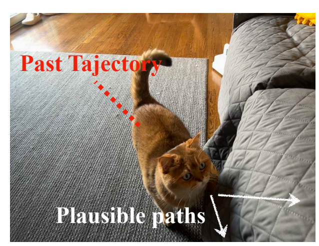
Agent-to-Sim: Learning Interactive Behavior from Casual Videos

Anonymous Author(s)
Affiliation
Address
email

Abstract

Agent behavior simulation empowers robotics, gaming, movies, and VR applications, but building such simulators often requires laborious effort of manually crafting the agent's decision process and motion patterns. Recent advances in visual tracking and motion capture have enabled learning agent behavior from real-world data, but these methods are limited to a few scenarios due to the dependence on specialized sensors (e.g., synchronized multi-camera systems). In a step towards scalable and realistic behavior simulators, we present Agent-to-Sim (ATS), a framework for learning simulatable 3D agents in a 3D environment from casually-captured monocular videos. To deal with partial views, our framework fuses observations in a canonical space for both the agent and the scene, resulting in a dense 4D spatiotemporal reconstruction. We then learn an interactive behavior generator by querying paired data of agents' perception and actions from the 4D reconstruction. ATS enables real-to-sim transfer of agents in their familiar environments given longitudinal video recordings captured with a smartphone over a month. We show results on pets (e.g., cat, dog, bunny) and a person, and analyse how the observer's motion and 3D scene affect an agent's behavior.

1 Introduction



Consider the scene of the cat in the living room: where will the cat go and how will it move? Since we have seen cats interact with the environment and other people many times, we know that cats like to go to the couch, often move slowly, and follow humans around, but run away if people come too close. Such a predictive model of a physical agent is what enables plausible behavior simulation, which is essential for embodied intelligence, immersive virtual environments and robot planning in safety-critical scenarios [9, 31, 41, 45, 54].

The key challenge with behavior simulation is how to generate *plausible* and *interactive* behavior (with respect to the scene and other agents). On one hand, prior works [2, 6, 46] utilize trajectory computed by path-planning algorithms or hand-designed logic from game simulators [13, 58]. While these approaches benefit from high-quality trajectory data paired with perfect object and scene geometries, it is laborious to manually craft simulators that suit the needs of each type of application, and the data distribution is fundamentally different from the real world, leading to unnatural motion and interactions. On the other hand, vision-based motion capture enables learning plausible behavior directly from data for certain scenarios, such as autonomous driving [9], human body motion [21, 36], and interaction with objects/scenes [14, 24]. However, due to the dependence on specialized sensor (synchronized multi-camera systems, IMUs, pre-scanned objects), such systems does not scale well to the full spectrum of natural behavior one may care about, such as behavior of animals, casual events, and long-term activities.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

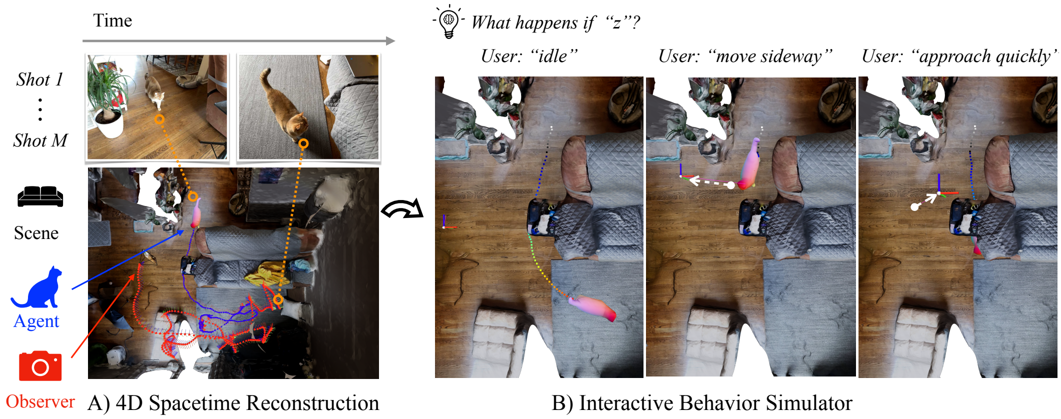


Figure 1: **Learning agent behavior from longitudinal casual video recordings.** We answer the following question: can we simulate the behavior of an agent, by learning from casually-captured videos of the *same* agent recorded across a long period of time (*e.g.*, a month)? A) We first reconstruct videos in 4D (3D & time), which includes the scene, the trajectory of the agent, and the trajectory of the observer (i.e., camera held by observer). Such individual 4D reconstruction are registered across time, resulting in a *complete* 4D reconstructions. B) Then we learn a representation of the agent that allows for interactive behavior simulation. The behavior model explicitly reasons about goals, paths, and full body movements conditioned on the agent's ego-perception and past trajectory. Such agent representation allows us to simulate novel scenarios through conditioning. For example, conditioned different observer trajectories, the cat agent choose to walk to the carpet, stays still while quivering his tail, or hide under the tray stand. *Please see videos and results of other agents in the supplement.*

Recent advances in differentiable rendering [10, 12, 23, 38, 42, 52, 59, 65] and monocular MoCap [28, 43, 69, 70] provide a pathway to obtain high-quality models of scenes and agents from monocular videos alone. Despite the potential of covering diverse data of agent behavior that match the real-world distributions, none of the existing works brings a solution of reconstructing dense 3D structures of both the agent and scene, which is crucial for learning agent behavior grounded in real world environments. To address this, we present ATS (Agent-to-Sim), a framework for learning simulatable agent from casual videos captured over a long time horizon (*e.g.* 1 month), as shown in Fig. 1.

The crucial technical challenge is the presence of partial visibility – in each video captured from an observer's viewpoint, only parts of the agent and the environment are visible. *How do we infer the states of agent and the environment that are not visible?* To build a dense 4D spatiotemporal reconstruction, our key insight is to leverage the observations from multiple videos by fusing them in a canonical 3D space. We introduce a novel coarse-to-fine registration approach that re-purposes "foundational" visual features [40] as a neural localizer, which "registers" the camera with respect to a canonical structure. This enables capturing interactive behavior data in a casual setup (*e.g.*, with a smartphone), and provides paired training data of perception and action of an agent that is grounded in a natural environment (Fig. 2). To learn an interactive behavior model, we condition the action of an agent on their ego-perception, and leverage diffusion models [18, 53] to account for the multimodal nature of goals and planned trajectories. The resulting framework, ATS, can simulate interactive behaviors like those described at the start: agents like pets that leap onto furniture, dart quickly across the room, timidly approach nearby users, and run away if approached too quickly. Our contributions are summerized as follows:

1. **Agent-to-Sim (ATS) Framework.** We introduce a real-to-sim framework, ATS, to learn simulators of interactive agent behavior from casually-captured videos. ATS learns plausible agent behavior that matches the real-world, and is scalable to diverse scenarios, such as animal behavior and casual events.
2. **Environment-Interactive Behavior Simulation.** ATS learns behavior that is *interactive* to the environment, including both the observer and 3D scene. We show the first result of generating plausible behavior of animals that are reactive to observer's motion, and are aware of the 3D scene.

Table 1: **Related works** in behavior data capture. ATS is the only method that builds a complete 4D reconstruction of both the agents and the environment. Different from prior work that focus on specific domains, ATS can be applied to capture interactive behavior of both animals and humans from casual RGBD videos (*e.g*. captured by a smartphone).

Method	Agent Model	Scene Model	Capture Setup	Domain
UCY [30] & ETH [44]	Point	N.A.	Manual Anno.	Pedestrian
nuScenes [9]	Point	Dense 3D Map	Manual Anno.	Pedestrian, Vehicle
SAMP [14]	Parametric Body	Furniture & Objects	Multi-Camera	Human
AMASS [36]	Parametric Body	N.A.	Multi-Camera	Human
ActionMap [47]	Action Class	Sparse 3D Map	Egocentric Camera	Human
ATS (Ours)	Non-parametric	Dense 3D Map	Casual RGBD	Animal, Human

3. **Complete 4D Registration & Reconstruction.** We present a method to register and reconstruct a temporally-evolving 3D scene, whiling accounts for changes in scene layout and appearance.

2 Related Works

Behavior Prediction and Generation. Behavior prediction has a long history, starting from simple physics-based models such as social forces [17] to more sophisticated "planning-based" models that cast prediction as reward optimization [26, 76], where the reward is learned via inverse reinforcement learning [75]. With the advent of large-scale pedestrian and vehicle motion data collected in the navigation and autonomous driving domains [1, 34, 37, 48, 50], generative prediction models such as diffusion models have been able to express behavior multi-modality while being easily controlled via additional signals such as cost functions [20] or logical formulae [74]. However, to capture plausible behavior of agents, these approaches are extremely dependant on high-quality agent trajectory data collected "in the wild" with the associated scene context (*e.g*., 3D map of the scene) [9]. Such data are often manually annotated at a bounding box level (Tab. 1), which limits the scale and the level of detail they can capture. Beyond autonomous driving setup, existing works for human motion prediction and generation [46, 57, 62] have been primarily using simulated data [6] or motion capture data collected with multiple synchronized cameras [14, 24, 36]. Such data provide high-quality full body motion of human using parametric body models [32], but the interactions with the environment are often restricted to a set of pre-defined furnitures and objects [15, 29, 73]. Furthermore, the use of simulated data and motion capture data inherently limits the realism of these behavior generators, since real agents will behave very differently in their familiar environment. To bridge the gap, we develop 4D reconstruction method to obtain high-quality trajectories of agents in their natural environment, with a simple setup that can be achieved with a smartphone. Close to our setup, ActionMap [47] associate daily actions performed by a human agent with an reconstructed 3D environment given egocentric videos. However, they focus on actions performed by hand and do not reconstruct the full body motion of the agent.

4D Reconstruction from Monocular Videos. Reconstructing agents and the environment from monocular videos is challenging due to its under-constrained nature. Given a monocular video, there are multiple different interpretations of the underlying 3D geometry, motion, appearance, and lighting [56]. As such, reconstructing agents often require category-specific 3D prior (*e.g*., 3D humans) [11, 27, 32]. Along this line of work, researchers reconstruct 3D humans aligned to the world coordinate with the help of SLAM and visual odometry [28, 69, 70]. Sitcoms3D [43] reconstructs both the scene and human parameters, while relying on shot changes to determine the scale of the scene. However, the use of parametric body models limits the degrees of freedom they can capture, and makes it difficult to reconstruct agents from arbitrary categories which do not have a pre-built body model, for example, animals. Another line of work avoids using category-specific 3D priors and optimizes the shape and deformation parameters of the agent given richer visual signals (*e.g*., optical flow and object silhouette) [61, 64, 65], which is shown to work well for a broad range of category including human, animals, and vehicles. TotalRecon [52] further incorporates the background scene into the model-free reconstruction pipeline, such that the agent's motion can be decoupled from the camera motion and aligned to the scene space. However, none of the existing methods can reconstruct both the agent and the scene in high-quality. In practice, individual videos may not contain sufficient views, leading to inaccurate and incomplete reconstructions. Our method registers both the agent and the environment from multiple videos into a shared space, which leverages large-scale data collection to build a high-quality agent and scene model.

3 Approach

We describe a method to learn interactive behavior models given longitudinal video recordings of an agent in the same environment. We first build a spatiotemporal 4D reconstruction, including the agent, the scene, and the observer (Sec. 3.1), which is solved by an optimization involving multi-video registration (Sec. 3.2). We then train an interactive behavior model of the agent that is *interactive* with the surrounding environment, including the scene and the motion of the observer (Sec. 3.3).

3.1 4D Representation: Agent, Scene, and Observer

Given multiple monocular videos, our goal is to build a dense spatiotemporal 4D reconstruction of the underlying world, including a deformable agent, a background scene, and a moving observer.

The task is ill-posed due to partial visibility – from an observer's viewpoint, the agent and the environment are only partially visible. To deal with this problem, one principle approach is geometric registration, where structures not visible from one view can be inferred from the other views they appear [51]. We build upon this idea to reconstruct a *complete* spatiotemporal model of an agent and their familiar environment by registering videos captured at different time.

Problem Setup. Specifically, given images from M videos represented by color and feature descriptors [40], $\{\mathbf{I}_i, \boldsymbol{\psi}_i\}_{i=\{1,\ldots,M\}}$, our goal is to find a 4D spatiotemporal representation that explains the video, while pixels with the same semantics can be mapped to consistent canonical 3D locations. Our representation factorizes the 4D structure into a static component and a time-varying component.

Static Representation. $\mathbf{T} = \{\sigma, \mathbf{c}, \boldsymbol{\psi}\}$**.** We represent the static component as agent fields and scene fields. Both define densities, colors, and semantic features in a canonical space,

$$(\sigma_s, \mathbf{c}_s, \boldsymbol{\psi}_s) = \mathrm{MLP}_{scene}(\mathbf{X}, \boldsymbol{\beta}_i), \tag{1}$$

$$(\sigma_a, \mathbf{c}_a, \boldsymbol{\psi}_a) = \mathrm{MLP}_{agent}(\mathbf{X}), \tag{2}$$

where $\mathbf{X}$ corresponds to a 3D point. To account for structures that change across videos, we modify the scene fields to take a per-video latent code $\boldsymbol{\beta}_i$ as input, which allows fitting video-specific details.

Time-varying Representation. $\mathcal{D} = \{\boldsymbol{\xi}, \mathbf{G}, \mathbf{W}\}$**.** The time-varying component includes a moving observer, represented by the camera pose $\boldsymbol{\xi}_t \in SE(3)$, and the motion of an agent, represented by a set of rigid bodies, $\{\mathbf{G}_t^b\}_{\{b=1,\ldots,25\}}$, referred to as "bones". Given a time t, the canonical space of the agent can be mapped to the camera space by blend-skinning deformation [35, 65],

$$\mathbf{X}_t = \mathbf{G}^a \mathbf{X} = \left(\sum_{b=1}^{B} \mathbf{W}^b \mathbf{G}_t^b\right) \mathbf{X}, \tag{3}$$

which computes the motion of a point by blending the bone transformations (we do so in the dual quaternion space [22, 66] to ensure $\mathbf{G}^a$ is a valid rigid transformation). The skinning weights $\mathbf{W}$ are defined as the probability of a point assigned to each bone.

Rendering. To turn the 4D representation into images, we sample rays in the camera space, map them separately to the canonical space of the scene and the agent with $\mathcal{D}$, and query values (e.g., density, color, feature) from corresponding fields of the scene and the agent. The values are then combined before ray integration [39, 52]. Consequently, the rendered pixel values are compared against the observations to update the world representation $\{\mathbf{T}, \mathcal{D}\}$.

Decoupling Agent Motion from Observer. $\{\mathbf{G}_t^b\}_{\{b=1,\ldots,25\}}$ defines the motion of an agent with respect to the observer. Given the observer, we compute the motion of the agent in the scene space as,

$$\mathbf{G}_t^{b\rightarrow s} = \boldsymbol{\xi}_t^{-1} \mathbf{G}_t^b, \tag{4}$$

where the results of extracted trajectories of the agent is shown in Fig. 2

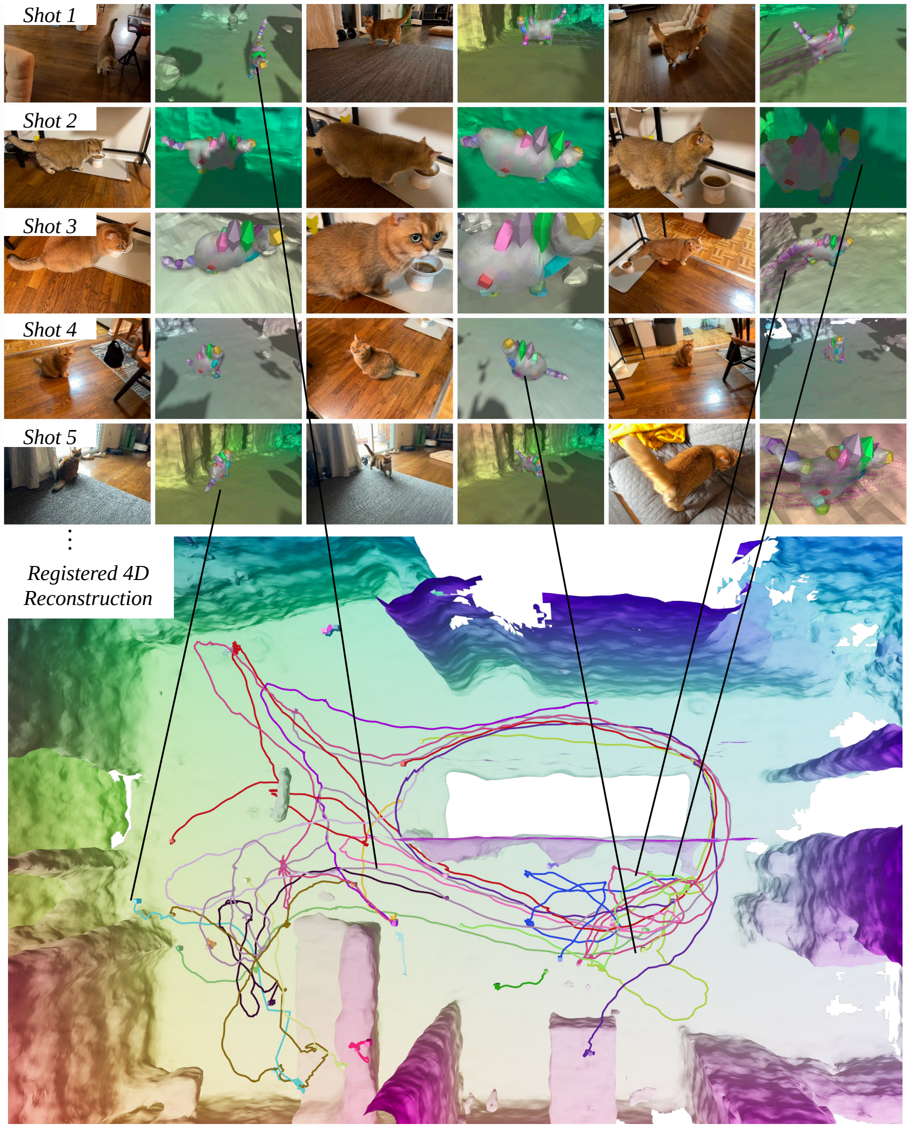


Figure 2: **Results of 4D reconstruction**. Top: reference images and renderings of the reconstructions. The color on the background represents correspondence. The colored blobs on the agent body represent $B = 25$ body parts of the agent (*e.g.*, head is represented by the yellow blob). Bottom: Bird's eye view of the reconstructed scene and agent trajectories, registered to the same scene coordinate. Each colored line represents a unique video sequence where boxes and spheres indicate the starting and the end location. *Please see videos and results on other agents in the supplement*.

3.2 Optimization: Multi-Video Registration

To deal with bad local optima caused by camera poses (Fig. 4), we design a coarse-to-fine registration approach that globally aligns the cameras to a shared canonical space with a feed-forward network, and then jointly optimizes the 3D structures while adjusting the cameras locally.

Initialization: Neural Localization. Due to the evolving nature of scenes across a long period of time [55], there exist both global layout changes (*e.g.*, furniture get rearranged) and appearance changes (*e.g.*, table cloth gets replaced), making it challenging to find accurate geometric correspondences [4, 5, 49]. With the observation that "foundational" visual features have good 3D and viewpoint awareness [3], we adapt them for camera localization. We learn a scene-specific neural localizer that directly regresses the camera pose of an image with respect to a canonical structure,

$$\boldsymbol{\xi} = f_\theta(\boldsymbol{\psi}), \tag{5}$$

where f_θ is a ResNet-18 [16] and $\boldsymbol{\psi}$ is the DINOv2 [40] feature of the input image. We find it to be more robust than geometric correspondence, while being more computationally efficient than performing pairwise matches [49]. To learn the neural localizer, we first capture a walk-through video and build a dense map of the scene. Then we use it to train the neural localizer by randomly sampling camera poses $\mathbf{G}^* = (\mathbf{R}^*, \mathbf{t}^*)$ and rendering images on the fly,

$$\arg\min_\theta \sum_j \left(\| \log(\mathbf{R}_0^T(\theta)\mathbf{R}^*) \| + \|\mathbf{t}_0(\theta) - \mathbf{t}^*\|_2^2 \right), \tag{6}$$

where we use geodesic distance [19] for camera rotation and L_2 error for camera translation. For the agent, we follow BANMo [65] to initialize the root pose $\{\mathbf{G}^b\}_{b=0}$ with a pre-trained pose network.

Objective: Feature-metric Alignemnt. Given a coarse initialization of the observer (scene camera) and the agent's root pose, we use both photometric and featuremetric losses to optimize $\{\mathbf{T}, \mathcal{D}\}$,

$$\min_{\mathbf{T},\mathcal{D}} \sum_t \left(\|I_t - \mathcal{R}_I(t; \mathbf{T}, \mathcal{D})\|_2^2 + \|\boldsymbol{\psi}_t - \mathcal{R}_{\boldsymbol{\psi}}(t; \mathbf{T}, \mathcal{D})\|_2^2 \right) + L_{reg}(\mathbf{T}, \mathcal{D}), \tag{7}$$

where $\mathcal{R}(\cdot)$ is the rendering function described in Sec 3.1. In contrast to prior works, using feature-metric errors makes the optimization robust to change of lighting, appearance, and helps find accurate alignment over multiple videos (Fig. 4). The regularization term includes eikonal loss, silhouette loss, flow loss and depth loss similar to prior works [52, 65].

Scene Annealing. To encourage the reconstructed scene across videos to share a similar structure, we randomly *swap* the code $\boldsymbol{\beta}$ of two videos during optimization, and gradually decrease the probability of swaps from $\mathcal{P} = 1.0 \to 0.05$ over the course of optimization. This regularizes the model to effectively share information across all videos, and keeps video-specific details (Fig. 4).

3.3 Interactive Behavior Generation

Now that we build a complete 4D reconstruction from multiple videos, we can extract a scene structure $\mathbf{T}$, and M trajectories of the agent $\{\mathbf{G}^t\}_{t=\{T_1,\dots,T_M\}}$ as well as the observer $\{\boldsymbol{\xi}^t\}_{t=\{T_1,\dots,T_M\}}$ grounded in the environment. We aim to learn an agent that is interactive with the world.

Hierarchical Behavior Representation. We model the behavior of an agent by bone transformations in the scene space $\mathbf{G} \in \mathbb{R}^{6B\times T^*}$ over a fixed time horizon $T^* = 5.6$s, . We design a hierarchical model as shown in Fig. 3. The body motion $\mathbf{G}$ is conditioned on path $\mathbf{P} \in \mathbb{R}^{3\times T^*}$, which is further conditioned on goal $\mathbf{Z} \in \mathbb{R}^3$. Such decomposition allows agents to react by predicting goals with low latency

Goal Generation. We represent a multi-modal distribution of goals $\mathbf{Z} \in \mathbb{R}^3$ by its score function $s(\mathbf{Z}, \sigma) \in \mathbb{R}^3$ [18, 53]. The score function is implemented as a coordinate MLP [38],

$$s(\mathbf{Z}; \sigma) = \mathrm{MLP}_{\theta_\mathbf{Z}}(\mathbf{Z}, \sigma), \tag{8}$$

trained by predicting the amount of noise $\boldsymbol{\epsilon}$ added to the clean goal, given the corrupted goal $\mathbf{Z} + \boldsymbol{\epsilon}$:

$$\arg\min_{\theta_\mathbf{Z}} \mathbb{E}_{\boldsymbol{Z}} \mathbb{E}_{\sigma\sim q(\sigma)} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})} \|\mathrm{MLP}_{\theta_\mathbf{Z}}(\boldsymbol{Z} + \boldsymbol{\epsilon}; \sigma) - \boldsymbol{\epsilon}\|_2^2 . \tag{9}$$

Compared to methods directly learning the multi-modal distribution [8, 25], diffusion models are easy to train and can be used to generate diverse and high-quality samples [18, 53].

Path Generation with Control. To guide path generation with goals, we represent its score as

$$s(\mathbf{P}; \sigma) = \mathrm{ControlUNet}_{\theta_\mathbf{P}}(\mathbf{P}, \mathbf{Z}, \sigma), \tag{10}$$

where the Control UNet contains two standard UNets with the same architecture [72], one performing unconditional generation taking $(\mathbf{P}, \sigma)$ as input, another injecting goal conditions densely into the neural network blocks of the first one taking $(\mathbf{Z}, \sigma)$ as inputs. Compared to concatenating the goal condition to the noise latent, this encourages close alignment between the goal and the path [62]. We apply the same architecture to control pose generation with paths,

$$s(\mathbf{G}; \sigma) = \mathrm{ControlUNet}_{\theta_\mathbf{G}}(\mathbf{G}, \mathbf{P}, \sigma). \tag{11}$$

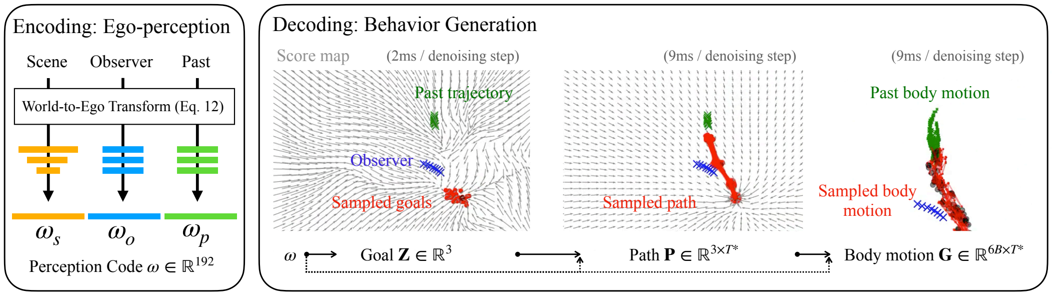


Figure 3: Pipeline for behavior generation. We first encode egocentric information into a perception code ω and then generate full body motion in a hierarchical fashion. We start by generating goals $\mathbf{Z}$ with low latency, and then generate a path $\mathbf{P}$ and body motion $\mathbf{G}$ conditioned on the previous node. Each node is represented by the gradient of its log distribution, trained with the denoising objectives (Eq. 9). Given $\mathbf{G}$, the dense deformation of an agent can be computed via blend skinning (Eq. 3).

Compared to concatenation, we observe better alignment between the path and the full body pose using the Control Unet.

.

Ego-Perception Encoding. To generate plausible interactive behaviors, we encode the world *egocentrically* perceived by the agent, and use it to condition the behavior generation. We use the reconstructed environment $\mathbf{T}$ and the observer $\boldsymbol{\xi}$ as a proxy of the world, and transform them to the egocentric coordinate of the agent,

$$\boldsymbol{\xi}^{s \to a} = \mathbf{G}_{b=0}^{-1} \boldsymbol{\xi}, \quad \mathbf{T}^{s \to a} = \mathbf{G}_{b=0}^{-1} \mathbf{T} \tag{12}$$

Transforming the world to the egocentric coordinates avoids over-fitting to specific locations of the scene (Tab. 2). To encode ego-perception of the scene, we querying feature values from ψ_s with a 3D grid around the agent and extract a latent scene representation,

$$\omega_s = \mathrm{ResNet3D}_{\theta_\psi}(\psi_s). \tag{13}$$

where $\mathrm{ResNet3D}_{\theta_\phi}$ is a 3D ConvNet with residual connections, and $\omega_s \in \mathbb{R}^{64}$ represents the scene perceived by the agent. We encode the observer's motion in the past $T' = 0.8$s seconds with

$$\omega_o = \mathrm{MLP}_{\theta_o}(\boldsymbol{\xi}^{s \to a}), \tag{14}$$

where $\omega_o \in \mathbb{R}^{64}$ represents the observer perceived by the agent. Accounting for the external factors from the "world" enables interactive behavior generation, where the motion of an agent follows the environment constraints and is influenced by the trajectory of the observer (Fig. 5).

History Encoding. We additionally encode the past motion of the agent in T' seconds,

$$\omega_p = \mathrm{MLP}_{\theta_p}(\mathbf{G}_{b=0}^{s \to a}). \tag{15}$$

By conditioning on the past motion, we can generate long sequences by chaining individual ones.

4 Experiments

Dataset. We collect the a dataset that emphasizes the casual interactions of an agent with their familiar environment and the observer. It contains iPhone-captured RGBD video collections of 4 types of agents, including 26 videos of a cat, 3 videos of a dog, 2 videos of a bunny, and 2 videos of a human. The time span of the video capture ranges from 1 day to a month, and each video contains 30 seconds to 2 minutes of content. The dataset is curated to contain diverse motion of agents, including walking, lying down, eating, as well as diverse interaction patterns with the environment, including following the camera, sitting on a coach, etc. Please refer to the supplement for more details.

4.1 4D Reconstruction of Agent & Scene

Implementation Details. We extract frames from the videos at 10 FPS, and use off-the-shelf models to produce augmented image measurements, including object segmentation [68], optical flow [63],

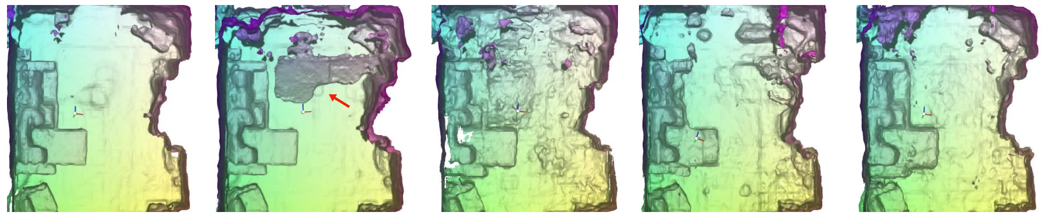


Figure 4: **Comparison on multi-video scene reconstruction**. We show a top-down visualization of the reconstructed scene using the bunny dataset. Compared to TotalRecon that does not register multiple videos, ATS produces higher-quality scene reconstruction. Neural localizer and featuremetric losses are shown important for camera registration. Scene annealing is important for reconstructing high-quality scenes from limited views in a video.

DINOv2 features [40]. We use AdamW to first optimize the environment with featuremetric loss for 30k iterations, and then jointly optimize the environment and agent for another 30k iterations with a combination of optical flow, silouette, and featuremetric losses. Optimization takes roughly 24 hours. 8 A100 GPUs used to optimize 26 videos (for the cat data), and 1 A100 GPU is used in a 2-3 video setup (for dog, bunny, and human data).

Results. We run 4D reconstruction on all video sequences and report the results qualitatively. A visual comparison on scene registration is shown in Fig. 2. Without the ability to register multiple videos, TotalRecon produces protruded and misaligned structures (as pointed by the red arrow). In contrast, our method reconstructs a single coherent scene. With featuremetric alignment (FBA) alone but without a good camera initialization from neural localization (NL), our method produces inaccurate reconstruction due to global misalignment in cameras poses. Removing FBA while keeping NL, the method fails to accurately localize the cameras and produces noisy scene structures. Finally, removing scene annealing procures lower quality scene structures due to lack of training views. A visual comparison with TotalRecon (Single Video) is shown in Fig. 8, where we show that multiple videos helps reconstructing a higher-quality agent, and a more complete scene.

4.2 Interactive Behavior Prediction

Dataset. We use the cat dataset for quantitative evaluation, where the data are split into a training set of 22 videos and a validation set of 4 videos. The validation set is representative of three dominant motion patterns of the agent: (1) trying to engage with the observer, (2) exploring the space and (3) performing activities while not paying attention to the observer.

Implementation Details. To train the behavior model, we slice the reconstructed trajectory in the training set into overlapping window of 6.4s, resulting in 12k data samples. We use AdamW to optimize the parameters of the scores functions $\{\theta_{\mathbf{Z}}, \theta_{\mathbf{P}}, \theta_{\mathbf{G}}\}$ and the ego-perception encoders $\{\theta_\psi, \theta_o, \theta_p\}$ for 120k steps with batch size 1024. Training takes 10 hours on a single A100 GPU.

Metrics. The behavior of an agent can be evaluated along multiple axes, and we focus on goal, path, and body motion prediction. For goal prediction, we use a combination of displacement error (DE) and minimum displacement error (minDE) [7]. The evaluation asks the model to produce K=64 samples. DE computes the avarage distance of the samples to the ground-truth, and minDE finds the one closest to the ground-truth to compute the distance. For path and body motion prediction, we use average displacement error (ADE) and minimum average displacement error (minADE), which are similar to goal prediction, but additionally averages the distance over path and joint locations before taking the min. When evaluating path prediction and body motion prediction, the output is conditioned on the ground-truth goal and path respectively.

Comparisons. We re-purpose related methods and adapt them to our new setup of interactive behavior prediction of animal agents. The quantitative results are shown in Tab. 2. To predict the goal of an agent, classic methods build statistical models of how likely an agent visits a spatial location of the scene, referred to as location prior [26, 76]. Given the extracted 3D trajectories of an agent in the egocentric coordinate, we build a 3D preference map over 3D locations as a histogram, which can be turned into probabilities and used to sample goals. Since this method does not take into account

Table 2: **Evaluation of interactive behavior prediction.** We separately evaluate goal, path, and full body motion prediction. Metrics are displacement errors (DE) in meters and the lower the better. FaF [33] is re-purposed and re-trained with our data.

Method	Goal: minDE	Goal: DE	Path: minADE	Path: ADE	Body: minADE	Body: ADE
Location prior [76]	0.575	2.134	N.A.	N.A.	N.A.	N.A.
FaF [33]	N.A.	1.200	N.A.	0.057	N.A.	0.265
ATS (Ours)	**0.395**	1.299	**0.006**	**0.007**	0.226	0.234
w/o observer ω_o	0.525	1.586	**0.006**	**0.007**	0.225	0.234
w/o scene ω_s	0.702	**1.058**	**0.006**	**0.007**	0.225	0.234
w/o egocentric	0.639	1.424	0.025	0.034	**0.212**	**0.222**

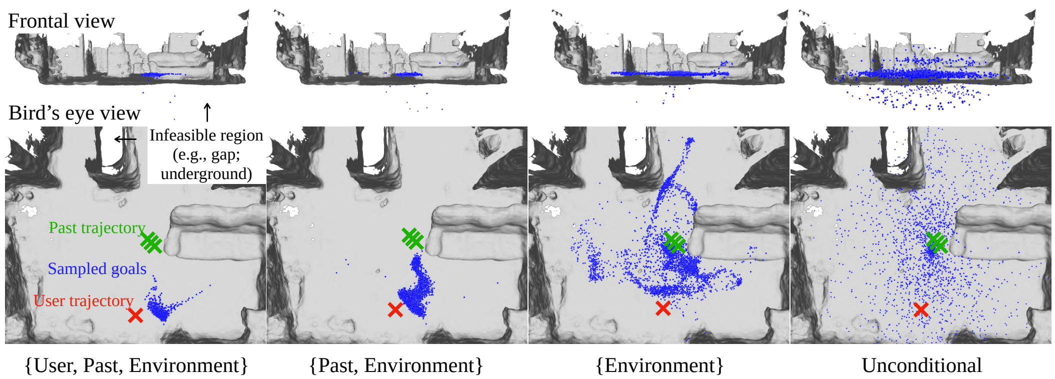


Figure 5: Analysis of conditioning signals. We show results of removing one conditioning signal at a time. Removing observer conditioning and past trajectory conditioning makes the sampled goals more spread out (e.g., regions both in front of the agent and behind the agent); removing the environment conditioning introduces infeasible goals that penetrate the ground and the walls.

of the scene and the observer, it fails to accurately predict the goal. We then re-purpose FaF [33] (Fast-and-Furious), a data-driven approach for motion forecasting to our task. FaF takes the same input as ATS but regresses the goal, path, and body poses. It produces worse results than ATS for all metrics since directly regressing the target treats the underlying distribution as a unit-variance Gaussian and fails to account for the multi-modal nature of agent behaviors.

Analysing Interactions. We analyse the agent's interactions with the environment and the observer by removing the conditioning signals and study their influence on behavior prediction. In Fig. 5, we show that by gradually removing conditional signals, the generated goal samples become more spread out. In Tab. 2, we drop one of the conditioning signals at a time. Dropping the observer conditioning increases the error in goal prediction, indicating observer's trajectory is helpful goal prediction. Dropping the environment conditioning produces worse results on goal prediction (minDE: 0.395 vs 0.702) as well. Surprisingly, it does not affect path prediction. We posit that the scenarios in the test set are too simple. Conditioned on ground-turth goals, it performs well even without environment conditioning. Finally learning behavior generation in the world coordinates performs worse for all metrics since it over-fits to specific locations in the scene.

5 Conclusion

We have presented a framework for learning interactive behavior of agents grounded in natural environments. To achieve this, we turn multiple casually-captured video recordings into complete 4D reconstructions including the agent, the environment, and the observer. Such data collected over a long time period allows us to learn a behavior model of the agent that is reactive to the observer and respects the environment constraints. We validate our design choices on casual video collections, and show better results than prior work for 4D reconstruction and interactive behavior prediction.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[2] A. Bajcsy, A. Loquercio, A. Kumar, and J. Malik. Learning vision-based pursuit-evasion robot policies. *arXiv preprint arXiv:2308.16185*, 2023.

[3] M. E. Banani, A. Raj, K.-K. Maninis, A. Kar, Y. Li, M. Rubinstein, D. Sun, L. Guibas, J. Johnson, and V. Jampani. Probing the 3d awareness of visual foundation models. *arXiv preprint arXiv:2404.08636*, 2024.

[4] E. Brachmann and C. Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019.

[5] E. Brachmann, T. Cavallari, and V. A. Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In *CVPR*, 2023.

[6] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik. Long-term human motion prediction with scene context. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 387–404. Springer, 2020.

[7] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

[8] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[9] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021.

[10] H. Gao, R. Li, S. Tulsiani, B. Russell, and A. Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022.

[11] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa*, and J. Malik*. Humans in 4D: Reconstructing and tracking humans with transformers. In *ICCV*, 2023.

[12] C. Guo, T. Jiang, X. Chen, J. Song, and O. Hilliges. Vid2Avatar: 3D Avatar Reconstruction from Videos in the Wild via Self-supervised Scene Decomposition. *CVPR*, 2023.

[13] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[14] M. Hassan, D. Ceylan, R. Villegas, J. Saito, J. Yang, Y. Zhou, and M. J. Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11374–11384, 2021.

[15] M. Hassan, Y. Guo, T. Wang, M. Black, S. Fidler, and X. B. Peng. Synthesizing physical character-scene interactions. *arXiv preprint arXiv:2302.00883*, 2023.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51 (5):4282, 1995.

[18] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[19] D. Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35:155–164, 2009.

[20] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9644–9653, 2023.

[21] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 41 (1):190–204, 2017.

[22] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007.

[23] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[24] J. Kim, J. Kim, J. Na, and H. Joo. Parahome: Parameterizing everyday home activities towards 3d generative modeling of human-object interactions. *arXiv preprint arXiv:2401.10232*, 2024.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*, pages 201–214. Springer, 2012.

[27] M. Kocabas, N. Athanasiou, and M. J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, June 2020.

[28] M. Kocabas, Y. Yuan, P. Molchanov, Y. Guo, M. J. Black, O. Hilliges, J. Kautz, and U. Iqbal. Pace: Human and camera motion estimation from in-the-wild videos. *arXiv preprint arXiv:2310.13768*, 2023.

[29] J. Lee and H. Joo. Locomotion-action-manipulation: Synthesizing human-scene interactions in complex 3d environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[30] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[31] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. Martinez, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024.

[32] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, 2015.

[33] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018.

[34] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 774–782, 2017.

[35] T. Magnenat, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface'88*, pages 26–33. Canadian Inf. Process. Soc, 1988.

[36] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[37] K. Mangalam, Y. An, H. Girase, and J. Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[38] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[39] M. Niemeyer and A. Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021.

[40] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[41] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[42] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021.

[43] G. Pavlakos, E. Weber, M. Tancik, and A. Kanazawa. The one where they reconstructed 3d humans and environments in tv shows. In *European Conference on Computer Vision*, pages 732–749. Springer, 2022.

[44] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[45] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. Clegg, M. Hlavac, S. Y. Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *The Twelfth International Conference on Learning Representations*, 2023.

[46] D. Rempe, Z. Luo, X. Bin Peng, Y. Yuan, K. Kitani, K. Kreis, S. Fidler, and O. Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13756–13766, 2023.

[47] N. Rhinehart and K. M. Kitani. Learning action maps of large environments via first-person vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–588, 2016.

[48] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

[49] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

[50] A. Seff, B. Cera, D. Chen, M. Ng, A. Zhou, N. Nayakanti, K. S. Refaat, R. Al-Rfou, and B. Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023.

[51] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 2008.

[52] C. Song, G. Yang, K. Deng, J.-Y. Zhu, and D. Ramanan. Total-recon: Deformable scene reconstruction for embodied view synthesis. In *ICCV*, 2023.

[53] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[54] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on robot learning*, pages 477–490. PMLR, 2022.

[55] T. Sun, Y. Hao, S. Huang, S. Savarese, K. Schindler, M. Pollefeys, and I. Armeni. Nothing stands still: A spatiotemporal benchmark on 3d point cloud registration under large geometric and temporal change. *arXiv preprint arXiv:2311.09346*, 2023.

[56] R. Szeliski and S. B. Kang. Shape ambiguities in structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):506–512, 1997.

[57] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[58] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, pages 3–19. Springer, 2011.

[59] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, pages 16210–16220, 2022.

[60] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. *arXiv preprint arXiv:2312.02981*, 2023.

[61] S. Wu, T. Jakab, C. Rupprecht, and A. Vedaldi. Dove: Learning deformable 3d objects by watching videos. *arXiv preprint arXiv:2107.10844*, 2021.

[62] Y. Xie, V. Jampani, L. Zhong, D. Sun, and H. Jiang. Omnicontrol: Control any joint at any time for human motion generation. *arXiv preprint arXiv:2310.08580*, 2023.

[63] G. Yang and D. Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019.

[64] G. Yang, D. Sun, V. Jampani, D. Vlasic, F. Cole, H. Chang, D. Ramanan, W. T. Freeman, and C. Liu. LASR: Learning articulated shape reconstruction from a monocular video. In *CVPR*, 2021.

[65] G. Yang, M. Vo, N. Natalia, D. Ramanan, A. Vedaldi, and H. Joo. Banmo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022.

[66] G. Yang, C. Wang, N. D. Reddy, and D. Ramanan. Reconstructing Animatable Categories from Videos. *CVPR*, 2023.

[67] G. Yang, S. Yang, J. Z. Zhang, Z. Manchester, and D. Ramanan. Physically plausible reconstruction from monocular videos. In *ICCV*, 2023.

[68] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. Track anything: Segment anything meets videos, 2023.

[69] V. Ye, G. Pavlakos, J. Malik, and A. Kanazawa. Decoupling human and camera motion from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21222–21232, 2023.

[70] Y. Yuan, U. Iqbal, P. Molchanov, K. Kitani, and J. Kautz. Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11038–11049, 2022.

[71] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16010–16021, 2023.

[72] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[73] K. Zhao, Y. Zhang, S. Wang, T. Beeler, and S. Tang. Synthesizing diverse human motions in 3d indoor scenes. *arXiv preprint arXiv:2305.12411*, 2023.

[74] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3560–3566. IEEE, 2023.

[75] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[76] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936. IEEE, 2009.

A Additional Implementation Details

Model Architecture. The score function of the goal is implemented as 6-layer MLP with hidden size 128. The the score functions of the paths and body motions are implemented as 1D UNets taken from MDM [57]. The sampling frequency is set to be 0.1s, resulting a sequence length of 56. The environment encoder is implemented as a 6-layer 3D ConvNet with kernel size 3 and channel dimension 128. The observer encoder and history encoder are implemented as a 3-layer MLP with hidden size 128.

We use a linear noise schedule at training time and 50 denoising steps. At test time, each goal denoising step takes 2ms and each path/body denoising step takes 9ms on a GeForce RTX 3090 GPU.

Data Collection. We collect RGBD videos using an iPhone, similar to TotalRecon [52]. To train the neural localizer, we use Polycam to take the walkthrough video and extract a textured mesh. For behavior capture, we use Record3D App to record videos and extract color images and depth images.

B Additional Results

Histogram of Agent / Observer Visitation. We show final camera and agent registration to the canonical scene in Fig. 6. The registered 3D trajectories provides statistics of agent's and user's preference over the environment.

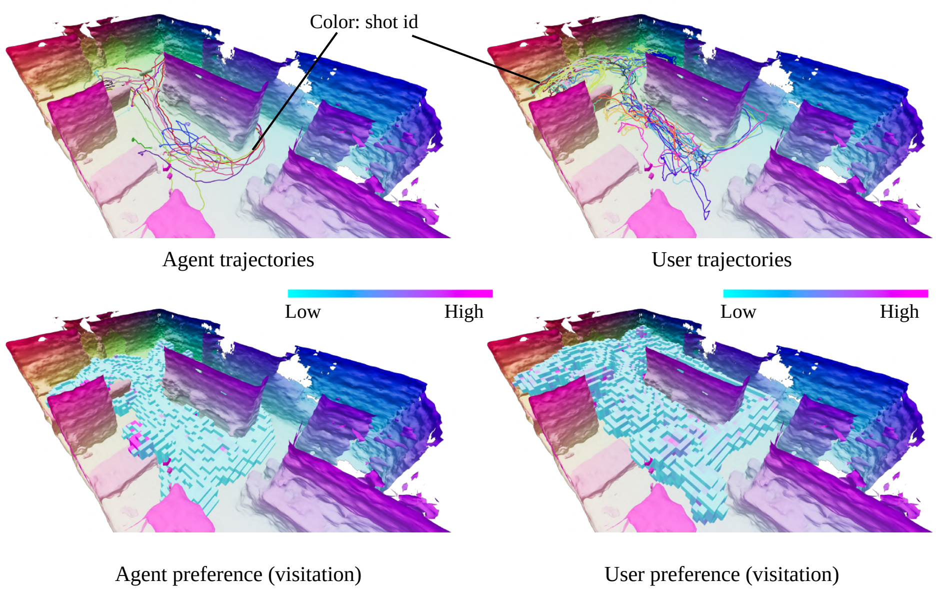


Figure 6: Given the 3D trajectories of the agent and the user accumulated over time (top), one could compute their preference represented by 3D heatmaps (bottom). Note the high agent preference over table and sofa.

Varying Observer's Motion. We find that various interactive behaviors can be generated by conditioning the model on different observer motion. The results are shown in Fig. 7.

Comparison to TotalRecon. In the main paper, we compare to TotalRecon on scene reconstruction by providing it multiple videos. Here, we include additional comparison in their the original single video setup. We find that TotalRecon fails to build a good agent model, or a complete scene model given limited observations, while our method can leverage multiple videos as inputs to build a better agent and scene model. The results are shown in Fig. 8.

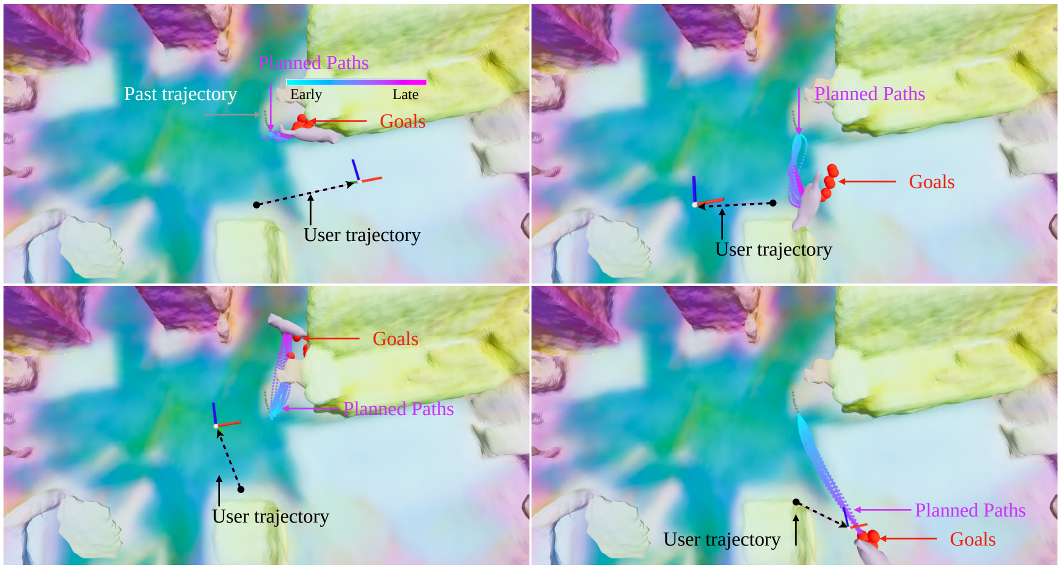


Figure 7: Interactive behavior simulation with user conditioning. By changing the trajectory of the user, one could influence the behavior of the agent. Given different control inputs, the agent may follow the user or run away from the user.

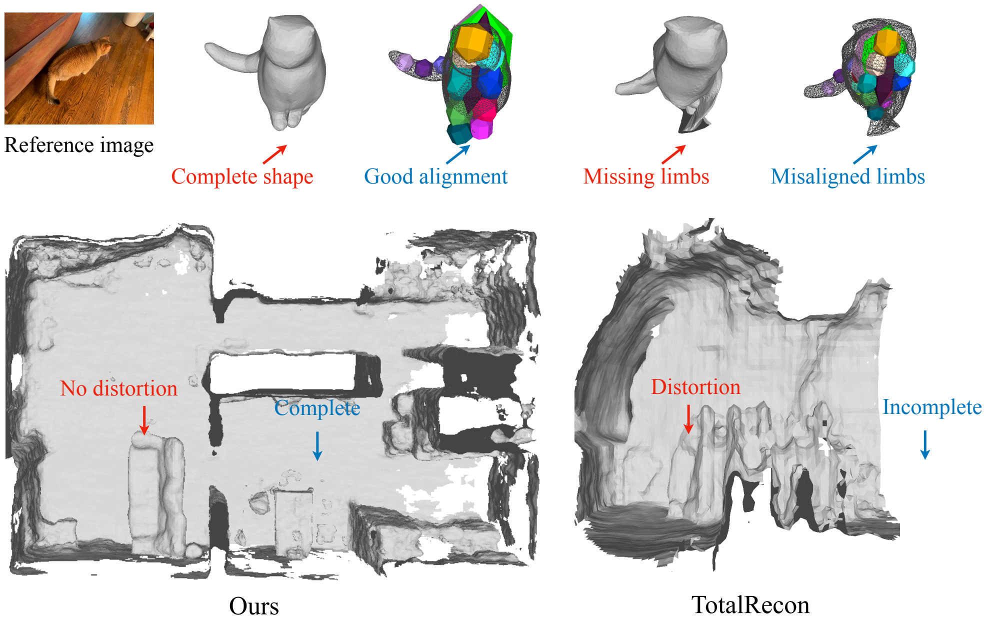


Figure 8: Qualitative comparison with TotalRecon [52] on 4D reconstruction. Top: reconstruction of the agent at at specific frame. Total-recon produces shapes with missing limbs and bone transformations that are misaligned with the shape, while our method produces complete shapes and good alignment. Bottom: reconstruction of the environment. TotalRecon produces distorted and incomplete geometry (due to lack of observations from a single video), while our method produces an accurate and complete environment reconstruction.

C Limitations and Future Works

High-level Behavior. The current ATS model is trained with time-horizon of $T^* = 6.4$ seconds. We observe that the model only learns mid-level behaviors of an agent (e.g., trying to move to a destination; staying at a location; walking around). We hope incorporating a memory module and training with longer time horizon will enable learning higher-level behaviors of an agent.

Scaling-up. As indicated by the experimental results, the goals sampled from ATS may fail to cover the actual goal when evaluated on the (unseen) test data. This raises safety concerns when using ATS for the prediction task (e.g., predicting the behavior of pedestrains in autonomous driving). One potential solution of improving the generalization ability is to collect more diverse behavior data from in the wild videos, or leverage "large" video priors trained on internet-scale videos.

Multiple Agents. We show results of learning behavior models of a single agent, but our method for 4D reconstruction and interactive goal-driven behavior modeling is not limited to a single agent. We leave learning multi-agent behavior simulation from videos as future work.

Physical Interactions. Our method reconstructs and generates the kinematics of an agent, which may produce physically-implausible results (e.g., penetration with the ground and foot sliding). One promising way to deal with this problem is to add physics constraints to the reconstruction and motion generation [67, 71].

Environment Reconstruction. To build a complete reconstruction of the environment, we register multiple videos to a shared canonical space. However, the transient structures (e.g., cushion that can be moved over time) may not be reconstructed well due to lack of observations. One potential solution of reconstructing these transient structures is to combine generative image priors with the reconstruction pipeline [60].

D Social Impact

Our method is able to learn interactive behavior from videos, which could help build simulators for autonomous driving, gaming, and movie applications. It is also capable of building personalized behavior models from casually collected video data, which can benefit users who do not have access to a motion capture studio. On the negative side, the behavior generation model could be used as "deepfake" and poses threats to user's privacy and social security.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The paper discusses the limitations of the work performed by the authors.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The authors tried their best to disclose the information needed to reproduce the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code will be released once we put it in a better shape.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The authors tried their best to specify all the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The results currently do not have error bars, but we will try adding them later. Based on empirical evidence of running the experiments, we think it will not affect the conclusion.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides information about computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the code of ethics and think the paper follows the guideline.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discussed potential positive and negative impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: Thee paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper discussed the new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not deal with crowdsourcing or external human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not deal with crowdsourcing or external human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.