Envisioning the Future, One Step at a Time

Stefan Andreas Baumann* [1,2] Jannik Wiese* [1,2]
Tommaso Martorella[1,2] Mahdi M. Kalayeh[3] Björn Ommer[1,2]

[1]CompVis @ LMU Munich [2]Munich Center for Machine Learning (MCML) [3]Netflix

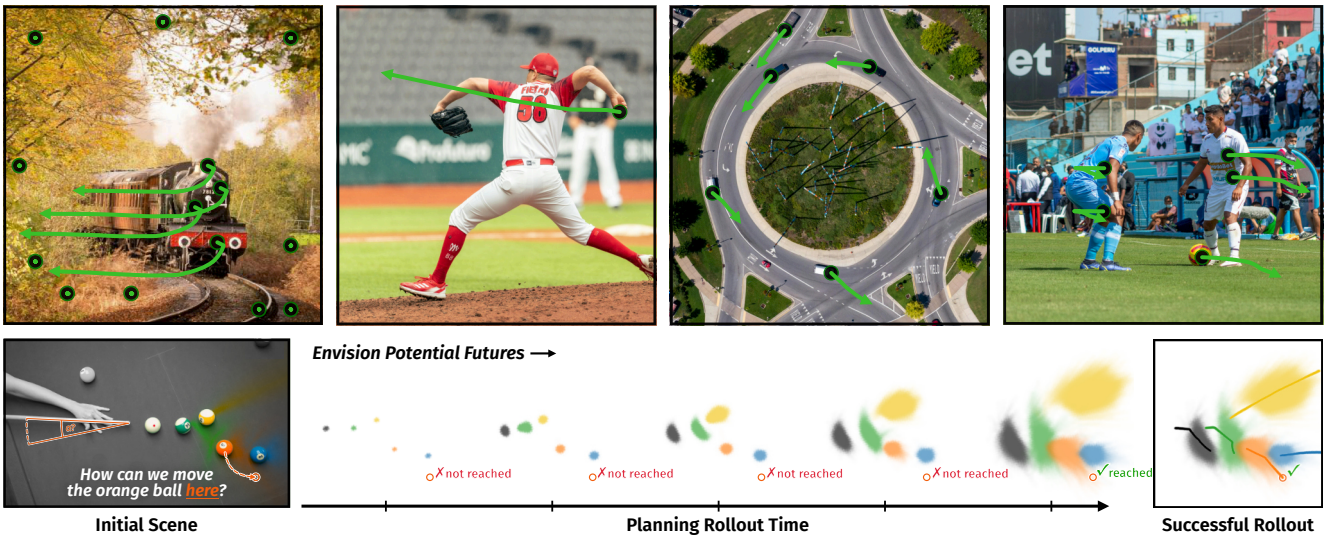


Figure 1. From a single image, our model envisions diverse, physically consistent futures in open-set environments (top). By exploring directly in motion space, it can rapidly perform thousands of counterfactual rollouts – here to select a candidate billiard shot (bottom).

Abstract

Accurately anticipating how complex, diverse scenes will evolve requires models that represent uncertainty, simulate along extended interaction chains, and efficiently explore many plausible futures. Yet most existing approaches rely on dense video or latent-space prediction, expending substantial capacity on dense appearance rather than on the underlying sparse trajectories of points in the scene. This makes large-scale exploration of future hypotheses costly and limits performance when long-horizon, multi-modal motion is essential. We address this by formulating the prediction of open-set future scene dynamics as step-wise inference over sparse point trajectories. Our autoregressive diffusion model advances these trajectories through short, locally predictable transitions, explicitly modeling the growth of uncertainty over time. This dynamics-centric representation enables fast rollout of thousands of diverse futures from a single image, optionally guided by initial constraints on motion, while maintaining physical plausibility and long-range coherence. We further introduce OWM, a benchmark for open-set motion prediction based on diverse in-the-wild videos, to evaluate accuracy and variability of predicted trajectory distributions under real-world uncertainty. Our method matches or surpasses dense simulators in predictive accuracy while achieving orders-of-magnitude higher sampling speed, making open-set future prediction both scalable and practical.
Project page: http://compvis.github.io/myriad.

1. Introduction

A key feature of intelligence is the ability to *envision* possible futures and use them to guide behavior [88–90, 92], rather than merely reacting after they have become reality – anticipating how motion might unfold [9, 55, 104] rather than retracing how it already has. Since we live in a highly dynamic world, we need to quickly predict and simulate potential *future* movements and interactions in the environment

*Equal Contribution.

around us. Yet, the complexity of our world is staggering: every hidden contact, every subtle interaction could, in principle, dramatically change future scene dynamics. Our minds cope with this open-set chaos through abstraction [13, 52]: we do not "paint" a picture of the future, we trace only the important changes that matter. This sparsity is what makes efficiently envisioning the future possible, as long as it remains the future.

In contrast, most current generative (world) models, however, attempt the opposite. Video [6, 19, 75] and latent space [45, 54, 124] simulators predict dense representations of entire scenes, expending enormous capacity on aspects irrelevant to scene dynamics. This makes envisioning the future in open-ended settings, precisely when many possible futures must be considered, prohibitively costly.

Moreover, the world is deeply interwoven and stochastic: between now and any future moment lies an immense chain of interactions and entanglements. Thus, predicting what the world will look like even a few seconds from now cannot be done in a single leap. Instead, we must simulate the intervening interactions step by step – just as we do not foresee the outcome of billiard shot all at once, but unroll it gradually and abstractly, collision by collision. Previous models that tried to predict that distant outcome in one step [10] must implicitly account for every interaction at once. This implies an impossible burden unless tasks are trivially "one-hop" or model capacity is unbounded. Otherwise, the only feasible approach is to unfold the future step by step, progressing through short, locally predictable transitions where the web of interactions remains manageable.

Each step depends on the previous and models the growth of uncertainty over time. This incremental structure is what makes reasoning under real-world complexity feasible. Implementing this principle computationally allows us to envision the future, and the many ways of getting there, not just once, but thousands of times, effectively simulating the inherent stochasticity of our environment.

Technically, we formulate this as an autoregressive diffusion model over sparse trajectories. It learns from diverse in-the-wild videos and generalizes to open-set dynamics of everyday scenes. The model perceives the world through a single image and subsequently envisions diverse futures through fast rollouts, optionally guided by initial motion cues. The efficient sparse representation of scene dynamics, rather than appearance, allows us to enumerate hypotheses and capture the stochasticity of our world orders of magnitude faster than dense video models.

To ground this task, we introduce OWM, a benchmark for open-world motion prediction that evaluates whether models can generate physically consistent, diverse trajectories under real-world uncertainty. Across both structured and open-set domains, our model achieves accuracy on par with or surpassing dense approaches, while enabling exploration of far more futures within the same compute budget.

By focusing on dynamics instead of pixels, we make motion prediction not only faster, but fundamentally more scalable: a model that does not paint the world frame by frame, but *envisions how it moves*.

We summarize our main contributions as follows:

- We cast visual motion prediction as *open-set*, *step-wise* modeling of distributions over *sparse point trajectories* from a single image, allowing models to envision how complex, unconstrained scenes evolve without rendering appearance.
- We introduce an autoregressive diffusion model tailored to this formulation, with an efficiency-optimized architecture that enables large-scale, fast sampling of diverse futures.
- We present OWM, a benchmark designed to evaluate the physical plausibility and accuracy of trajectory distributions under open-set conditions.
- We demonstrate that our approach matches or surpasses dense models in accuracy while being orders of magnitude faster, thereby enabling the exploration of thousands of plausible futures within the same compute budget.

2. Related Work

We can examine the relevant literature on motion prediction from four distinct perspectives: *visual tax*, *granularity*, *domain*, and *paradigm*. A model that requires video generation as a prerequisite for motion prediction is considered *dense* and incurs the *visual tax*, as it must generate every pixel before it can reason about motion dynamics. *Domain* refers to the environment in which a model operates and its ability to generalize to previously unobserved settings. For instance, a physics simulator may not incur the *visual tax* because, after interpreting the scene, it relies solely on physics engines to reason about possible futures. However, such models often suffer from a limited *domain*, rendering them obsolete or irrelevant in real-world and in-the-wild scenarios. Finally, *paradigm* pertains to whether motion is modeled in a *single-shot* or *step-by-step* manner. The latter enables more sophisticated reasoning, allowing not only for the prediction of the final state but also for the explanation of motion dynamics (i.e., *how* the system evolves to that state). In the remainder of this section, we adopt these definitions to briefly review the literature and clearly position our work relative to prior approaches.

Generation of potential motion from static images has been widely explored in the literature. Modern video generation [1, 16, 19, 24, 57, 74, 78, 86, 100, 118] and video world models [3, 6, 8, 14, 15, 26, 28, 29, 38, 45, 51, 59, 91, 105, 108, 126] can produce *dense* sequences of possible futures from a single starting image and/or a short context. However, these approaches incur a significant *visual tax*: they model appearance and its temporal evolution alongside the *dense* motion dynamics of

the entire scene, making open-ended prediction, and especially branching, extremely expensive. Image-to-dense motion techniques [12, 17, 61, 63, 94, 111, 125] primarily aim to produce motion. When directly generating motion [12, 61, 63, 94], these methods can avoid the *visual tax*. Nevertheless, by modeling *all* motion rather than a decision-centric subset, they significantly increase computational demands for prediction and are prone to error accumulation. The same limitation applies to feature-space world models that operate on generic representations [5, 54, 124] or domain-specific image embeddings [20, 40–44]. In contrast, our approach not only completely avoids the *visual tax*, but also focuses computation *only* on understanding motion by modeling distributions over a *sparse* set of user-defined points. This eliminates the need for dense prediction of motion dynamics and enables extensive exploration of potential motion, including branching.

Another group of prior works first estimate the physical properties (e.g., object shape, mass, friction, pose) of the scene and then leverage off-the-shelf physics engines to predict scene motion [9, 23, 50, 62, 66, 70, 115–117]. These methods can produce highly accurate motion when the dynamics are fully in-domain for the physics engine and parameter estimation is exact, but they fail to generalize to truly open-set motion, including everyday scenarios or in-the-wild visuals. In contrast, our approach performs motion prediction in a fully open-set regime and learns all dynamics in a purely data-driven manner, without relying on external components such as a physics engine.

Most existing literature [10, 37, 79, 83, 95, 110] frames the problem of motion prediction from a single image as a one-shot task. These approaches either demand extremely high model capacity to handle multi-contact and long-horizon scenarios, or they incur a substantial *visual tax*, comparable to that seen in auto-regressive video models. This limitation arises because such methods depend on pixel-level outputs to reason across multiple steps. In essence, after making a single-step prediction, the model must convert this prediction back into the visual domain [83] before it can be used as input for generating the next step, resulting in a back-and-forth (i.e. encoding-decoding) process between real and latent spaces. In contrast, we employ *step-wise* auto-regressive generation over *sparse* points, enabling long horizons and explicit explorations. In other words, we demonstrate that multi-step reasoning about a scene's motion does not require attention to every single pixel, a property that unlocks significant potential for efficient, long-horizon, multi-step reasoning.

It is worth mentioning that prior efforts exist in predicting the motion of a sparse set of objects; however, these methods typically operate in narrow domains such as multi-agent/social forecasting [2, 39, 73, 87], autonomous driving [21, 36, 64, 72, 106, 122], human-pose motion [18, 85, 119], or fully specified custom environments [35, 71], and typically require abstract inputs, thereby limiting their general applicability. Unlike these works, we specifically target *open-set* motion prediction in unconstrained scenes at a granularity specified by the user during inference, learning to parse and reason directly in a multi-step manner from appearance at sparse decision points.

In summary, compared to prior work, our approach offers key advantages across all four axes defined at the outset. Unlike dense video generation and feature-based models, which pay a high *visual tax* by operating at the pixel level, our method entirely avoids this cost by modeling motion only over a *sparse* set of user-defined points, thus achieving fine control over *granularity*. In terms of *domain*, whereas physics-based and domain-specific models are limited to narrow or closed environments, our data-driven approach generalizes to open-set, unconstrained scenes. Finally, rather than relying on a *single-shot* paradigm, we employ *step-wise* auto-regressive reasoning, enabling efficient, interpretable, and long-horizon motion prediction, including branching, without the need for dense reconstruction at each step. This combination of low visual tax, user-controlled granularity, open-set domain coverage, and step-wise paradigm distinguishes our method from the existing literature.

3. Methodology

We consider a single reference frame $\mathcal{I}_0$ at time $t = 0$. Given a sparse set of K visible query points $\mathbf{x}_0 := \{x_0^{(i)}\}_{i=1}^K$, with $x_t^{(i)} \in \mathbb{R}^2$, the goal is to model a distribution over their full future trajectories

$$p(\underbrace{\mathbf{x}_{t=1}, \mathbf{x}_{t=2}, \ldots, \mathbf{x}_{t=T}}_{=:\,\mathbf{x}_{1:T}} \mid \mathbf{x}_0, \mathcal{I}_0), \tag{1}$$

in the same 2D reference frame, assuming a static camera. This joint distribution captures the independent evolution of trajectories, their interactions, and their interdependencies. We model incremental motion at each timestep $\Delta x_t^{(i)} := x_{t+1}^{(i)} - x_t^{(i)}$, with trajectories obtained by accumulating increments over time starting from $\mathbf{x}_0$. Optionally, an initial motion hint ("poke") $\Delta x_0^{(i)}$ can be provided as conditioning to guide the predicted trajectories.

Autoregressive Formulation. We parametrize the joint with an autoregressive transformer [107] p_θ, factorizing causally over time and, within each step, over trajectories, as

$$\begin{aligned} & p_\theta(\mathbf{x}_{1:T} \mid \mathbf{x}_0, \mathcal{I}_0) \\ &= \prod_{t=1}^{T} p_\theta(\mathbf{x}_t \mid \mathbf{x}_{<t}, \mathcal{I}_0) && \triangleright \text{ Time} \\ &= \prod_{t=1}^{T} \prod_{i=1}^{K} p_\theta(x_t^{(i)} \mid \mathbf{x}_t^{(<i)}, \mathbf{x}_{<t}, \mathcal{I}_0). && \triangleright \text{ Trajectories} \end{aligned} \tag{2}$$

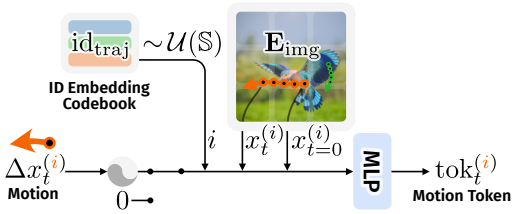


Figure 2. **Motion Token Construction.** The fourier-embedded motion $\Delta x_t^{(i)}$ (alternatively a zero-vector) is combined with a per-trajectory unique randomized trajectory identifier $\mathrm{id}_{\mathrm{traj}}^{(i)}$ and the local image features, retrieved at the current $x_t^{(i)}$ and original position $x_{t=0}^{(i)}$, providing information about what it is and local context.

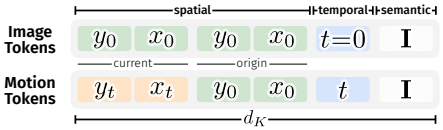


Figure 3. **Positional Encoding Scheme.** We encode the current and original spatial position of each token, alongside its time. Motion tokens attend to each other and to image tokens.

This factorization reflects how humans often reason step by step temporally [9, 34, 120] and makes the interdependence between trajectories explicit by conditioning each update on all previously realized points at the current time and the full past. Importantly, this formulation enables fast decoding with KV caching. In practice, the model predicts $\Delta x_t^{(i)}$ and updates $x_t^{(i)}$ online. We encode the image $\mathcal{I}_0$ into spatial features $\mathbf{E}_{\mathrm{img}}$ via an encoder [30] $\mathcal{E}_\psi$ with parameters ψ.

Motion Tokens. Each motion token corresponds to a particular (t, i) pair and aggregates three kinds of information. First, we retrieve appearance ("what") from the spatial image features $\mathbf{E}_{\mathrm{img}}$ at the trajectory's *origin* $x_0^{(i)}$ using bilinear sampling. Similarly, we retrieve local context ("where") from the features at the *current* position $x_t^{(i)}$. Second, we encode current motion $\Delta x_t^{(i)}$ as a Fourier embedding [68, 99] when observed; for query tokens, we substitute a zero vector of the same dimension. Third, we encode identity ("who") by a trajectory-specific vector $\mathrm{id}_{\mathrm{traj}}^{(i)} \in \mathbb{R}^d$, which we find to be critical for successful modeling in multi-trajectory settings. Rather than using a finite codebook, we draw $\mathrm{id}_{\mathrm{traj}}^{(i)} \sim \mathcal{U}(\mathbb{S}^{d-1})$ (the unit sphere in $\mathbb{R}^d$) each iteration. Random unit-sphere directions yield nearly orthogonal IDs, scale to arbitrary K, and prevent the model from becoming overly reliant on specific indices. We fuse these three sources into the motion token $\mathrm{tok}_t^{(i)} \in \mathbb{R}^{d_{\mathrm{model}}}$ using a small MLP. We show an illustration of the whole mechanism in Fig. 2.

Shared Spatiotemporal Positional Encoding. Motion and image tokens share one reference coordinate frame, so we apply a single positional encoding scheme to both. We base our positional encoding on axial RoPE [27, 97]. Each motion token receives spatial encodings for the *current* position $x_t^{(i)}$, the *origin* $x_0^{(i)}$, plus time t. Image tokens use the same position at $t = 0$ for both 2D position slots. This way, motion tokens can attend to both context about them ("what") at their original location, and local context ("where") at their current position. Finally, we reserve a slice of channels without positional encoding to enable global (semantic) attention [7, 27]. We illustrate the layout in Fig. 3.

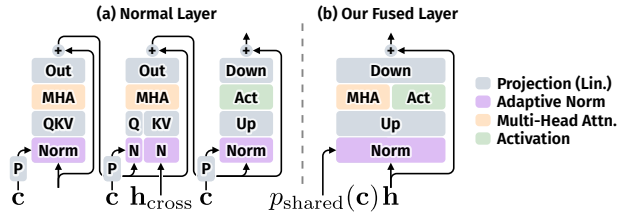


Figure 4. **Fast Reasoning Blocks. (a)** Previous methods [cf. 10] use normal transformer layers, incurring significant overhead due to the multitude of operations performed per block. **(b)** Our fused layers reduce complexity significantly, improving efficiency.

Fast Reasoning Blocks. We aim to explore a multitude of motion hypotheses efficiently, so we design the backbone for high rollout throughput. Instead of evolving the hidden state $\mathbf{h}$ using normal sequential transformer layers, i.e.,

$$\begin{aligned}
\mathbf{h} &\leftarrow \mathbf{h} + \mathrm{SA}(\mathbf{h}), && \triangleright \text{ Self-Attention} \\
\mathbf{h} &\leftarrow \mathbf{h} + \mathrm{CA}(\mathbf{h}, \mathbf{h}_{\mathrm{cross}}), && \triangleright \text{ Cross-Attention} \\
\mathbf{h} &\leftarrow \mathbf{h} + \mathrm{FFN}(\mathbf{h}), && \triangleright \text{ Feedforward Network}
\end{aligned}$$

we adopt parallel transformer blocks [113] with one residual:

$$\mathbf{h} \leftarrow \mathbf{h} + \mathrm{SA}(\mathbf{h}) + \mathrm{CA}(\mathbf{h}, \mathbf{h}_{\mathrm{cross}}) + \mathrm{FFN}(\mathbf{h}). \quad (3)$$

We share pre-normalization and fuse projections such that one "up" computes QKV and FFN-up, and one "down" merges attention and FFN outputs. Further, we combine self- and cross-attention in a prefix layout, concatenating $[\mathbf{h}_{\mathrm{image}}|\mathbf{h}_{\mathrm{motion}}]$ and masking such that image tokens attend to nothing (emulating cross-attention, unlike previous approaches [32, 81], that modify these tokens over depth) and motion tokens attend (causally) to both streams. This cuts down kernel launches significantly. The final fused step becomes

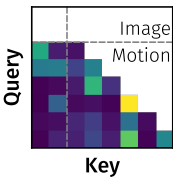


Figure 5. Our attention mask.

$$\mathbf{h} \leftarrow \mathbf{h} + \mathrm{Down} \circ \left[\frac{\mathrm{MHA}}{\mathrm{Act}}\right] \circ \mathrm{Up} \circ \mathrm{Norm}(\mathbf{h}, p_{\mathrm{shared}}(\mathbf{c})), \quad (4)$$

with conditioning implemented via adaptive norms [48] with a shared [25] control vector $p_{\mathrm{shared}}(\mathbf{c})$, mapping the (optional) model condition $\mathbf{c}$. We show a comparison of our blocks with a typical layer structure in Fig. 4.

Posterior Parametrization with Flow Matching (FM). We parametrize the conditional in Eq. (2) as a distribution over stepwise motion $\Delta x_t^{(i)}$. A flow matching [65] head [cf. 60]

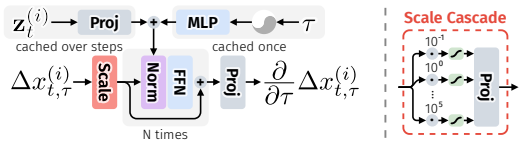


Figure 6. **Posterior FM Head.** *Left:* Our FM Head consists of multiple FFN blocks conditioned on $\mathbf{z}_t^{(i)}$ and flow matching time τ via adaptive norms [48]. We set up the conditioning mechanism such that every component can be cached, reducing computations. *Right:* Our multiscale, tanh-saturated input stack helps stabilize behavior when modeling motion with heavy-tailed behavior.

v_ϕ predicts the ODE velocity of a noisy motion $\Delta x_{t,\tau}^{(i)}$ as it evolves from $\tau = 0$ (Gaussian prior) to $\tau = 1$ (data):

$$v_\phi : (\Delta x_{t,\tau}^{(i)}, \tau, \mathbf{z}_t^{(i)}) \mapsto \frac{\partial}{\partial \tau} \Delta x_{t,\tau}^{(i)}, \tag{5}$$

with parameters ϕ. The AR backbone maps the conditioning to a compact representation $\mathbf{z}_t^{(i)}$ that conditions the head. We set up the head architecture such that separate branches encode τ and $\mathbf{z}_t^{(i)}$ (see Fig. 6), enabling caching instead of recomputation at every sampling step. Compared to parametrizing the distribution using GMMs [10, 103], we find that this leads to both significantly faster convergence during training and significantly more accurate predictions.

Scale Cascade. Motion shows significant heavy tail-like behavior, unlike typical image distributions for which similar heads were previously applied [33, 60], with excess kurtosis κ in the hundreds instead of around 0. We account for this using a high-variance noise prior, setting $\sigma_{\text{noise}} \gg \sigma_{\text{data}}$, and help the head deal with the large range of value scales present on the input side. Specifically, we create a cascade of logarithmically spaced scale coefficients $\mathbf{s}$ and feed $\tanh(\mathbf{s} \cdot \Delta x_{t,\tau}^{(i)})$ component-wise to the head, where small scales preserve fine motion detail while large scales saturate, bounding the influence of rare extremes (see Fig. 6, right). This gives the network stable features for tiny motions and large jumps at once, without letting outliers dominate.

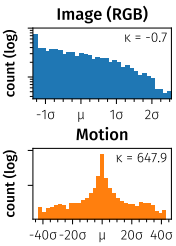


Figure 7. Value distribution.

Objective and Training. We train with teacher forcing [10] and maximize the likelihood in Eq. (2) through the augmented ELBO defined by the flow matching loss [65]

$$\mathcal{L}_{\text{FM}} = \mathop{\mathbb{E}}_{\tau, \Delta x_{t,0}^{(i)}, \Delta x_{t,1}^{(i)}} \| v_\phi(\Delta x_{t,\tau}^{(i)} | \mathbf{z}_t^{(i)}) + \Delta x_{t,0}^{(i)} - \Delta x_{t,1}^{(i)} \|_2^2. \tag{6}$$

We train the FM head v_ϕ, AR transformer p_θ, and image encoder $\mathcal{E}_\psi$ end-to-end, jointly optimizing (θ, ψ, ϕ). Supervision is obtained from videos with (pseudo-)ground truth trajectories obtained, e.g., from off-the-shelf trackers [53, 123].

Inference. We decode step by step with KV caching following the factorization in Eq. (2). For each (i, t), the transformer predicts $p_\theta(\Delta x_t^{(i)} \,|\, \mathbf{x}_t^{(<i)}, \mathbf{x}_{<t}, \mathcal{I}_0)$ via $\mathbf{z}_t^{(i)}$. Sampling $\Delta x_t^{(i)}$ is done by solving the ODE defined by $v_\phi(\,\cdot \mid \mathbf{z}_t^{(i)})$.

Figure 8. **OWM Composition.** We curate OWM to cover a wide variety of settings. *Top:* dataset statistics. *Bottom:* some examples.

4. Benchmarking Efficient Open-World Motion Prediction

Open-world scenes are messy, and ambiguous, but, more importantly, realized only once as we only ever observe a single future. Therefore, to properly evaluate open-world motion prediction, one must assess not a point estimate, but rather the *distribution* of all feasible trajectories that are consistent with the observed future. To make such distribution evaluations feasible, given only a single ground truth observation, the distribution of plausible motion has to be limited in complexity. To this end, we curate a diverse open-world benchmark dataset for motion prediction under a static-camera assumption to remove viewpoint confounders.

4.1. Data

OWM. We curate a set of 95 diverse in-the-wild videos selected for varied motion dynamics. For each scene, we provide a reference frame $\mathcal{I}_0$, query points $\mathbf{x}_0$ with the observed ground truth motion $\mathbf{x}_{1:T}$ for a duration between 2.5s and 6.5s (obtained using off-the-shelf trackers and verified to be accurate). The cameras are verified to be static to enable objective evaluation of predicted scene motion. We show composition statistics in Fig. 8. OWM is solely used for evaluation and will be made publicly available.

Physical Diagnostics Sets. We supplement OWM with two additional sets of videos in more constrained settings, focusing on simple physics motion principles. We source these sets from PhysicsIQ [69], specifically the "solid mechanics" subset, and Physion [11], and manually annotate reference frames and query points consistent with OWM.

4.2. Efficient Motion Hypothesis Generation

Task. Given a single RGB input image $\mathcal{I}_0$ and a short warm-up hint h_0 (the motion over the first 2 frames $\mathbf{x}_{0:2}$), predict a *set of future trajectory samples* for provided query points $\mathbf{x}_0$ over timesteps $t = 1, \ldots, T$. When evaluating video generation models, we provide the hint as full additional

frames and obtain trajectories from generated videos using off-the-shelf point trackers.

Hypothesis Generation. We report results under two standardized budgets:

1. **Best-of-N.** Sample $N = 5$ sets of trajectories, and evaluate the closest to the ground truth observation.
2. **Best-within-Timelimit (primary).** Allocating fixed wall-clock on a reference GPU per scene (5min on an Nvidia H200 to enable the evaluation of video models), methods may generate *any number of hypotheses*, which are subsequently evaluated following the *Best-of-N* setting. This setting enables measuring *search efficiency*.

Further implementation details are specified in the appendix.

Metrics. From the multiple generated hypotheses, we compute prediction error via the pointwise distance of each predicted trajectory $\{\hat{\mathbf{x}}_{n,1:T}\}_{n=1}^{N}$ with the ground truth observation $\mathbf{x}_{1:T}$, using the mean distance over the prediction horizon T for the closest trajectory

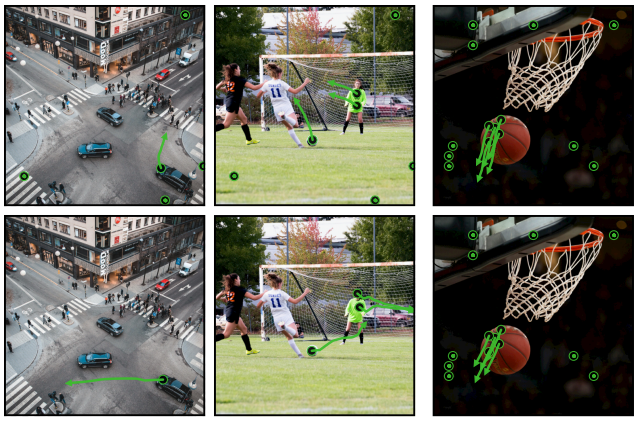
$$\text{minADE}_N = \min_k \Big[\frac{1}{KT} \sum_{i=1}^{K} \sum_{t=1}^{T} \big\| \hat{\mathbf{x}}_{n,t}^{(i)} - \mathbf{x}_t^{(i)} \big\|_2^2 \Big], \quad (7)$$

akin to a one-sided Wasserstein distance over motion space. This captures whether the distribution covers the true outcome without penalizing alternative plausible futures.

5. Experiments

5.1. Implementation Details

We use L-scale transformers [107] for both the motion model and the image encoder, the latter of which we initialize with DINOv3-L/16 [96], with input resolution 512^2. Our flow matching head shares its width with the motion model and has a depth of 3. In total, we have 665M trainable parameters. We train using bfloat16 mixed precision with AdamW [56, 67] with a peak learning rate of 3e-5, betas (0.9, 0.99), and weight decay 0.01. The learning rate is linearly warmed up over the first 5k steps with subsequent linear decay to 1e-8. In general, we train with a global batch size of 128 scenes, using $K = 16$ trajectories and $T = 16$ timesteps for 400k steps, taking about 20 hours to converge on 16 Nvidia H200 GPUs. We primarily train our models on a diverse dataset of 10M open-set video clips collected from the internet, with pseudo ground-truth motion obtained using TAPNext [123]. We additionally train a model using 3D tracks obtained using V-DPM [98]. These tracks are then projected to the first camera view to induce a static camera, enabling direct learning of scene motion disentangled from camera motion. These models are trained on a smaller subset of $\sim$1.5M clips due to the high cost of running such tracker models. For planning tests, we train separate models on data obtained from a billiard simulation [31]. Further details and ablations are in supplementary Secs. A and B.



(a) Diverse Actions from a Single Image. (b) Object Coherence.

Figure 9. **(a)** Given different input pokes (initial motion), our model produces different motions (visualized as **green** lines) that adhere to constraints of the environment. **(b)** Our model predicts coherent motion for multiple points on the same object.

5.2. Motion Prediction

We evaluate our model's ability to predict motion in intricate real-world scenes using the OWM dataset in Tab. 1a. Using the same number of trials for all models in the **Best-of-5** setting, our approach generates a prediction that matches the observed motion with a higher degree of accuracy than state-of-the-art video generation models. Therefore, our approach is able to capture realistic motion more accurately than prior methods while being substantially faster and using significantly less parameters. Under a constrained inference time budget in the **Best-within-5 min**, our approach has a strong advantage due to its orders of magnitude better efficiency achieved by avoiding the visual tax of RGB world simulation, leading to a substantial widening of the accuracy gap. In addition to OWM, we further evaluate the physical understanding of our model on the PhysicsIQ [69] and Physion [11] subsets in Tab. 1b-c. Similar to the open-world setting, we find that our model is competitive with or outperforms state-of-the-art video models in the **Best-of-5** setting already, with the gap widening if time constraints are used.

Qualitative samples in Fig. 9a show our model's capability to produce motion that is informed by visual cues in the scene. The motion rollouts respect constraints and adhere to specific kinematics of the objects visible in the scene. This also applies when predicting the motion of multiple points that move together in the context of the scene (see Fig. 9b).

5.3. Action Selection by Envisioning Futures

We push beyond passive motion prediction and test whether the our model can be applied to choosing an action that leads to a desired outcome, in a fully zero-shot manner. In billiard terms: can it plan a shot? Unlike pure forward prediction compared to one observed future , this setting forces exploration of counterfactual futures – many possible

Method	Param	Throughput (samples/min) ↑	(a) OWM		(b) PhysicsIQ [69]		(c) Physion [11]	
			BEST-5↓	BEST-5MIN↓	BEST-5↓	BEST-5MIN↓	BEST-5↓	BEST-5MIN↓
MAGI-1 [100]	4.5B	0.303	0.037	0.066	0.126	0.169	0.061	0.081
Wan2.2 [112]	14B	0.141	0.039	DNF	0.116	DNF	0.069	DNF
CogVideo-X 1.5 [118]	5B	0.051	0.051	DNF	**0.100**	DNF	0.063	DNF
SkyReels V2 [24]	1.3B	0.304	0.058	0.068	0.128	0.137	0.069	0.084
SVD 1.1 [16]	1.5B	0.714	0.054	0.119	0.138	0.241	0.070	0.147
Myriad (Ours)	665M	**2200**	**0.029**	**0.013**	0.115	**0.045**	**0.048**	**0.020**
Myriad$_{\text{trained on 3→2D Tracks}}$	665M	2200	0.036	0.020	0.117	0.043	0.048	0.028

Table 1. **Open-world & Physical Motion Prediction.** We evaluate motion prediction capabilities across both open-world and constrained physical settings using the benchmark introduced in Sec. 4. Eliminating the need to model fine-grained pixel-level details lets our model focus on the dynamics of the scene, making it competitive with state-of-the-art video models in the Best-5 setting across all three subsets, despite having substantially fewer parameters and being substantially more efficient. The gap widens significantly in the efficiency-focused Best-5min setting, driven by the higher throughput.

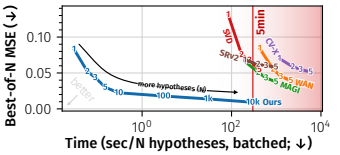


Figure 10. **Time-Accuracy Trade-off on OWM.** Higher numbers of hypotheses N (denoted as numbers in lines) allow more accurate recovery of the observed motion. Across models, the relative improvement in accuracy with N is comparable; the sparsity of our method makes it orders of magnitude more efficient.

actions, many possible rollouts, one desired goal.

Setup. We use a billiard simulator [31] to generate training data and evaluate all methods on an equal footing; every model is trained from scratch at a comparable scale. Each episode starts with a single image of the table, from which the model predicts future trajectories given the initial ball configuration $\mathbf{x}_0$ and an initial cue-ball impulse $\Delta x_0^{(0)}$. A "plan" constitutes selecting an initial strike direction and magnitude $a = (\theta, m)$. We sample a set of candidate actions $\{a_j\}$, predict the corresponding rollouts $\mathbf{x}_{j,1:T}$, and evaluate each rollout using a goal reward $R(\mathbf{x}_{j,1:T})$. This is repeated until a time budget expires, then the plan that maximizes the expected reward is chosen and executed:

$$a^* = \arg\max_{a_j} \mathbb{E}_{\mathbf{x}_{1:T} \sim p_\theta(\,\cdot \mid \mathcal{I}_0, \mathbf{x}_0, a_j)}[R(\mathbf{x}_{1:T})]. \quad (8)$$

Performance is measured by the minimal ℓ_2 distance between the target ball's location and the goal. We calculate the accuracy of solving the task by thresholding the distance using the size of the ball. We provide a visual explanation of the Billiard planning task in Fig. 11-top.

Baselines. We compare against a wide range of baselines representative of common approaches. First, we compare with image-to-video generation methods, starting from an original image, with the initial cue ball impulse specified via either a second frame or a "poke conditioning" mechanism specifying the initial motion. We combine this with either full-sequence video diffusion, following standard video diffusion methods [16, 19, 118], or framewise autoregressive video diffusion [22, 24, 84, 100]. We also include full-sequence trajectory diffusion [cf. 12] and the flow poke transformer [10].

Results. We show our findings in Tab. 2. Compared to image models, sparse trajectory models show at least an order of magnitude improvement in throughput, enabling higher accuracies. At the same time, directly "leaping" to the final state [10], while having the highest throughput, is not accurate enough to enable accurate predictions in such complex settings. Similarly, full trajectory diffusion [12],

Method	Accuracy↑	Throughput (actions/min) ↑
Simulator Oracle	84%	55,162.2
Image to Video Diff. (poke-cond.) [cf. 63, 94]	16%	20.4
Images to Video Diff. [cf. 22, 112, 118]	16%	19.8
AR Image to Video Diff. (poke-cond.)	12%	22.2
AR Images to Video Diff. [cf. 22, 24, 84, 100]	8%	18.6
Full Trajectory Diffusion [cf. 12]	8%	160.8
Flow Poke Transformer [10]	4%	**13,422.6**
Myriad$_{\text{Regression Head}}$	36%	754.6
Myriad$_{\text{GMM Head}}$	24%	753.4
Myriad (Ours)	**78%**	496.4

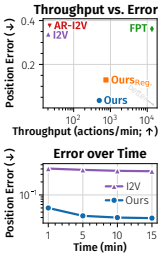


Table 2. **Planning Billiard Shots through Future Exploration.** *Left:* We compare the *Accuracy* of landing a ball at a randomly selected goal position in a billiard simulation by unrolling potential futures starting from varying cue ball impulses. Under a fixed compute budget, our model surpasses dense world models from scratch using the same data. This is enabled by our methods' low *Latency*, enabling us to sample a large number of potential futures. *Right:* We visualize results w.r.t. final target error and show its evolution over planning time for our model and an I2V baseline.

where the model does not gradually unroll the future step by step temporally but immediately has to denoise even steps far in the future, also significantly underperform. Our model combines both sparsity, enabling high throughput by forgoing the "visual tax", and step-by-step unrolling of the future, resulting in the highest accuracy. We also ablate regressing the next step instead of modeling the posterior. For a highly predictable environment like billiard, where little uncertainty is present, this also performs well, although it still underperforms to full distributional modeling. Using a GMM posterior [10, 103] is also worse than our FM head. We visualize our method's planned actions in Fig. 11-bottom.

5.4. Calibration

We explore the relation of our model's posterior uncertainty (as measured by standard deviation on the head's posterior) in Fig. 12. There is a large concentration around pixel-level error (error $< \frac{1}{512}$); above that, the posterior uncertainty predicts the final error well (linear relation in log-log space).

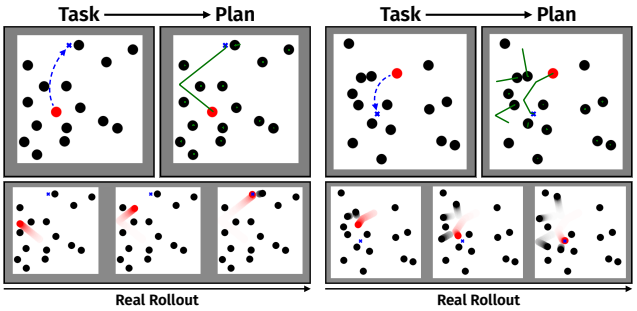


Figure 11. **Planning a Billiard Shot.** We search for a plan to move the red ball to the goal (top left). Our model derives a plan (top right) by predicting motion for different initial actions. Executing the action moves the ball to the desired location (bottom).

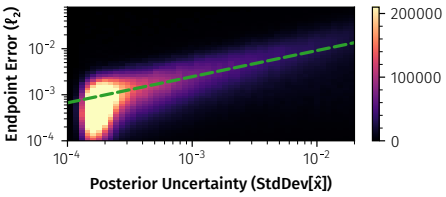


Figure 12. **Posterior Uncertainty vs. Error.** Starting around pixel-level ($\frac{1}{512}$), our model's posterior uncertainty is well-correlated (**green** line) with true error.

5.5. Ablations

We ablate core architectural components, training for 400k steps on our open-set training data. Performance metrics are calculated on OWM following previous experiments.

Fast Reasoning Blocks. We compare the inference speed of our efficient fused attention layers with that of a standard unfused attention layer, using self-attention for motion tokens and cross-attention for image tokens. For a 32-timestep rollout (batch size 4, 16 trajectories), we achieve $\sim$2$\times$ faster sampling, enabling substantially more efficient exploration of the search space. This extends to $\sim$3.7$\times$ at batch size 1.

Posterior Parametrization. Our model uses a point-wise FM head to represent the distribution over future motion. Furthermore, the input to the FM head is scaled using a cascade of exponentially separated value ranges, enabling the model to focus on different granularities of motion as needed. Removing the cascade results in a severe degradation in prediction quality as shown in Tab. 3.

Alternatively, the distribution over future motion could be modeled with a Gaussian Mixture (GMM) Distribution head [103] similar to the single-step Flow Poke Transformer [10]. However, not only is the Gaussian Mixture constrained in what it can represent, leading to higher errors in Tab. 3, the GMM head is also harder to train, converging more slowly as seen in Tab. 3-right.

Posterior Type	Scale Cascade	Best-5↓
GMM [10, 103]	n/a	0.110
FM Head (Ours)	✗	0.033
FM Head (Ours)	✓(Ours)	**0.029**

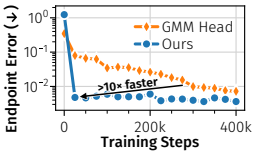


Table 3. **Posterior Parametrization Ablation.** Substituting previously used GMM-based heads with flow matching heads leads to significant improvements in accuracy and increases convergence substantially. Adding our scale cascade improves accuracy further.

6. Conclusion

To envision the many different futures of a scene in a stochastic open world, we have proposed an autoregressive diffusion model that can effectively explore the space of all potential trajectories step-by-step into the future. Our transformer-based model and a lightweight diffusion head model the multi-modal distribution of motion trajectories and allow for efficient training and inference, making our approach especially valuable under compute- and time-constrained settings. The autoregressive approach also naturally lends itself to conditioning motion generation on user-provided initial motion, allowing for the exploration of the effects of actions under uncertainty about how the future will unfold.

To evaluate this setting, we presented *OWM*, a benchmark for open-world motion prediction designed to test whether models can produce coherent, diverse trajectory distributions in realistic conditions. Across diverse domains – from in-the-wild videos to controlled physical setups – our method achieves accurate long-range predictions while dramatically reducing sampling cost, highlighting the advantage of directly modeling motion over future frame generation when accurate dynamics matter. This efficiency of our model further facilitates rapid exploration of the space of possible actions and their outcomes to enable determining optimal action, such as how to select a billiard shot to take to achieve a specific outcome. Taken together, our results highlight the value of a dynamics-centric representation for future reasoning. By focusing on how the world can move rather than how it should look, we provide an efficient, probabilistic mechanism for exploring possible futures – one that can serve as a foundation for forecasting, planning, and interaction in complex real-world environments.

Limitations. Our main formulation assumes a static camera, which simplifies evaluation and improves interpretability of predictions, but limits applicability to scenes with ego-motion or dynamic viewpoints – a setting that contemporary video generation baselines already handle. We explored a formulation that enables *learning* from videos with dynamic cameras by compensating for it during preprocessing, but joint *prediction* of ego and scene motion remains an important direction for future work. Additionally, our model relies on pseudo ground-truth trajectories from off-the-shelf trackers for training, inheriting their biases and failure modes.

Acknowledgments

This project has been supported by a research grant from Netflix, the Horizon Europe project ELLIOT (GA No. 101214398), the project "GeniusRobot" (01IS24083) funded by the Federal Ministry of Research, Technology and Space (BMFTR), the BMWE ZIM-project (No. KK5785001LO4) "conIDitional LoRA", the German Federal Ministry for Economic Affairs and Energy within the project "NXT GEN AI METHODS - Generative Methoden für Perzeption, Prädiktion und Planung", and the bidt project KLIMA-MEMES. The authors gratefully acknowledge the Gauss Center for Supercomputing for providing compute through the NIC on JUWELS/JUPITER at JSC and the HPC resources supplied by the NHR@FAU Erlangen. We thank Timy Phan, Nick Stracke, Kosta Derpanis, Kolja Bauer, Thomas Ressler-Antal, Frank Fundel, Enrico Shippole, Felix Krause, and Meimingwei Li for their helpful feedback and support, and Owen Vincent for continuous technical support.

Author Contributions

SB and JW co-led the project. SB conceived the initial idea (with BO), built the billiard prototype, and optimized the final model. JW developed the final model and handled data processing and evaluation. TM designed, curated, and implemented the OWM benchmark. All authors contributed to writing. MK and BO supervised the project and reviewed the manuscript.

References

[1] Veo: a text-to-video generation system (veo 3 tech report). Technical report, Google DeepMind, 2025. Technical report. 2

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio 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. 3, 5

[3] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37: 58757–58791, 2024. 2

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1, 3

[5] Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: Dino as a foundation for video world models, 2025. 3

[6] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025. 2

[7] Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations*, 2025. 4

[8] Florent Bartoccioni, Elias Ramzi, Victor Besnier, Shashanka Venkataramanan, Tuan-Hung Vu, Yihong Xu, Loick Chambon, Spyros Gidaris, Serkan Odabas, David Hurych, Renaud Marlet, Alexandre Boulch, Mickael Chen, Éloi Zablocki, Andrei Bursuc, Eduardo Valle, and Matthieu Cord. Vavim and vavam: Autonomous driving through video generative modeling, 2025. 2

[9] Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the national academy of sciences*, 110 (45):18327–18332, 2013. 1, 3, 4

[10] Stefan Andreas Baumann, Nick Stracke, Timy Phan, and Björn Ommer. What if: Understanding motion through sparse interactions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2, 3, 4, 5, 7, 8, 1

[11] Daniel Bear, Elias Wang, Damian Mrowca, Felix Jedidja Binder, Hsiao-Yu Tung, RT Pramod, Cameron Holdaway, Sirui Tao, Kevin A. Smith, Fan-Yun Sun, Li Fei-Fei, Nancy Kanwisher, Joshua B. Tenenbaum, Daniel LK Yamins, and Judith E Fan. Physion: Evaluating physical prediction from vision in humans and machines. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 5, 6, 7, 2, 4

[12] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation, 2024. 3, 7

[13] Randolph Blake and Maggie Shiffrar. Perception of human motion. *Annu. Rev. Psychol.*, 58(1):47–73, 2007. 2

[14] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. ipoke: Poking a still image for controlled stochastic video synthesis. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision, pages 14707–14717, 2021. 2

[15] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Bjorn Ommer. Understanding object dynamics for interactive image-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5171–5181, 2021. 2

[16] 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. 2, 7, 5

[17] Gabrijel Boduljak, Laurynas Karazija, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. What happens next? anticipating future motion by generating point trajectories. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[18] Leo Bringer, Joey Wilson, Kira Barton, and Maani Ghaffari. Mdmp: Multi-modal diffusion for supervised motion predictions with uncertainty. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2889–2899, 2025. 3

[19] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. 2, 7

[20] 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

[21] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning*, pages 86–99. PMLR, 2020. 3

[22] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024. 7

[23] Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6178–6189, 2025. 3

[24] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. Skyreels-v2: Infinite-length film generative model, 2025. 2, 7, 5

[25] Junsong Chen, Jincheng YU, Chongjian GE, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. 4

[26] Xinle Cheng, Tianyu He, Jiayi Xu, Junliang Guo, Di He, and Jiang Bian. Playing with transformer at 30+ fps via next-frame diffusion. *arXiv preprint arXiv:2506.01380*, 2025. 2

[27] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Proceedings of the 41st International Conference on Machine Learning*, pages 9550–9575. PMLR, 2024. 4, 1

[28] Etched Decart, Quinn McIntyre, Spruce Campbell, Xinlei Chen, and Robert Wachen. Oasis: A universe in a transformer. *URL: https://oasis-model.github.io*, 2024. 2

[29] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G Derpanis, and Bjorn Ommer. Stochastic image-to-video synthesis using cinns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3742–3753, 2021. 2

[30] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 4, 1

[31] Markus Ebke. python-billiards. https://github.com/markus-ebke/python-billiards, 2025. 6, 7, 1, 2

[32] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 4

[33] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. In *The Thirteenth International Conference on Learning Representations*, 2025. 5

[34] Birte U Forstmann, Roger Ratcliff, and E-J Wagenmakers. Sequential sampling models in cognitive neuroscience: Advantages, applications, and extensions. *Annual review of psychology*, 67(1):641–666, 2016. 4

[35] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards, 2016. 3

[36] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11525–11533, 2020. 3

[37] Ruohan Gao, Bo Xiong, and Kristen Grauman. Im2flow: Motion hallucination from static images for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5937–5947, 2018. 3

[38] Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025. 2

[39] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018. 3, 5

[40] David Ha and Jürgen Schmidhuber. World models. In *Advances in Neural Information Processing Systems 31*, pages 2451–2463. Curran Associates, Inc., 2018. 3

[41] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[42] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[43] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

[44] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. 3

[45] Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025. 2

[46] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. 1

[47] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[48] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 4, 5, 2, 3

[49] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2375–2384, 2019. 5

[50] Miguel Jaques, Michael Burke, and Timothy Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *International Conference on Learning Representations*, 2020. 3

[51] Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong HE, Chiming Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse-AC: Envisioning embodied environments with action condition. 2025. 2

[52] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973. 2

[53] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025. 5

[54] Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-foresight: Looking into the future with DINO. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 2, 3

[55] Georg B Keller and Thomas D Mrsic-Flogel. Predictive processing: a canonical cortical computation. *Neuron*, 100 (2):424–435, 2018. 1

[56] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[57] Kuaishou Technology. Kling: Kuaishou's proprietary text–to–video generation model. https://ir.kuaishou.com/news-releases/news-release-details/kuaishou-unveils-proprietary-video-generation-model-kling, 2024. Press release. 2

[58] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, pages 655–664. Wiley Online Library, 2007. 4, 5

[59] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Puppet-master: Scaling interactive video generation as a motion prior for part-level dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13405–13415, 2025. 2

[60] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *NeurIPS 2024*, 2024. 4, 5, 1

[61] Zhengqi Li, Richard Tucker, Noah Snavely, and Aleksander Holynski. Generative image dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24142–24153, 2024. 3

[62] Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9080–9090, 2025. 3

[63] Jingyun Liang, Yuchen Fan, Kai Zhang, Radu Timofte, Luc Van Gool, and Rakesh Ranjan. Movideo: Motion-aware video generation with diffusion models. In *European Conference on Computer Vision*, 2024. 3, 7

[64] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*, pages 541–556. Springer, 2020. 3

[65] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 4, 5, 1

[66] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision (ECCV)*, 2024. 3

[67] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6, 2

[68] 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. In *European Conference on Computer Vision*, pages 405–421, 2020. 4

[69] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 948–958, 2026. 5, 6, 7, 2, 4

[70] Roozbeh Mottaghi, Hessam Bagherinezhad, Mohammad Rastegari, and Ali Farhadi. Newtonian scene understanding: Unfolding the dynamics of objects in static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3521–3529, 2016. 3

[71] Roozbeh Mottaghi, Mohammad Rastegari, Abhinav Gupta, and Ali Farhadi. "what happens if..." learning to predict the effect of forces in images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 269–285. Springer, 2016. 3

[72] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987, 2022. 3

[73] Jiquan Ngiam, Vijay Vasudevan, Benjamin Caine, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David J Weiss, Benjamin Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2022. 3

[74] OpenAI. Sora 2 system card. https://cdn.openai.com/pdf/50d5973c-c4ff-4c2d-986f-c72b5d0ff069/sora_2_system_card.pdf, 2025. System card. 2

[75] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024. 2

[76] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 3

[77] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc 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. 4, 5

[78] Pika Labs. Pika 2.1. https://pika.art/faq, 2025. Product documentation/FAQ. 2

[79] Silvia L Pintea, Jan C van Gemert, and Arnold WM Smeulders. Déja vu: Motion prediction in static images. In *European Conference on Computer Vision*, pages 172–187. Springer, 2014. 3

[80] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[81] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 4

[82] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 3

[83] Pol Rosello. Predicting future optical flow from static video frames. *Retrieved on: Jul*, 18:2, 2016. 3

[84] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *International Conference on Machine Learning*, pages 42818–42835. PMLR, 2024. 7

[85] Pablo Ruiz-Ponce, German Barquero, Cristina Palmero, Sergio Escalera, and José García-Rodríguez. Mixermdm: Learnable composition of human motion diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12380–12390, 2025. 3

[86] Runway Research. Introducing runway gen–4. https://runwayml.com/research/introducing-runway-gen-4, 2025. 2

[87] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pages 683–700. Springer, 2020. 3, 4, 5

[88] Daniel L Schacter, Donna Rose Addis, and Randy L Buckner. Remembering the past to imagine the future: the prospective brain. *Nature reviews neuroscience*, 8(9):657–661, 2007. 1

[89] Daniel L Schacter, Donna Rose Addis, and Randy L Buckner. Episodic simulation of future events: Concepts, data, and applications. *Annals of the New York Academy of Sciences*, 1124(1):39–60, 2008.

[90] Daniel L Schacter, Roland G Benoit, and Karl K Szpunar. Episodic future thinking: Mechanisms and functions. *Current opinion in behavioral sciences*, 17:41–50, 2017. 1

[91] Rami Seid. Lucid v1. https://ramimo.substack.com/p/lucid-v1-a-world-model-that-does, 2024. 2

[92] Martin EP Seligman, Peter Railton, Roy F Baumeister, and Chandra Sripada. Navigating into the future or driven by the past. *Perspectives on psychological science*, 8(2):119–141, 2013. 1

[93] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 1

[94] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. *SIGGRAPH 2024*, 2024. 3, 7

[95] Joonghyuk Shin, Daehyeon Choi, and Jaesik Park. Instantdrag: Improving interactivity in drag-based image editing. In *SIGGRAPH Asia 2024 conference papers*, pages 1–10, 2024. 3

[96] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 6

[97] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4, 1

[98] Edgar Sucar, Eldar Insafutdinov, Zihang Lai, and Andrea Vedaldi. V-dpm: 4d video reconstruction with dynamic point maps, 2026. 6, 2

[99] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 4

[100] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025. 2, 7

[101] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[102] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[103] Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision*, pages 292–309. Springer, 2024. 5, 7, 8

[104] Tomer D Ullman, Elizabeth Spelke, Peter Battaglia, and Joshua B Tenenbaum. Mind games: Game engines as an architecture for intuitive physics. *Trends in cognitive sciences*, 21(9):649–665, 2017. 1

[105] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[106] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7814–7821. IEEE, 2022. 3

[107] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 3, 6, 1

[108] Rahul Venkatesh, Honglin Chen, Kevin Feigelis, Daniel M Bear, Khaled Jedoui, Klemen Kotar, Felix Binder, Wanhee Lee, Sherry Liu, Kevin A Smith, et al. Understanding physical dynamics with counterfactual world modeling. In *European Conference on Computer Vision*, pages 368–387. Springer, 2024. 2

[109] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 2

[110] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106, 2016. 3

[111] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *Proceedings of the IEEE international conference on computer vision*, pages 2443–2451, 2015. 3

[112] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 7, 2, 5

[113] Ben Wang and Aran Komatsuzaki. Gpt-j-6b: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax, 2021. 4

[114] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam

Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. 2

[115] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015. 3

[116] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual deanimation. *Advances in neural information processing systems*, 30, 2017.

[117] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 3

[118] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan.Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 7, 5

[119] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[120] Jeffrey M Zacks and Khena M Swallow. Event segmentation. *Current directions in psychological science*, 16(2):80–84, 2007. 4

[121] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[122] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on robot learning*, pages 895–904. PMLR, 2021. 3

[123] Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi SM Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9693–9703, 2025. 5, 6, 2, 3

[124] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*. 2, 3

[125] Mo Zhou, Jianwei Wang, Xuanmeng Zhang, Dylan Campbell, Kai Wang, Long Yuan, Wenjie Zhang, and Xuemin Lin. Probdiffflow: an efficient learning-free framework for probabilistic single-image optical flow estimation. *Frontiers of Computer Science*, 20(8):2008342, 2026. 3

[126] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025. 2

Envisioning the Future, One Step at a Time

Supplementary Material

A. Additional Implementation Details

We provide more context on implementation details of our main model described in the paper. Please also refer to the supplementary model code, which contains extensive further comments, for reference.

A.1. Transformer Block

We implement our transformer [30, 107] blocks primarily following the standard Llama [101, 102]-style block architecture in a similar setup as Baumann et al. [10]. Specifically, we use pre-normalization with RMSNorm [121], omit bias terms in linear layers, and use rotary positional embeddings [97] in an axial setup with scaled cosine similarity attention following Crowson et al. [27]. Our feedforward network setup does not follow Llama's SwiGLU [93] activation, but instead uses the more classical GELU [46], while still retaining the omission of bias terms. We observed that both choosing GELU with the typical $\tanh$ approximation as the activation and omitting the GLU-style [93] gating leads to small speed improvements without significant decreases in quality. Importantly, we implement a fully fused parallel transformer layer, where cross-attention and self-attention are combined into a single attention across both kinds of tokens, and projections are shared between the attention and feedforward network, as described in the main paper.

A.2. Posterior Flow Matching Head

Our flow matching posterior head follows similar high-level hyperparameters as Li et al. [60], with three layers of width 1024. Unlike them, we use a standard flow matching [65] objective instead of the DDPM [47] formulation and perform substantial architectural changes to enable efficient sampling. Each block is a standard pre-LayerNorm [4] FFN block with GELU [46] activation.

Conditioning. We implement conditioning such that every component can be cached. Typically, conditioning would be implemented with a local, per-layer MLP that projects a conditioning vector into channel scales, shifts, and, optionally, output gating coefficients. This causes a large number of extra kernel launches, which, as this head will perform tens to hundreds of forward passes per AR sampling step, would cause significant wall-clock overhead. Instead, we precompute all scales and shifts centrally once. Additionally, we factorize the conditioning on flow matching time τ and the conditioning on the parameters $\mathbf{z}_t^{(i)}$ additively, such that the time conditioning can be precomputed offline once, and the parameter conditioning can be computed once per sampling loop, further reducing computational overhead. Conditioning inside each block is implemented via predicted scale and shift on the output of each pre-LayerNorm [4]. We do not perform output gating.

Input Value "Scale Cascade". For the posterior FM head, we use an input scale cascade to stabilize training when modeling motion. Practically, this is implemented as a logarithmically spaced set of scale coefficients

$$\mathbf{s} = \exp(\text{linspace}(\log(0.1), \log(1e5), \text{num} = 512)), \quad (1)$$

with $\text{linspace}(\min, \max, \text{num})$ denoting the standard numpy/PyTorch operation, using which the features for the noisy input x_τ are computed component-wise as

$$[\tanh(\mathbf{s} \cdot x_{\tau,0}) | \tanh(\mathbf{s} \cdot x_{\tau,1})]\mathbf{W}_{in}^\top, \quad (2)$$

with $[\cdot|\cdot]$ denoting channelwise concatenation, and $\mathbf{W}_{in}$ being the output projection to the transformer's hidden dimension.

Sampling. For sampling, we solve the ODE parametrized by the FM head using an Euler solver with uniform spacing of flow matching time τ, matching our training setting of sampling from a uniform distribution $\tau \sim \mathcal{U}[0, 1]$. Unless specifically noted otherwise, we use 50 sampling steps. During AR sampling, we simply sample one motion sample from the posterior, update the latest position of that trajectory, and then sample the next step defined by the AR factorization, while also conditioning on this new information. This process can be started from partial motion information, initial motion hints (pokes), or no prior motion information.

A.3. Hyperparameters

Tab. A provides a comprehensive list of hyperparameters that describe our training setup and model configuration. We train the open-set motion model for 400k steps with a peak learning rate of 3×10^{-5}. We train with a linear learning rate warmup of 5000 steps, after which we apply a linear learning rate schedule. The training setup for the Billiard simulation is similar, but trajectory positions are obtained from the Billiard physics engine [31] and thus represent ground truth motion instead of tracker annotations. Further, we focus on longer-horizon prediction in the Billiard setup. We train the model to predict 50 timesteps, where each timesteps corresponds to a $\Delta t = 0.01\,s$ interval for 300k iterations.

A.4. Training Data

We use three sources of training data for our models.

Open-set Video Data. To train our model for open-world motion generation, we source diverse videos from the internet, while ensuring no overlap with our evaluation data. We

Parameter	Value		
	Open-set		Billiard
Dataset	Open-Set Videos	Open-Set Video 3→2D	Billiard Simulations
Number of clips	10M	1.5M	–
Tracker	TapNext [123]	V-DPM [98]	Ground-truth
Tracker position seeding	1024 random positions	16,641 grid positions	random ball starting positions
Flow scale	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
Image size	512×512	512×512	512×512
Training track number	16	16	16
Training timesteps	16	16	50
Batch size	128	128	128
Optimizer	AdamW [67]	AdamW [67]	AdamW [67]
Betas	(0.09, 0.99)	(0.09, 0.99)	(0.09, 0.99)
Peak learning rate	3×10^{-5}	3×10^{-5}	3×10^{-5}
Learning rate schedule	linear decay to 10^{-8}	linear decay to 10^{-8}	linear decay to 10^{-8}
Warm-up steps	5k	5k	5k
Total steps	400k	400k	300k
Precision	bfloat16 AMP	bfloat16 AMP	bfloat16 AMP
Total Parameters	665M	665M	665M
GPUs	16 Nvidia H200	16 Nvidia H200	16 Nvidia H200s
Training Time	20 h	20 h	20 h
Depth	24	24	24
Width	1024	1024	1024
Head dim	128	128	128
Normalization	RMSNorm	RMSNorm	RMSNorm
FFN expand factor	4	4	4
Activation	GELU	GELU	GELU
Positional Encoding	see Sec. 3	see Sec. 3	see Sec. 3
Static scene conditioning	Adaptive Norm [48]	–	–
Denoiser width	1024	1024	1024
Denoiser depth	3	3	3

Table A. Hyperparameters of our main models and training setup.

then apply an off-the-shelf tracker [123] to obtain pseudo ground-truth annotations. For training, we center crop images to square resolution, cropping in both axis slightly to avoid border points for which the tracker commonly fails. We then resize frames to a uniform 512×512 resolution.

Reprojected 3D Data. Large-scale open-set videos typically suffer from ego camera motion, limiting the interpretability of trajectories. We aim to train a motion model, predicting interpretable static camera trajectories on unconstrained video data for scalability. We thus apply V-DPM [98] a 3D tracker that also estimates camera motion to open-set videos. Then, we reproject tracks into the first camera view, resulting in stabilized trajectories without camera motion interference. We apply the same center crop and resize.

Billiard Data. Training data for the Billiard simulation is obtained using a billiard physics simulation [31]. Ball positions and velocity are sampled randomly while ensuring balls do not overlap with other balls or the border. The physics engine produces future positions of balls, which are used as tracks to train the model.

A.5. Benchmark Construction

To create the OWM dataset, we source 95 permissively licensed videos from Pexels[1] that have been verified to have a static camera and cover a large variety of different kinds of motion from different kinds of entities (*e.g.*, people, vehicles, animals, objects, ...). We prioritize structured or kinematically constrained dynamics (*e.g.*, articulated bodies, rigid object movement) and avoid stochastic or disconnected movement (*e.g.*, excessive background movement, excessively unconstrained motion). We further manually annotate a start frame and select points of interest on moving objects. Ground truth trajectories are obtained with TAPNext [123] and the tracking quality is manually verified.

We complement our dataset with samples from existing solid mechanics benchmarks with known high complexity. For this purpose, we obtain 97 samples from Physics-IQ [69] (subset "solid mechanics") and 134 samples from Physion [11] (excluding the "Drape" subset because of it's focus on soft-body collisions). We manually verify the correctness of motion in the Physion subset, as we observed some examples with unrealistic physical simulation. We, again, manually select starting frames and query points, and verify the correctness of motion annotations for all the additional samples.

A.6. Metrics

Open-World Motion Prediction. For the open-world and physical motion prediction benchmark, we rely on a simple MSE objective between the ground truth trajectory points $\mathbf{p}_{gt}$ and the predicted trajectory $\mathbf{p}_{pred}$ by evaluated methods, where $\mathbf{p}$ is a sequence of T 2D points $\mathbf{p} \in \mathbb{R}^{T \times 2}$. The ground truth is obtained by applying TapNext [123] to the full original video. As in a given initial configuration, multiple outcomes could be reasonable, we give each method the chance to produce an ensemble of predictions, whereby the ensemble size is $N_{ens} = 5$ for the **Best-of-5** setting and N_{ens} depends on the throughput of each method in the **Best-in-5min** setting. Throughput is calculated using best effort, meaning we utilize optimized implementations and lower-precision calculations when possible.

Billiard Planning. We calculate throughput similarly under optimized settings. To calculate the planning accuracy, we use the best action found during rollouts using the principle from Eq. (8). Then, we perform rollouts of the true Billiard simulation using the found action as the initial motion, while all balls except the action ball are initialized as stationary. A selected action is counted as correct if the target ball at least touches or covers the predefined goal position within the allocated time frame. If not the selected action is counted as incorrect. The accuracy is then calculated by dividing the number of correct actions $N_{correct}$ by the total number of trials N_{total}.

A.7. Baselines

Open-World Baselines. For the open-world and physics evaluation, we compare against five state-of-the-art video generation models: MAGI-1 4.5B [100], Wan2.2 I2V-A14B [112], CogVideo-X 1.5 5B-I2V [118], SkyReels V2 DF 1.3B 540P [24], and Stable Video Diffusion 1.1 (SVD) [16]. We utilize the implementation provided in the diffusers [109, 114] library for Wan, CogVideo-X, SkyReels,

[1] https://www.pexels.com/

and SVD. For MAGI, no diffusers implementation is available as of the writing of the paper, therefore we instead adopt the official repository and checkpoint and use the provided 4.5B distill+quant variant. All models except MAGI are run in I2V mode. Thus, they receive the last known image as conditioning and are tasked to simulate the video rollout. As multiple continuations are possible, we sample the Best-out-of-5 and Best-in-5min motion, respectively, giving the models the chance to explore multiple possible outcomes under uncertainty. For MAGI-1, we run the model in video-2-video mode and provide frames preceding the last known frame as hint conditioning. We subsequently apply TapNext [123] tracking to generated videos to obtain predicted trajectories, which we use to compute metrics.

Billiard Baselines. We compare billiard action search performance against four video generation baselines and two trajectory prediction baselines, which we implement and train from scratch to ensure fair comparison. We match the training setup as closely as possible to the setup for our model.

Video generation models are implemented as image-conditioned spatio-temporal Diffusion Transformers [76]. For efficiency, we utilize latent diffusion [82] and perform diffusion in the latent space of the pretrained VAE from Stable Diffusion-XL [80]. Image-conditioning is achieved by cross-attending to the VAE-produced tokens of the start image. We train four variants of video diffusion models, differing along two axes to cover a variety of previous approaches. Our video diffusion models either use auto-regressive generation or full sequence diffusion. In the former setting, the image conditioning is auto-regressively updated to include the prior N_{hist} images. The auto-regressive video generation model then generates the single next frame, conditioned on the history of previous images. The full sequence diffusion approach, on the other hand, is conditioned solely on the initial image and generates the full video from a single noise sample $x_1 \in \mathbb{R}^{T\times H\times W\times C}$. The models further differ in how they are informed about motion prompts. The *Images to Video* variants receive an additional second conditioning image to which they cross-attend. Note that this is natively supported by AR video generation models, while full sequence diffusion requires modification. Therefore, these models can infer the initial motion from visual cues. The *poke-cond.* models receive the instantaneous flow as an additional conditioning similar to our method. The flow and positions are first embedded using Fourier Embeddings and then passed through a small-scale MLP before being pooled into a fixed-size vector with a linear layer for multiple trajectories. The model is then conditioned on the flow embedding using Adaptive Layer Normalization [4, 48]. We use L-sized DiT backbones [76] for our experiments and train the video diffusion models until convergence.

For the full trajectory diffusion baseline, we ensure a fair comparison by reusing our motion models' backbone, but replacing the auto-regressive point-wise diffusion head with a DiT [76]. The training setup and motion model hyperparameters are consistent with our standard setup; however, we ensure that the model always receives only the first step flow.

For the FPT [10] baseline, we utilize the official implementation. Note that all other models predict step-wise motion, while FPT samples future positions in a single step. We align the horizon of the FPT baseline with that of the step-wise models and predict the final position of the balls at the end of the prediction window.

B. Additional Ablations

In the following, we elaborate further on design choices in our implementation.

B.1. Number of Function Evaluations

We test the impact of using more evaluations of the denoising flow matching head on the endpoint error (EPE) in the Billiard setting. Results in Tab. B show that our approach yields lower endpoint error with more function evaluations. Beyond 10 function evaluations, the benefits begin to diminish. Therefore, for our main evaluations in Sec. 5 we use 50 evaluations to balance quality and speed.

NFEs	Mean-best-of-5-EPE
1	0.00361
5	0.00143
10	0.00140
25	<u>0.00139</u>
50	**0.00138**

Table B. **Inference Time Scaling:** Our approach achieves lower End-Point-Error in the Billiard simulation with more function evaluations of the diffusion head.

B.2. Trajectory ID Embedding

As outlined in Sec. 3 we draw random, (nearly) orthogonal trajectory embeddings $\mathrm{id}_{\mathrm{traj}}^{(i)} \sim \mathcal{U}(\mathbb{S}^{d-1})$ to indicate trajectory correspondence to the model. Other, more common approaches would be to use no explicit embedding and instead only rely on positional embeddings, or to use a learnable trajectory embedding with a fixed-size codebook.

We compare these options in Tab. C on the Billiard simulation data. We find that our randomized embeddings outperform both learnable embeddings (likely attributable to a reduction in the likelihood of the model learning position-related biases) and the setting with no extra embeddings. Importantly, unlike learnable embeddings, the model is capable of zero-shot trajectory number extrapolation from 16 (the number observed during training) to larger and smaller numbers, with minimal performance degradation.

Traj. Emb.	Num. Traj.	Mean-best-of-5-EPE
No Emb.	8	0.00116
	16	0.00150
	24	0.00277
Learnable	8	0.00112
	16	0.00149
	24	not possible
Ours	8	**0.00108**
	16	**0.00141**
	24	**0.00263**

Table C. **Trajectory ID Embedding:** Our trajectory ID embeddings provide lower end-point-error in billiard simulations and enable zero-shot generalization to *both* increased and reduced number of trajectories.

B.3. Multi-Step Reasoning

T	Δt	Num. Steps	Mean-best-of-5-EPE
0.5	0.01	50	**0.00141**
0.5	0.05	10	0.00999
0.5	0.5	1	0.02823

Table D. **Reasoning in multiple steps.** We compare predicting 0.5 s into the future using models trained with different step sizes. Our standard method integrates 50 steps, while the other models perform fewer steps. Therefore, these models require fewer auto-regressive steps, yet have to model more of the dynamics internally.

Our approach predicts the motion over a short time horizon Δt in one step and then auto-regressively samples movement to predict motion over the entire time horizon T, thus factorizing the full motion prediction over ΔT into a sequence of small-step predictions. In theory, a motion model with infinite capacity should be able to predict the final position of all scene elements in a single step by internally accounting for all potential interactions. However, we argue that predicting step-wise motion is a substantially more practically viable task when not assuming abundant model capacity. We investigate this assumption by comparing model variants in the Billiard setting.

We compare our standard model, predicting $\Delta t = 0.01\,s$ into the future per step, against variants predicting over a larger Δt. We perform a 0.5 s rollout (making the largest-step model perform predictions over the full horizon T in a single step, as in [10]) and evaluate the end-point-error of each model. The results in Tab. D show that multi-step motion prediction improves modeling performance, with overall improved performance for smaller step intervals. The single-step model performs significantly worse than both multi-step variants, mirroring the planning results in Tab. 2. We attribute this failure to the complexity of internally modeling and accounting for all interactions in a large Δt timeframe.

B.4. Classic Trajectory Forecasting Setting

We explore our approach's efficacy in classic trajectory forecasting settings in a *zero-shot* setting. We compare on the canonical ETH-UCY [58, 77] benchmark following the setting of Trajectron++ [87]. All baselines are trained exclusively on in-domain data, while we apply our model zero-shot. The baselines directly operate on tracked abstract agents in a 2D top-down view space (obtained from the camera-space tracks via projection), while we operate directly on the original images, as our model uses that as the input. Since the given homographies are not accurate for reprojecting back into the camera space, we manually annotate correspondences and fit homographies, obtaining the equivalent tracks in camera space, which serve as input and output space for our model. ETH generally annotates people's heads, while UCY seems to rely on people's feet. This does not matter in a top-down view, as the head will typically be in the same 2D position as the feet, but it matters in camera space. We annotate homographies to follow the ETH convention. Metrics are computed in the original 2D top-down space directly following the baselines.

We show results in Tab. E. Despite not being trained for this setting, our method achieves competitive results with canonical task-specific baselines. This demonstrates that our much more generic approach can still perform well even in specific settings. With additional finetuning on sufficiently large-scale in-domain data, results should further improve significantly.

B.5. OWM Breakdown

We report metrics for subsets of OWM focusing on specific kinds of motion in Tab. F. Our model is competitive with substantially larger video baselines for all subsets, including intricate multi-agent interactions. In the time constraint setting our method achieves the best results across all subsets as it's fast inference allows to explore a much larger variety of potential futures.

C. Additional Qualitative Samples

Benchmark samples. We provide qualitative samples from the OWM benchmark (Fig. A), Physics-IQ subset (Fig. B), and Physion subset (Fig. C) with the **Best-out-of-5** motion annotation for our approach and all baseline methods.

Qualitatively, our approach predicts motion that is on par with state-of-the-art video generation approaches in open-world settings, found in the OWM benchmark. Comparing on Physics-IQ [69], video generation approaches tend to predict overly simplified, physically implausible trajectories, whereas our method is able to capture the complexity of real-world physical interactions. For Physion [11], state-of-the-art video generation models hallucinate overly complex motion, whereas our approach is able to capture the rigid

Method	ETH		Hotel		Zara01		Zara02	
	Deterministic	Best-of-20	Deterministic	Best-of-20	Deterministic	Best-of-20	Deterministic	Best-of-20
SocialLSTM [2]	1.09/2.35	–	0.79/1.76	–	0.47/1.00	–	0.56/1.17	–
SocialGAN [39]	–	0.81/1.52	–	0.72/1.61	–	0.34/0.69	–	0.34/0.69
Trajectron [49]	–	0.59/1.14	–	0.35/0.66	–	0.43/0.83	–	0.43/0.83
Trajectron++ [87]	**0.71**/1.66	0.39/0.83	**0.22/0.46**	**0.12/0.19**	**0.39/0.77**	**0.15/0.33**	**0.23/0.59**	**0.11/0.25**
Myriad (ours, zero-shot)	0.81/**1.50**	**0.31/0.80**	0.30/0.54	0.17/0.30	0.79/1.75	0.53/1.21	0.58/1.30	0.40/0.91

Table E. **Zero-shot Comparison with Closed-Domain Trajectory Forecasting on ETH-UCY [58, 77].** All numbers (except ours) are sourced from Trajectron++ [87]. Note that some sequences from UCY are missing due to missing RGB videos. Values are (following Trajectron++) (min-){ADE/FDE}, "–" means not reported by Trajectron++. In the "deterministic" setting, we sample from our model once with fixed seed.

Method	Rigidity				Number of Agents				Agents with Free Will				Avg. Rank		Throughput	Params↓
	Rigid		Non-rigid		Single-Agent		Multi-Agent		w/ Free Will		w/o Free Will					
	N=5↓	T=5min↓	N=5↓	T=5min↓	N=5↓	T=5min↓	N=5↓	T=5min↓	N=5↓	T=5min↓	N=5↓	T=5min↓	N=5↓	T=5min↓	samples/min↑	
MAGI-1	0.032	0.058	0.039	0.069	0.020	0.044	0.048	0.080	0.040	0.066	0.030	0.065	2.00	3.00	0.303	4.5B
Wan2.2 [112]	0.042	DNF	0.038	DNF	0.039	DNF	0.039	DNF	0.036	DNF	0.045	DNF	2.33	DNF	0.141	14B
CogVideo-X 1.5 [118]	0.051	DNF	0.051	DNF	0.041	DNF	0.052	DNF	0.049	DNF	0.054	DNF	4.50	DNF	0.051	5B
SkyReels V2 [24]	0.061	0.071	0.056	0.066	0.048	0.056	0.064	0.075	0.054	0.063	0.065	0.076	5.50	3.00	0.304	1.3B
SVD 1.1 [16]	0.048	0.055	0.057	0.073	0.037	0.053	0.065	0.077	0.060	0.069	0.042	0.064	4.66	3.00	0.714	1.5B
Myriad (Ours)	0.031	0.007	0.039	0.016	0.036	0.008	0.044	0.017	0.037	0.014	0.044	0.011	2.00	1.00	2200	0.6B

Table F. **OWM Subset-wise Metrics.** Breakdown of Tab. 1 results. While orders of magnitude faster, our method is consistently competitive with state-of-the-art video models across not only the overall benchmark, but also when split across multiple properties (rigidity of motion, number of agents, presence of agents with free will).

body physics of the benchmark setting. Therefore, our approach is able to balance complexity and simplicity better than previous approaches making it applicable to a wider range of inference contexts.

Open-set samples. We provide samples for a variety of open-set conditioning images, sourced from the internet. Fig. D shows that our approach predicts motion informed by the context provided through the starting image. We provide two examples where we edit the image using nano banana and use the same motion hint. The qualitative samples show that for a person on a trampoline, more bouncy motion is predicted compared to a person jumping on a wooden floor. Further, a ball rolling across a table has a more straight trajectory compared to an egg rolling across the same table.

Fig. E shows samples generated with initial motion hints. The samples show that unrolling hinted trajectories results in consistent motion across entities and realistic long-term behaviour.

In Fig. F we provide samples generated without an initial motion hint. While the trajectories tend to be more simplistic, realistic motion is obtained based solely on the input image. Query points without motion hint are marked in grey.

Fig. G illustrates samples where only a single query point on the object received a hint, and motion for other queries has to be inferred from appearance alone. The results highlight that sampled motion is coherent across objects. Further, our model is able to capture multi-modal behaviour if two outcomes are realistic given the same input. Query points without motion hint are marked in grey while queries with motion hint are colored black.

Billiard samples. We show qualitative predictions from our model trained on billiard simulation data in Fig. H. We show the predicted simulation given an initial impulse in the upper row, and the ground truth simulation overlaid with the prediction in the lower row for each respective sample. Simulation time increases linearly from left to right. Our model is able to accurately predict the ground truth motion, up to stochastic uncertainties.

D. Language Model Usage

We employed large language models (OpenAI GPT-5.2, Claude Opus 4.6) for text refinement purposes, including improving grammar and as inspiration for rephrasing sections. They were also employed to provide feedback on early drafts and propose initial implementations for auxiliary utility functions not directly related to the paper's contributions, subsequently verified and reworked by the authors. No scientific content, experimental results, or novel ideas were generated by LLMs – all technical contributions were conceived, implemented, and verified by the authors.

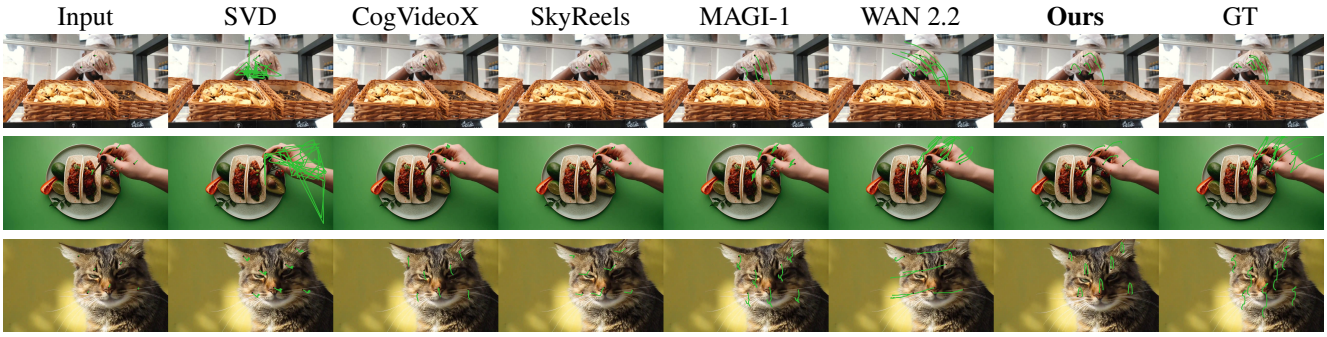


Figure A. **Qualitative comparison on OWM:** Our model produces motion samples that are qualitatively on par with much larger models such as WAN2.2 and MAGI-1.

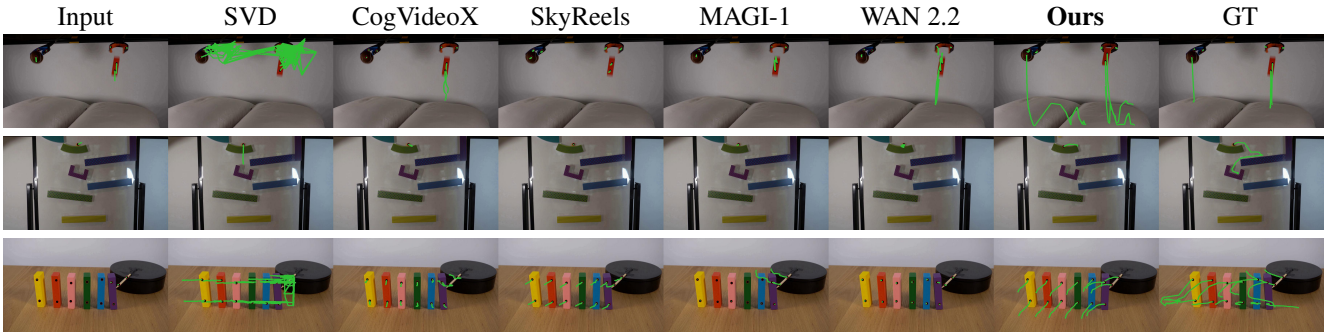


Figure B. **Qualitative comparison on Physics-IQ:** While video generation models fail to capture the complexity of object interactions and predict simplified or no motion, our approach captures realistic physical interactions.

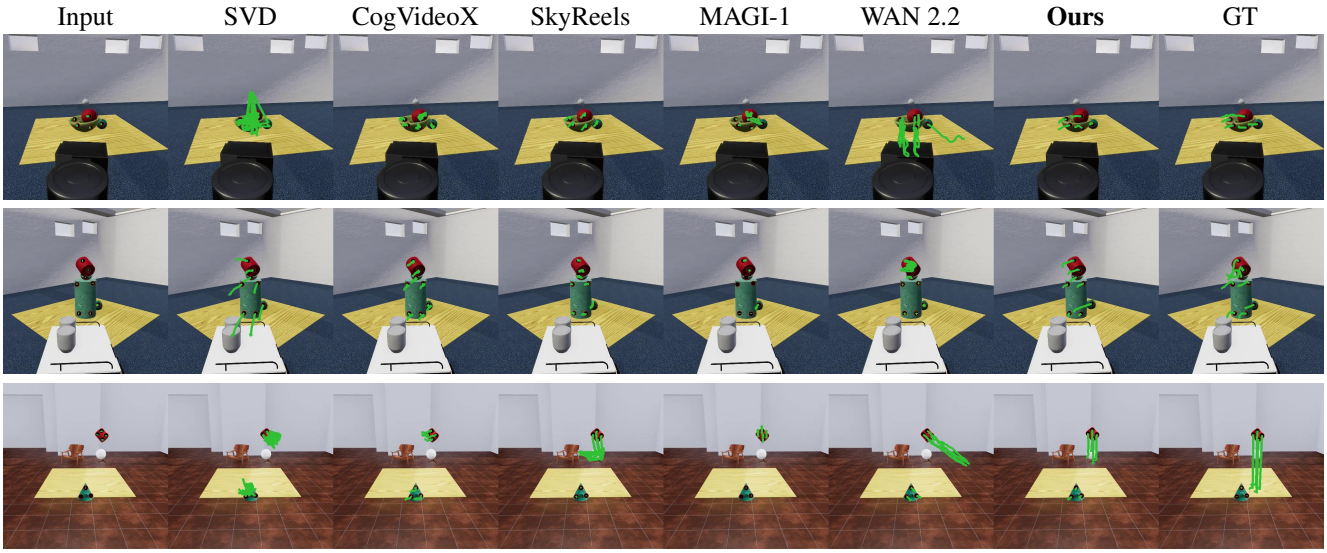


Figure C. **Qualitative comparison on Physion:** For simplified rigid body settings in Physion, video generation models hallucinate overly complex motion, while our approach is able to capture physical dynamics.

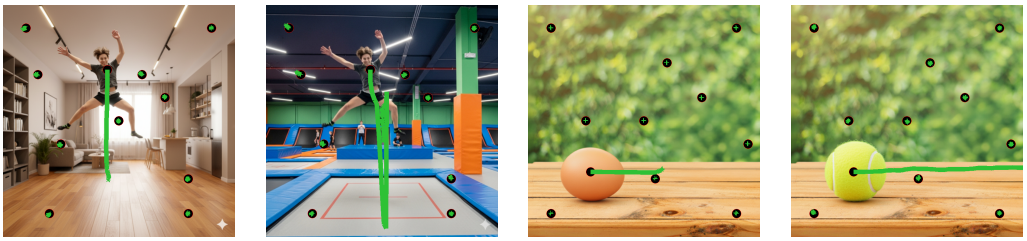


Figure D. **Context informed samples:** The above samples show our model's ability to take appearance information into account when predicting motion. In both comparisons, images were sampled with the same initial poke. Images where edited using nano banana for high similarity in appearance.

Figure E. **Hinted Samples** Our model is capable of producing complex, coherent, and appearance-informed motion given an initial motion hint.

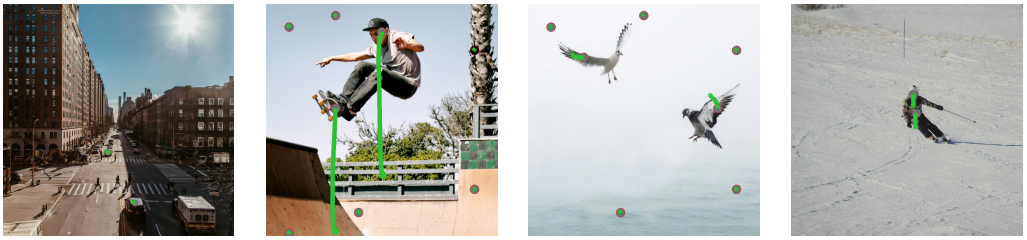


Figure F. **Un-hinted Samples** Given only appearance conditioning, our approach is able to produce physically correct and coherent motion, also show ing more complex understanding, such as that cars at an intersection should *not* move when pedestrians are blocking their path.

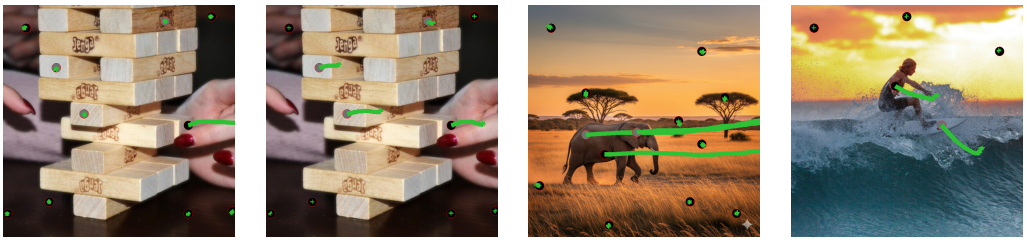


Figure G. **Partially-hinted Samples** Given only a single poke conditioning per example, our model produces coherent motion for queries on the same or linked objects. The Jenga example highlights that our model is able to capture multi-modality if two outcomes are possible given the same initial motion hint.

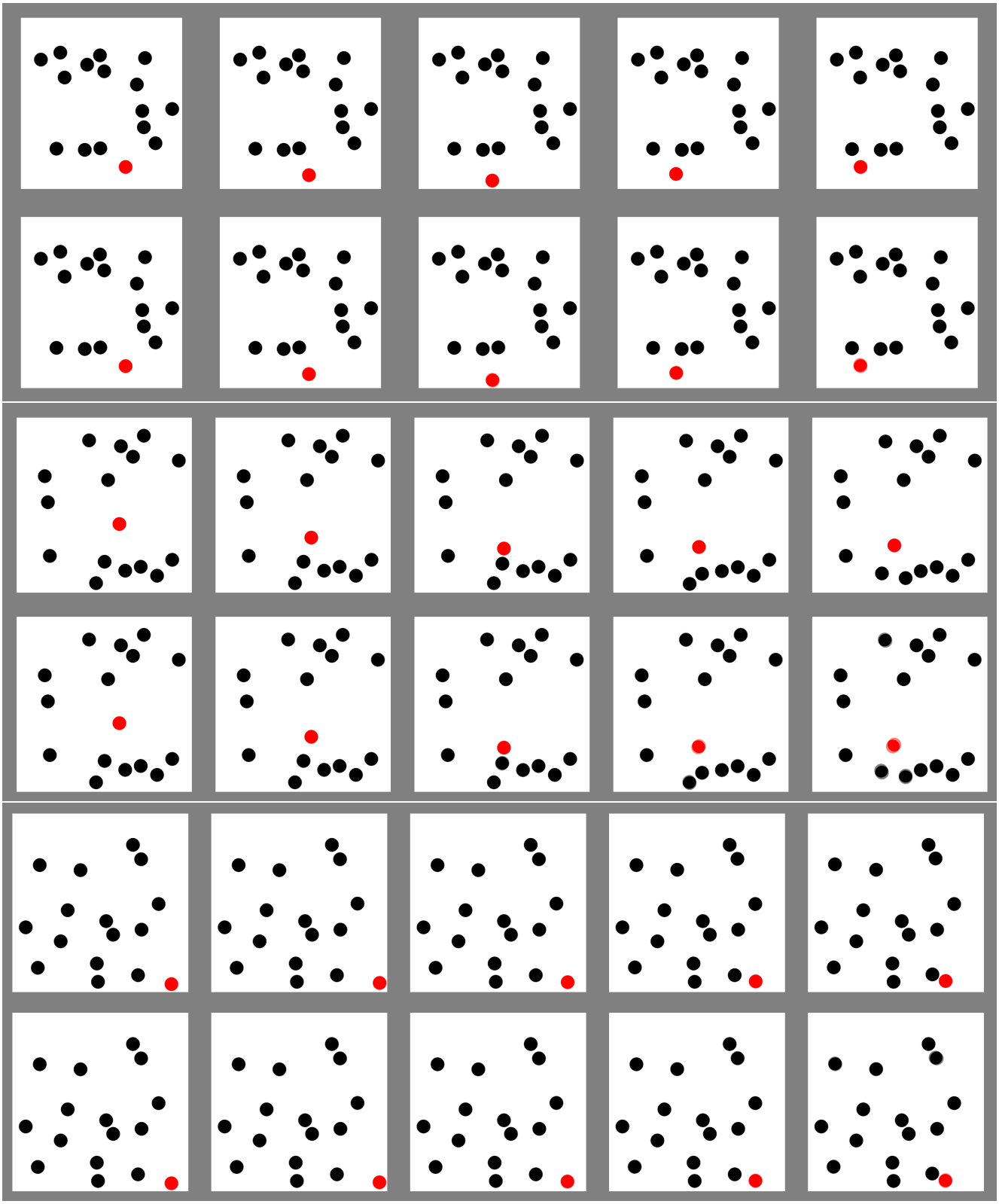


Figure H. **Qualitative samples on our billiard simulation.** The respective top row shows our model's prediction given an initial impulse for the ball marked in red, where we visualize the predicted trajectory state using a frame-wise renderer; the lower row shows an overlay of the ground truth simulation with the prediction to enable comparisons. Our model can successfully predict the observed motion up to minor stochastic details.