1 models action but does not allow control over the temporal dynamics. We extend this formulation with a time shift input $k \in [T_{\text{min}}, T_{\text{max}}]$, setting $a_\tau = (u, \phi, k)$, thus now a_τ specifies the time change k, used to determine how many steps should the model move into the future (or past). Hence, given a current state s_τ, we can randomly choose a timeshift k and use the corresponding time shifted video frame as our next state $s_{\tau+1}$. The navigation actions can then be approximated to be a summation from time τ to $m = \tau + k - 1$:

$$u_{\tau \to m} = \sum_{t=\tau}^{m} u_t \qquad \phi_{\tau \to m} = \sum_{t=\tau}^{m} \phi_t \mod 2\pi \tag{2}$$

This formulation allows learning both navigation actions, but also the environment temporal dynamics. In practice, we allow time shifts of up to ± 16 seconds.

One challenge that may arise is the entanglement of actions and time. For example, if reaching a specific location always occurs at a particular time, the model may learn to rely solely on time and ignore the subsequent actions, or vice versa. In practice, the data may contain natural counterfactuals—such as reaching the same area at different times. To encourage these natural counterfactuals, we sample multiple goals for each state during training. We further explore this approach in Section 4.

3.2. Diffusion Transformer as World Model

As mentioned in the previous section, we design F_θ as a stochastic mapping so it can simulate stochastic environments. This is achieved using a Conditional Diffusion Transformer (CDiT) model, described next.

Conditional Diffusion Transformer Architecture. The architecture we use is a temporally autoregressive transformer model utilizing the efficient CDiT block (see Figure 2), which is applied $\times N$ times over the input sequence of latents with input action conditioning.

[2]This can be naturally extended to three dimensions by having $u \in \mathbb{R}^3$ and $\theta \in \mathbb{R}^3$ defining yaw, pitch and roll. For simplicity, we assume navigation on a flat surface with fixed pitch and roll.

CDiT enables time-efficient autoregressive modeling by constraining the attention in the first attention block only to tokens from the target frame which is being denoised. To condition on tokens from past frames, we incorporate a cross-attention layer, such that every query token from the current target attends to tokens from past frames, which are used as keys and values. The cross-attention then contextualizes the representations using a skip connection layer.

To condition on the navigation action $a \in \mathbb{R}^3$, we first map each scalar to $\mathbb{R}^{\frac{d}{3}}$ by extracting sine-cosine features, then applying a 2-layer MLP, and concatenating them into a single vector $\psi_a \in \mathbb{R}^d$. We follow a similar process to map the timeshift $k \in \mathbb{R}$ to $\psi_k \in \mathbb{R}^d$ and the diffusion timestep $t \in \mathbb{R}$ to $\psi_k \in \mathbb{R}^d$. Finally we sum all embeddings into a single vector used for conditioning:

$$\xi = \psi_a + \psi_k + \psi_t \tag{3}$$

ξ is then fed to an AdaLN [72] block to generate scale and shift coefficients that modulate the Layer Normalization [34] outputs, as well as the outputs of the attention layers. To train on unlabeled data, we simply omit explicit navigation actions when computing ξ (see Eq. 3).

An alternative approach is to simply use DiT [44], however, applying a DiT on the full input is computationally expensive. Denote n the number of input tokens per frame, and m the number of frames, and d the token dimension. Scaled Multi-head Attention Layer [68] complexity is dominated by the attention term $O(m^2n^2d)$, which is quadratic with context length. In contrast, our CDiT block is dominated by the cross-attention layer complexity $O(mn^2d)$, which is linear with respect to the context, allowing us to use longer context size. We analyze these two design choices in Section 4. CDiT resembles the original Transformer Block [68], without applying expensive self-attention over the context tokens.

Diffusion Training. In the forward process, noise is added to the target state $s_{\tau+1}$ according to a randomly chosen timestep $t \in \{1, \dots, T\}$. The noisy state $s_{\tau+1}^{(t)}$ can be defined as: $s_{\tau+1}^{(t)} = \sqrt{\alpha_t} s_{\tau+1} + \sqrt{1-\alpha_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise, and $\{\alpha_t\}$ is a noise schedule controlling the variance. As t increases, $s_{\tau+1}^{(t)}$ converges to pure noise. The reverse process attempts to recover the original state representation $s_{\tau+1}$ from the noisy version $s_{\tau+1}^{(t)}$, conditioned on the context $\mathbf{s}_\tau$, the current action a_τ, and the diffusion timestep t. We define $F_\theta(s_{\tau+1}|\mathbf{s}_\tau, a_\tau, t)$ as the denoising neural network model parameterized by θ. We follow the same noise schedule and hyperparams of DiT [44].

Training Objective. The model is trained to minimize the mean-squared between the clean and predicted target, aiming to learn the denoising process:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{s_{\tau+1}, a_\tau, \mathbf{s}_\tau, \epsilon, t}\left[\|s_{\tau+1} - F_\theta(s_{\tau+1}^{(t)}|\mathbf{s}_\tau, a_\tau, t)\|_2^2\right].$$

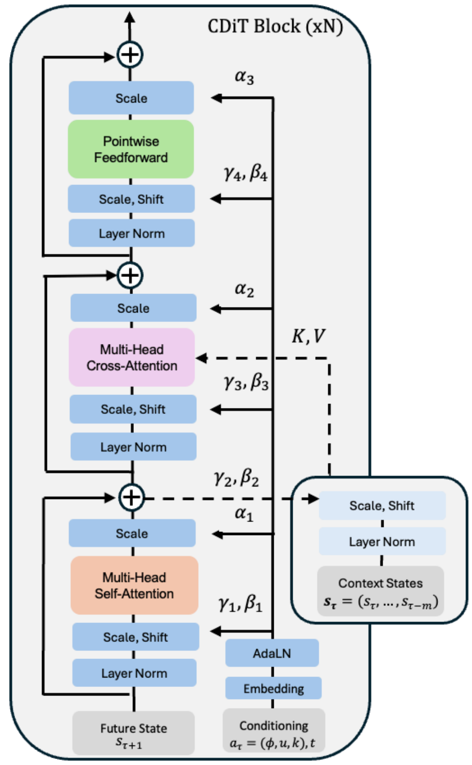

Figure 2. **Conditional Diffusion Transformer (CDiT) Block**. The block's complexity is linear with the number of frames.

In this objective, the timestep t is sampled randomly to ensure that the model learns to denoise frames across varying levels of corruption. By minimizing this loss, the model learns to reconstruct $s_{\tau+1}$ from its noisy version $s_{\tau+1}^{(t)}$, conditioned on the context $\mathbf{s}_\tau$ and action a_τ, thereby enabling the generation of realistic future frames. Following [44], we also predict the covariance matrix of the noise and supervise it with the variational lower bound loss $\mathcal{L}_{\text{vlb}}$ [42].

3.3. Navigation Planning with World Models

Here we move to describe how to use a trained NWM to plan navigation trajectories. Intuitively, if our world model is familiar with an environment, we can use it to simulate navigation trajectories, and choose the ones which reach the goal. In an unknown, out of distribution environments, long term planning might rely on imagination.

Formally, given the latent encoding s_0 and navigation target s^*, we look for a sequence of actions $(a_0, ..., a_{T-1})$ that maximizes the likelihood of reaching s^*. Let $\mathcal{S}(s_T, s^*)$ represent the unnormalized score for reaching state s^* with s_T given the initial condition s_0, actions $\mathbf{a} = (a_0, \dots, a_{T-1})$, and states $\mathbf{s} = (s_1, \dots s_T)$ obtained by autoregressively rolling out the NWM: $\mathbf{s} \sim F_\theta(\cdot|s_0, \mathbf{a})$.

We define the energy function $\mathcal{E}(s_0, a_0, \dots, a_{T-1}, s_T)$, such that minimizing the energy corresponds to maximizing the unnormalized perceptual similarity score and following potential constraints on the states and actions:

$$\mathcal{E}(s_0, a_0, \ldots, a_{T-1}, s_T) = -\mathcal{S}(s_T, s^*) + \tag{4}$$
$$+ \sum_{\tau=0}^{T-1} \mathbb{I}(a_\tau \notin \mathcal{A}_{\text{valid}}) + \sum_{\tau=0}^{T-1} \mathbb{I}(s_\tau \notin \mathcal{S}_{\text{safe}}),$$

The similarity is computed by decoding s^* and s_T to pixels using a pretrained VAE decoder [4] and then measuring the perceptual similarity [14, 75]. Constraints like "never go left then right" can be encoded by constraining a_τ to be in a valid action set $\mathcal{A}_{\text{valid}}$, and "never explore the edge of the cliff" by ensuring such states s_τ are in $\mathcal{S}_{\text{safe}}$. $\mathbb{I}(\cdot)$ denotes the indicator function that applies a large penalty if any action or state constraint is violated.

The problem then reduces to finding the actions that minimize this energy function:

$$\arg \min_{a_0, \ldots, a_{T-1}} \mathbb{E}_{\mathbf{s}} \left[\mathcal{E}(s_0, a_0, \ldots, a_{T-1}, s_T) \right] \tag{5}$$

This objective can be reformulated as a Model Predictive Control (MPC) problem, and we optimize it using the Cross-Entropy Method [48], a simple derivative-free and population-based optimization method which was recently used with with world models for planning [77]. We include an overview of the Cross-Entropy Method and the full optimization technical details in Appendix 7.

Ranking Navigation Trajectories. Assuming we have an existing navigation policy $\Pi(\mathbf{a}|s_0, s^*)$, we can use NWMs to rank sampled trajectories. Here we use NoMaD [55], a state-of-the-art navigation policy for robotic navigation. To rank trajectories, we draw multiple samples from Π and choose the one with the lowest energy, like in Eq. 5.

4. Experiments and Results

We describe the experimental setting, our design choices, and compare NWM to previous approaches. Additional results are included in the Supplementary Material.

4.1. Experimental Setting

Datasets. For all robotics datasets (SCAND [30], TartanDrive [60], RECON [52], and HuRoN [27]), we have access to the location and rotation of robots, allowing us to infer relative actions compare to current location (see Eq. 2). To standardize the step size across agents, we divide the distance agents travel between frames by their average step size in meters, ensuring the action space is similar for different agents. We further filter out backward movements, following NoMaD [55]. Additionally, we use unlabeled Ego4D [18] videos, where the only action we consider is time shift. SCAND provides video footage of socially compliant navigation in diverse environments, TartanDrive focuses on off-road driving, RECON covers open-world navigation, HuRoN captures social interactions. We train on unlabeled Ego4D videos and GO Stanford [24] serves as an unknown evaluation environment. For the full details, see Appendix 8.1.

Evaluation Metrics. We evaluate predicted navigation trajectories using Absolute Trajectory Error (ATE) for accuracy and Relative Pose Error (RPE) for pose consistency [57]. To check how semantically similar are world model predictions to ground truth images, we apply LPIPS [76] and DreamSim [14], measuring perceptual similarity by comparing deep features, and PSNR for pixel-level quality. For image and video synthesis quality, we use FID [23] and FVD [64] which evaluate the generated data distribution. See Appendix 8.1 for more details.

Baselines. We consider all the following baselines.

- **DIAMOND** [1] is a diffusion world model based on the UNet [47] architecture. We use DIAMOND in the offline-reinforcement learning setting following their public code. The diffusion model is trained to autoregressively predict at 56x56 resolution alongside an upsampler to obtrain 224x224 resolution predictions. To condition on continuous actions, we use a linear embedding layer.
- **GNM** [53] is a general goal-conditioned navigation policy trained on a dataset soup of robotic navigation datasets with a fully connected trajectory prediction network. GNM is trained on multiple datasets including SCAND, TartanDrive, GO Stanford, and RECON.
- **NoMaD** [55] extends GNM using a diffusion policy for predicting trajectories for robot exploration and visual navigation. NoMaD is trained on the same datasets used by GNM and on HuRoN.

Implementation Details. In the default experimental setting we use a CDiT-XL of $1B$ parameters with context of 4 frames, a total batch size of 1024, and 4 different navigation goals, leading to a final total batch size of 4096. We use the Stable Diffusion [4] VAE tokenizer, similar as in DiT [44]. We use the AdamW [39] optimizer with a learning rate of $8e-5$. After training, we sample 5 times from each model to report mean and std results. XL sized model are trained on 8 H100 machines, each with 8 GPUs. Unless otherwise mentioned, we use the same setting as in DiT-*/2 models.

4.2. Ablations

Models are evaluated on single-step 4 seconds future prediction on validation set trajectories on the known environment RECON. We evaluate the performance against the ground truth frame by measuring LPIPS, DreamSim, and PSNR. We provide qualitative examples in Figure 3.

Model Size and CDiT. We compare CDiT (see Section 3.2) with a standard DiT in which all context tokens are fed as inputs. We hypothesize that for navigating known environments, the capacity of the model is the most important, and the results in Figure 5, indicate that CDiT indeed performs

Figure 3. **Following trajectories in known environments.** We include qualitative video generation comparisons of different models following ground truth trajectories. **Click on the image to play the video clip in a browser**.

ablation		lpips ↓	dreamsim ↓	psnr ↑
1		0.312 ± 0.001	0.098 ± 0.001	15.044 ± 0.031
2	#goals	0.305 ± 0.000	0.096 ± 0.001	15.154 ± 0.017
4		**0.296** ± 0.002	**0.091** ± 0.001	**15.331** ± 0.027
1		0.304 ± 0.001	0.097 ± 0.001	15.223 ± 0.033
2	#context	0.302 ± 0.001	0.095 ± 0.000	15.274 ± 0.027
4		**0.296** ± 0.002	**0.091** ± 0.001	**15.331** ± 0.027
time only		0.760 ± 0.001	0.783 ± 0.000	7.839 ± 0.017
action only		0.318 ± 0.002	0.100 ± 0.000	14.858 ± 0.055
action + time		**0.295** ± 0.002	**0.091** ± 0.001	**15.343** ± 0.060

Table 1. **Ablations** of predicted goals per sample number, context size, and the use of action and time conditioning. We report prediction results 4 seconds into the future on RECON.

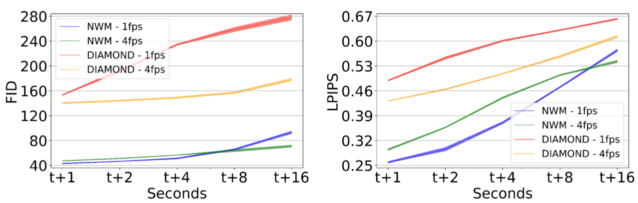

Figure 4. **Comparing generation accuracy and quality** of NWM and DIAMOND at 1 and 4 FPS as function of time, up to 16 seconds of generated video on the RECON dataset.

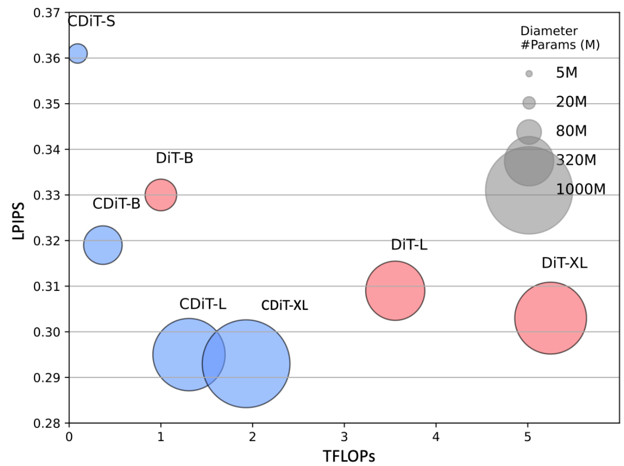

Figure 5. **CDiT vs. DiT**. Measuring how well models predict 4 seconds into the future on RECON. We report LPIPS as a function of Tera FLOPs, lower is better.

model	diamond	NWM (ours)
FVD ↓	762.734 ± 3.361	**200.969** ± 5.629

Figure 6. **Comparison of Video Synthesis Quality.** 16 second videos generated at 4 FPS on RECON.

better with models of up to 1B parameters, while consuming less than 2× FLOPs. Surprisingly, even with equal amount of parameters (e.g, CDiT-L compared to DiT-XL), CDiT is 4× faster and performs better.

Number of Goals. We train models with variable number of goal states given a fixed context, changing the number of goals from 1 to 4. Each goal is randomly chosen between ±16 seconds window around the current state. The results reported in Table 1 indicate that using 4 goals leads to significantly improved prediction performance in all metrics.

Context Size. We train models while varying the number of conditioning frames from 1 to 4 (see Table 1). Unsurprisingly, more context helps, and with short context the model often "lose track", leading to poor predictions.

Time and Action Conditioning. We train our model with both time and action conditioning and test how much each input contributes to the prediction performance (we include the results in Table 1. We find that running the model with time only leads to poor performance, while not conditioning on time leads to small drop in performance as well. This confirms that both inputs are beneficial to the model.

4.3. Video Prediction and Synthesis

We evaluate how well our model follows ground truth actions and predicts future states. The model is conditioned on the first image and context frames, then autoregressively predicts the next state using ground truth actions, feeding back each prediction. We compare predictions to ground truth images at 1, 2, 4, 8, and 16 seconds, reporting FID and LPIPS on the RECON dataset. Figure 4 shows performance over time compared to DIAMOND at 4 FPS and 1 FPS, showing that NWM predictions are significantly more accurate than DIAMOND. Initially, the NWM 1 FPS vari-

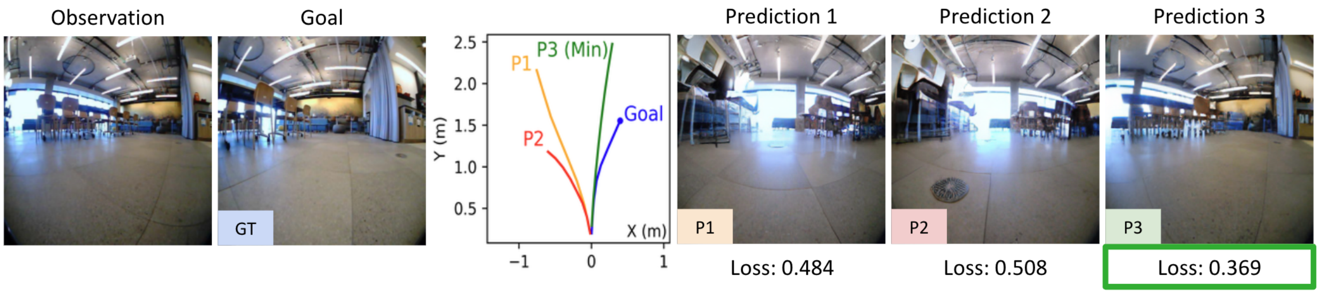

Figure 7. **Ranking an external policy's trajectories using NWM.** To navigate from the observation image to the goal, we sample trajectories from NoMaD [55], simulate each of these trajectories using NWM, score them (see Equation 4), and rank them. With NWM we can accurately choose trajectories that are closer to the groundtruth trajectory. **Click the image to play examples in a browser**.

model	ATE ↓	RPE ↓
GNM	1.87 ± 0.00	0.73 ± 0.00
NoMaD	1.93 ± 0.04	0.52 ± 0.00
NWM + NoMaD (×16)	1.83 ± 0.03	0.50 ± 0.01
NWM + NoMaD (×32)	1.78 ± 0.03	0.48 ± 0.01
NWM (planning)	**1.13** ± 0.02	**0.35** ± 0.01

Table 2. **Goal Conditioned Visual Navigation**. ATE and RPE results on RECON, predicting 2 second trajectories. NWM achieves improved results on all metrics compared to previous approaches NoMaD [55] and GNM [53].

model	Rel. δu ↓	Rel. $\delta \phi$ ↓
forward first	$+0.36 \pm 0.01$	$+0.61 \pm 0.02$
left-right first	-0.03 ± 0.01	$+0.20 \pm 0.01$
straight then forward	$+0.08 \pm 0.01$	$+0.22 \pm 0.01$

Table 3. **Planning with Navigation Constraints.** We present results for planning with NWM under three action constraints, reporting the differences in final position (δu) and yaw ($\delta \phi$) relative to the no-constraints baseline. All constraints are met, demonstrating that NWM can effectively adhere to them.

ant performs better, but after 8 seconds, predictions degrade due to accumulated errors and loss of context and the 4 FPS becomes superior. See qualitative examples in Figure 3.

Generation Quality. To evaluate video quality, we autoregressively predict videos at 4 FPS for 16 seconds to create videos, while conditioning on ground truth actions. We then evaluate the quality of videos generated using FVD, compared to DIAMOND [1]. The results in Figure 6 indicate that NWM outputs higher quality videos.

4.4. Planning Using a Navigation World Model

Next, we turn to describe experiments that measure how well can we navigate using a NWM. We include the full technical details of the experiments in Appendix 8.2.

Standalone Planning. We demonstrate that NWM can be effectively used independently for goal-conditioned navigation. We condition it on past observations and a goal image, and use the Cross-Entropy Method to find a trajectory that minimizes the LPIPS similarity of the last predicted image to the goal image (see Equation 5). To rank an action sequence, we execute the NWM and measure LPIPS between the last state and the goal 3 times to get an average score. We generate trajectories of length 8, with temporal shift of $k = 0.25$. We evaluate the model performance in Table 2. We find that using a NWM for planning leads to competitive results with state-of-the-art policies.

Planning with Constraints. World models allow planning under constraints—for example, requiring straight motion or a single turn. We show that NWM supports constraint-aware planning. In *forward-first*, the agent moves forward for 5 steps, then turns for 3. In *left-right first*, it turns for 3 steps before moving forward. In *straight then forward*, it moves straight for 3 steps, then forward. Constraints are enforced by zeroing out specific actions; e.g., in *left-right first*, forward motion is zeroed for the first 3 steps, and Standalone Planning optimizes the rest. We report the norm of the difference in final position and yaw relative to unconstrained planning. Results (Table 3) show NWM plans effectively under constraints, with only minor performance drops (see examples in Figure 9).

Using a Navigation World Model for Ranking. NWM can enhance existing navigation policies in a goal-conditioned navigation. Conditioning NoMaD on past observations and a goal image, we sample $n \in \{16, 32\}$ trajectories, each of length 8, and evaluate them by autoregressively following the actions using NWM. Finally, we rank each trajectory's final prediction by measuring LPIPS similarity with the goal image (see Figure 7). We report ATE and RPE on all in-domain datasets (Table 2) and find that NWM-based trajectory ranking improves navigation performance, with more samples yielding better results.

4.5. Generalization to Unknown Environments

Here we experiment with adding unlabeled data, and ask whether NWM can make predictions in new environments using imagination. In this experiment, we train a model on all in-domain datasets, as well as a susbet of unlabeled

Figure 8. **Navigating Unknown Environments**. NWM is conditioned on a single image, and autoregressively predicts the next states given the associated actions (marked in yellow). **Click on the image to play the video clip in a browser**.

data	unknown environment (Go Stanford)			known environment (RECON)		
	lpips ↓	dreamsim ↓	psnr ↑	lpips ↓	dreamsim ↓	psnr ↑
in-domain data	0.658 ± 0.002	0.478 ± 0.001	11.031 ± 0.036	**0.295** ±0.002	**0.091** ±0.001	**15.343** ±0.060
+ Ego4D (unlabeled)	**0.652** ±0.003	**0.464** ±0.003	**11.083** ±0.064	0.368 ± 0.003	0.138 ± 0.002	14.072 ± 0.075

Table 4. **Training on additional unlabeled data improves performance on unseen environments.** Reporting results on unknown environment (Go Stanford) and known one (RECON). Results reported by evaluating 4 seconds into the future.

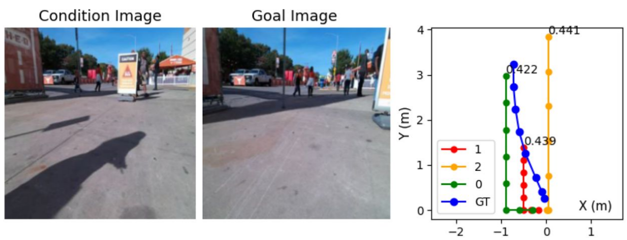

Figure 9. **Planning with Constraints Using NWM.** We visualize trajectories planned with NWM under the constraint of moving left or right first, followed by forward motion. The planning objective is to reach the same final position and orientation as the ground truth (GT) trajectory. Shown are the costs for proposed trajectories 0, 1, and 2, with trajectory 0 (in green) achieving the lowest cost.

Figure 10. **Limitations and Failure Cases.** In unknown environments, a common failure case is mode collapse, where the model outputs slowly become more similar to data seen in training. **Click on the image to play the video clip in a browser**.

videos from Ego4D, where we only have access to the time-shift action. We train a CDiT-XL model and test it on the Go Stanford dataset as well as other random images. We report the results in Table 4, finding that training on unlabeled data leads to significantly better video predictions according to all metrics, including improved generation quality. We include qualitative examples in Figure 8. Compared to in-domain (Figure 3), the model breaks faster and expectedly hallucinates paths as it generates traversals of imagined environments.

5. Limitations

We identify multiple limitations. First, when applied to out of distribution data, the model tends to slowly lose context and generates next states that resemble the training data, a phenomena that was observed in image generation and is known as mode collapse [56, 58]. We include such an example in Figure 10. Second, while the model can plan, it struggles with simulating temporal dynamics like pedestrian motion (although in some cases it does). Both limitations are likely to be solved with longer context and more training data. Additionally, the model currently utilizes 3 DoF navigation actions, but extending to 6 DoF navigation and potentially more (like controlling the joints of a robotic arm) are possible as well, which we leave for future work.

6. Discussion

Our proposed Navigation World Model (NWM) offers a scalable, data-driven approach to learning world models for visual navigation; However, we are not exactly sure yet what representations enable this, as our NWM does not explicitly utilize a structured map of the environment. One idea, is that next frame prediction from an egocentric point of view can drive the emergence of allocentric representations [65]. Ultimately, our approach bridges learning from video, visual navigation, and model-based planning and could potentially open the door to self-supervised systems that not only perceive but can also plan to inform action.

Acknowledgments. We thank Noriaki Hirose for his help with the HuRoN dataset and for sharing his insights, and to Manan Tomar, David Fan, Sonia Joseph, Angjoo Kanazawa, Ethan Weber, Nicolas Ballas, and the anonymous reviewers for their helpful discussions and feedback.

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Thirty-eighth Conference on Neural Information Processing Systems*. 2, 3, 5, 7

[2] Amir Bar, Roei Herzig, Xiaolong Wang, Anna Rohrbach, Gal Chechik, Trevor Darrell, and Amir Globerson. Compositional video synthesis with action graphs. In *International Conference on Machine Learning*, pages 662–673. PMLR, 2021. 3

[3] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024. 3

[4] 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. 3, 5

[5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators, 2024. 3

[6] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. 3

[7] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4217–4229, 2023. 3

[8] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*. 2

[9] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*. 2

[10] Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[11] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 3

[12] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022. 3

[13] J Frey, M Mattamala, N Chebrolu, C Cadena, M Fallon, and M Hutter. Fast traversability estimation for wild visual navigation. *Robotics: Science and Systems Proceedings*, 19, 2023. 2, 3

[14] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *Advances in Neural Information Processing Systems*, 36, 2024. 5, 1

[15] Zipeng Fu, Ashish Kumar, Ananye Agarwal, Haozhi Qi, Jitendra Malik, and Deepak Pathak. Coupling vision and proprioception for navigation of legged robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17273–17283, 2022. 2

[16] Junyu Gao, Xuan Yao, and Changsheng Xu. Fast-slow test-time adaptation for online vision-and-language navigation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 14902–14919. PMLR, 2024. 3

[17] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 3

[18] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 5, 2

[19] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. 2

[20] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, . 3

[21] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, . 3

[22] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*. 3

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5, 1

[24] Noriaki Hirose, Amir Sadeghian, Marynel Vázquez, Patrick Goebel, and Silvio Savarese. Gonet: A semi-supervised deep learning approach for traversability estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3044–3051. IEEE, 2018. 2, 5

[25] Noriaki Hirose, Amir Sadeghian, Fei Xia, Roberto Martín-Martín, and Silvio Savarese. Vunet: Dynamic scene view synthesis for traversability estimation using an rgb camera. *IEEE Robotics and Automation Letters*, 2019. 2

[26] Noriaki Hirose, Fei Xia, Roberto Martín-Martín, Amir Sadeghian, and Silvio Savarese. Deep visual mpc-policy learning for navigation. *IEEE Robotics and Automation Letters*, 4(4):3184–3191, 2019. 3

[27] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 2023. 5, 1, 2

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[29] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 3

[30] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7 (4):11807–11814, 2022. 5, 1

[31] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14738–14748, 2021. 3

[32] Dan Kondratyuk, Lijun Yu, Xiuye Gu, Jose 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. In *Forty-first International Conference on Machine Learning*. 3

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[34] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016. 4

[35] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation, 2024. 3

[36] Han Lin, Tushar Nagarajan, Nicolas Ballas, Mido Assran, Mojtaba Komeili, Mohit Bansal, and Koustuv Sinha. Vedit: Latent prediction architecture for procedural video representation learning, 2024. 3

[37] Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to model the world with language, 2024. 3

[38] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 2

[39] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[40] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[41] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. In *International Conference on Learning Representations*, 2022. 2

[42] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 4

[43] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018. 2

[44] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023. 2, 4, 5

[45] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 3

[46] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*. 3

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 5

[48] Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997. 5, 1

[49] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019. 3

[50] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023. 3

[51] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. In *7th Annual Conference on Robot Learning*. 2

[52] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-

world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021. 5, 1, 2

[53] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023. 2, 5, 7, 3

[54] 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. 3

[55] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024. 2, 5, 7, 1, 3

[56] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in neural information processing systems*, 30, 2017. 8

[57] Jürgen Sturm, Wolfram Burgard, and Daniel Cremers. Evaluating egomotion and structure-from-motion approaches using the tum rgb-d benchmark. In *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*, page 6, 2012. 5, 1

[58] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2020. 8

[59] Manan Tomar, Philippe Hansen-Estruch, Philip Bachman, Alex Lamb, John Langford, Matthew E. Taylor, and Sergey Levine. Video occupancy models, 2024. 3

[60] Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2546–2552. IEEE, 2022. 5, 1, 2

[61] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535, 2018. 3

[62] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 3

[63] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. Megascenes: Scene-level view synthesis at scale. In *Computer Vision – ECCV 2024*, pages 197–214, Cham, 2025. Springer Nature Switzerland. 3

[64] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019. 5, 1

[65] Benigno Uria, Borja Ibarz, Andrea Banino, Vinicius Zambaldi, Dharshan Kumaran, Demis Hassabis, Caswell Barry, and Charles Blundell. A model of egocentric to allocentric understanding in mammalian brains. *bioRxiv*, 2022. 8

[66] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024. 2, 3

[67] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. 2024. 2

[68] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 4

[69] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022. 3

[70] Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in Neural Information Processing Systems*, 37:83951–84009, 2024. 3

[71] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023. 3

[72] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization, 2019. 4

[73] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*. 3

[74] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023. 3

[75] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 1

[76] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[77] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning, 2024. 5