Schrödinger's Cat: Probabilistic Representation and Prediction of Potential Scene Kinematics

Timy Phan[*], **Jannik Wiese**[*], **Björn Ommer**
CompVis @ LMU Munich, Munich Center for Machine Learning (MCML)

Predicting how a scene may evolve from partial observations requires reasoning about multiple possible futures rather than committing to a single trajectory. Existing approaches either generate appearance-dominated video predictions or sample a small number of trajectories without explicitly modeling the distribution of possible motion. We introduce *Goal-Aware Representations of Future kInEmatic Latent Distributions* (GARFIELD), a probabilistic model of scene kinematics that learns a *structured spatio-temporal latent representation* of the distribution over possible futures given an image and optional *spatio-temporally sparse constraints*. The same latent representation enables both joint sampling of all trajectories and direct access to the underlying motion distribution through an efficient deterministic density decoder. As a result, uncertainty about future motion can be localized to specific scene elements and timesteps and progressively refined through additional constraints. Experiments demonstrate strong motion planning performance competitive with large video generation models while sampling trajectories 97× faster. Our method further estimates motion densities two orders of magnitude faster than Monte-Carlo sampling from motion generation models, enabling interactive exploration and uncertainty-aware planning.

Project Page: `https://compvis.github.io/schroedingers_cat`
Code: `https://github.com/CompVis/schroedingers_cat`
Accepted at ECCV 2026

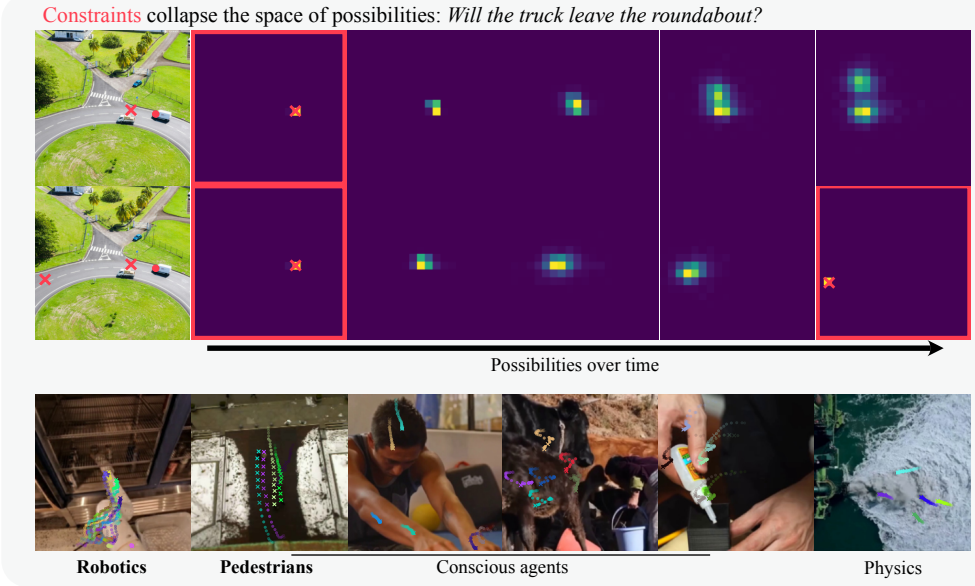


Figure 1: Schrödinger's Cat: Given the initial scene context as an image and a set of *spatio-temporally sparse constraints or goals* (x), GARFIELD encodes the space of possible motions. By sampling from this space, our approach is able to plan realistic motion for conscious agents and inanimate objects, which we **evaluate** in applications such as robotics and trajectory prediction.

[*]Equal Contribution.

1 Introduction

Our world is constantly unfolding and every moment carries multiple possible futures: the future is fundamentally under-determined given the present due to stochasticity, free will, and other hidden factors. Acting in such environments requires anticipating how a scene might change. In particular, we need to predict scene kinematics – how its constituents may move – estimating the probability of *what could possibly be* rather than merely perceiving and tracking *what has already happened*. Video models used as world simulators [4, 6, 23] generate the future as sequences of images, producing individual frames rather than explicitly modeling scene changes between frames. Much of their capacity is spent modeling appearance details that do not affect scene kinematics, inducing unnecessary latency and limiting exploration of possibilities. Modeling motion directly [10, 11, 42] improves efficiency and focuses capacity on kinematics, but existing approaches predict dense motion fields, wasting computational effort on static background instead of targeting relevant scene elements. Moreover, generative motion models sample individual realizations rather than predicting the probability of various potential outcomes, their distribution, and uncertainty.

We take inspiration from Schrödinger's famous thought experiment: under limited observability, the state of a system is represented as a distribution of possible states, which collapses as new observations become available. Analogously, the key idea of *Goal-Aware Representations of Future kInEmatic Latent Distributions* (GARFIELD) is to represent the future as a *distribution over trajectories of scene constituents* that can be *progressively refined* by incorporating additional observations or constraints. Our encoder takes an image of a scene together with possible intermediate or goal positions to produce a latent representation that encodes the distribution over possible future movements of scene elements. More constraints consistently sharpen this distribution, providing users with effective interactive control.

This kinematics distribution must be localized to specific scene elements and timesteps, so that uncertainty can be attributed correctly. We therefore learn a probabilistic embedding of scene kinematics using a joint motion encoder of the entire scene that produces *spatio-temporally localized latent variables*. Training with a generative decoder ensures that the representation captures the space of possible motions rather than a single trajectory. The structured latent space, which allows inspecting possible motion of specific elements at specific times is enforced by training the encoder with a point-wise decoder model. We propose efficient decoders which – based on the same latent representation – enable both (i) joint trajectory sampling that captures interdependencies, as well as a (ii) *deterministic density decoding* that produces probability heatmaps in a single ViT forward pass. The density decoder estimates densities and uncertainty orders of magnitude faster than Monte-Carlo sampling, which is crucial for real-time planning and fast exploration of uncertainty.

We consider the task of *motion planning*: generating trajectories that satisfy sparse goals while respecting scene constraints. Prior work typically relies on conditioning signals such as goal images [10], temporally dense trajectories [54], or text prompts [22, 54, 71], which are either ambiguous or difficult to obtain. In contrast, we enable *spatio-temporally sparse conditioning*, specifying constraints only for selected objects and timesteps while leaving the remaining motion for the model to infer. Coupled with fast density estimation, this enables efficient exploration of possible futures and uncertainty-aware motion planning with minimal user input.

Contributions: We learn a (i) *structured probabilistic latent representation of motion* (Sec. 3.1). The same latent representation enables (ii) jointly sampling *mutually consistent trajectories* that respect provided constraints (Sec. 3.4) and (iii) direct access to the underlying motion distribution via an efficient *density decoder* (Sec. 3.2, Fig. 4). Our model further supports (iv) *flexible spatio-temporally sparse conditioning*, allowing users to shape the distribution of possible futures through sparse constraints (Fig. 5). As a result, *Goal-Aware Representations of Future kInEmatic Latent Distributions* (GARFIELD) enable uncertainty-aware motion planning and interactive exploration of future possibilities (Sec. 3.3, Fig. 3c). Our approach samples full trajectories $97\times$ faster than video world models (Tab. 1) and estimates motion densities *two orders of magnitude faster* than Monte-Carlo sampling from motion prediction methods (Fig. 6b).

2 Related Work

Recently, video world models [4, 6, 13, 22, 23, 62, 71] based on diffusion transformers [44] have become the de facto standard approach to predict future scene evolution. These models are typically conditioned on a start image, text, and sometimes action classes, producing videos that depict future events in full pixel detail. They achieve impressive results in robotics [67, 75], autonomous driving [7, 64, 73], and physical simulation [6, 13, 38], but represent the future as a sequence of images instead of focusing on dynamics. The motion in the generated pixels can be extracted and analyzed post-hoc, but this approach adds additional overhead on top of the already expensive video synthesis.

Low-level temporal dynamics can be represented as dense optical flow [27, 59] or sparse point tracks [29, 74], which capture motion without modeling appearance. Recent methods such as Track2Act [10] and *What Happens Next?* [11] apply diffusion transformers to generate point tracks for motion-centric tasks (e.g. robotic control). Similarly, Motion-I2V [54] generates optical flow from a start image, text prompt, and spatially sparse conditioning tracks. The sampled flow then conditions a video generator to improve quality and control. However, these approaches rely on ambiguous text [11, 56] or unavailable goal-state images [10]. In contrast, GARFIELD conditions on spatio-temporally sparse goal states. Moreover, these methods predict motion on dense grids, wasting compute on mostly static background instead of focusing on relevant scene elements.

In comparatively narrow application domains such as autonomous driving and pedestrian prediction, methods like Social-LSTM [1], Social-GAN [21], and Trajectron++ [51] already focus on predicting motion only for relevant scene elements. This highlights the practical value of element-centric modeling. However, these models are tailored to specific domains and do not generalize to open-set motion prediction conditioned on arbitrary scene appearance.

Tracks and optical flow capture low-level motion but remain agnostic to semantics and constraints. Recent work therefore explores richer motion representations, including training with VAEs [14], normalizing flows [25], or self-supervised representation learning [47, 58, 65]. Motion Diffusion Autoencoders [48] further learn semantic motion representations in closed-domain settings. However, most approaches focus on reconstruction or semantics and do not model the stochasticity of future motion. Early works represented this stochasticity using probabilistic motion primitives [43], Bézier curve distributions [28], or Gaussian processes [30]. Recent diffusion models [9–11, 54, 56] enable sampling possible futures but do not provide direct access to the underlying distribution, requiring expensive Monte-Carlo sampling to estimate distributions. Motion Modes [42] improves exploration of this distribution through guided sampling but remains limited by high inference cost. In contrast, GARFIELD provides access to a non-parametric distribution over future positions in constant ($\mathcal{O}(1)$) time.

The Flow Poke Transformer (FPT) [8] made a significant step towards open-set, distribution-centric motion modeling by predicting the parameters of a Gaussian Mixture Model (GMM) over future positions given a set of spatially sparse known movements. However, the model is limited by the functional form of their output distribution, constraining the output space to those possibilities that can be represented by a GMM. More importantly, their approach only predicts distributions at one specific future timestep and does not account for scene evolution over multiple steps or allow specifying intermediate goals.

We propose GARFIELD to train an open-set motion foundation model that encodes possible future positions of scene elements given appearance cues and optionally sparsely known future goals. We further introduce decoder models that provide access to the underlying non-parametric distribution without repeated sampling and probabilistic sampling of future trajectories. Our general model can be applied to application domains either in a zero-shot setting or with minimal fine-tuning, highlighting the advantages of scalable open-set pretraining.

3 Goal-Aware Representations of Future kInEmatic Latent Distributions

Predicting scene kinematics given a finite set of constraints or goals requires modeling the probability distribution over many plausible futures. We represent one possible realization by the motion of scene elements i over time, described by trajectories $t^i = (t_1^i, t_2^i, \dots)$, where $t_\tau^i \in \mathbb{R}^2$ denotes the position of element i in the 2D image plane at timestep τ. The distribution $p(T)$ of possible futures, with $T = \{t_\tau^i\}_{i,\tau}$, can be narrowed and steered by conditioning on constraints or goals. Analogously to the *"Schrödinger's Cat"* thought experiment, additional constraints or observations collapse the space of possible futures towards trajectories that match the conditioning.

Providing scene context by means of a start frame I and a sparse set of positions $\widehat{T} \subset T$ through which certain trajectories must pass reduces uncertainty by partially controlling scene movement through intermediate goals. *Goal-Aware Representations of Future kInEmatic Latent Distributions* (GARFIELD) therefore aim to model the conditional distribution $p(\bar{T} \mid I, \widehat{T})$ over the unknown trajectory points $\bar{T} = T \setminus \widehat{T}$. For brevity, we summarize the conditioning via start frame and goal positions as $\mathcal{C} = (I, \widehat{T})$.

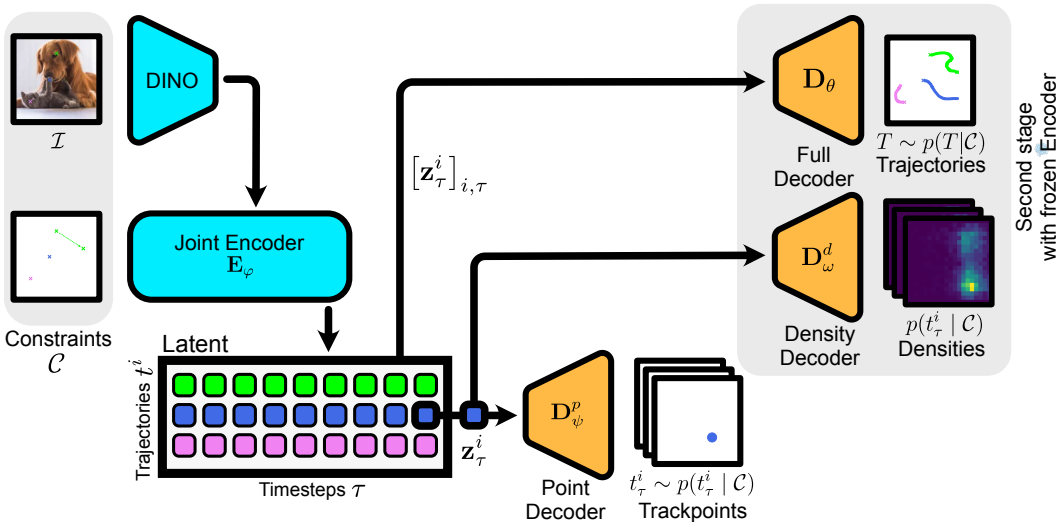


Figure 2: GARFIELD Architecture Overview. Given a set of constraints $\mathcal{C}$, namely an image $\mathcal{I}$ and kinematics $\widehat{T}$, our encoder produces conditional latents $\{\mathbf{z}_\tau^i\}$ representing distributions over possible future kinematics. These can be decoded into point estimates t_τ^i with $\mathbf{D}_\psi^{\text{p}}$, densities $p(t_\tau^i)$ with $\mathbf{D}_\omega^{\text{d}}$, and sets of kinematics T with $\mathbf{D}_\theta$.

3.1 Learning the Distribution of Scene Kinematics

We learn a latent representation of the distribution of plausible scene kinematics given scene context and constraints $\mathcal{C}$. Our goal with GARFIELD is twofold: (i) to jointly sample trajectories of all scene elements, $\bar{T} \sim p(\bar{T} \mid \mathcal{C})$, and (ii) to estimate the probability density $p(t_\tau^i \mid \mathcal{C})$ for the position of a scene element i at time τ. Modeling such point-wise probability densities enables efficient parallel exploration of motion distributions across the spatio-temporal volume and reveals how changes to $\mathcal{C}$ affect individual scene elements at specific points in time.

By annotating video data with an optical tracker [29, 74], we can harness a rich and scalable source of data for learning and understanding scene kinematics. However, the videos and their track annotations are only one possible realization $T \sim p(T)$ of kinematics for the respective scene I. This leaves the distribution of *possible* kinematics out of reach and thus there is no direct ground truth for $p(T)$. Generative modeling is an established way to capture a data distribution without direct access to its underlying density function by learning from samples of that data distribution.

Current methods [9–11, 54, 56] can sample T with generative models which underlines the viability of learning scene kinematics from video data. However, the ability to sample T does not equate to easily accessible densities $p(t_\tau^i|\mathcal{C})$. In theory, the density function $p(t_\tau^i|\mathcal{C})$ can be approximated with Monte-Carlo estimation by drawing multiple realizations of T. In practice, reliable estimates require many samples and the expensive sampling procedure of generative models renders this avenue infeasible.

Because both $\bar{T} \sim p(\bar{T}|\mathcal{C})$ and $p(t_\tau^i|\mathcal{C})$ are conditioned on the same $\mathcal{C}$, we propose to learn a shared latent representation which represents $\mathcal{C}$ and informs both sampling and density estimation. Density estimation $p(t_\tau^i|\mathcal{C})$ needs to be spatio-temporally localized. Thus, we factorize the latent

$$\mathbf{z} = \left[\mathbf{z}_\tau^i\right]_{i,\tau} \tag{1}$$

into separate $\mathbf{z}_\tau^i$ components which each corresponds to a specific point-wise motion distribution $p(t_\tau^i|\mathcal{C})$. This representation is learned end-to-end with a joint encoder $\mathbf{E}_\varphi$, which produces $\mathbf{z}$, and a point-wise decoder $\mathbf{D}_\psi^{\text{p}}$ which samples track positions from the point-wise distribution $t_\tau^i \sim \mathbf{D}_\psi^{\text{p}}(\mathbf{z}_\tau^i) \approx p(t_\tau^i|\mathcal{C})$. We train $\mathbf{E}_\varphi$ and $\mathbf{D}_\psi^{\text{p}}$ jointly to enforce this point-wise structural property in $\mathbf{z}$.

3.2 Sampling-free Density Estimation

Latent components $\mathbf{z}_\tau^i$ represent the distribution of possible positions of scene elements i at time τ. Using the point-wise decoder $\mathbf{D}_\psi^{\text{p}}$ with N_{MC} different initial noise seeds allows sampling N_{MC} track points from the point-wise distribution

and thus enable a Monte-Carlo (MC) estimation of the outcome probabilities. However, this presents a trade-off: accurate estimation requires many samples, leading to substantial computational overhead.

To circumvent this trade-off, we propose a deterministic density estimator $\mathbf{D}_\omega^{\mathrm{d}}$ for $\mathbf{z}_\tau^i$ with GARFIELD. We define the marginal probability density on a regular grid $\mathcal{G} = \{1, \dots, g\} \times \{1, \dots, g\}$ with $g \in \mathbb{N}$ as

$$p_\omega^{(u,v)}(\mathbf{z}_\tau^i) = \mathrm{P}\big(t_\tau^i \in \mathcal{B}_{u,v} \mid \mathbf{z}_\tau^i\big)\,, \tag{2}$$

which gives the probability that t_τ^i lies within the spatial region $\mathcal{B}_{u,v}$ corresponding to grid location $(u, v) \in \mathcal{G}$. We interpret the grid $\mathcal{G}$ relative to the conditional image $\mathcal{I}$, such that each bin $\mathcal{B}_{u,v}$ covers a patch within $\mathcal{I}$.

The density estimator then directly predicts the point-wise non-parametric distribution over the possible future motion for scene element i at timestep τ given the known context $\mathcal{C}$ through $\mathbf{z}_\tau^i$:

$$\mathbf{D}_\omega^{\mathrm{d}}(\mathbf{z}_\tau^i) = \{p_\omega^{(u,v)}(\mathbf{z}_\tau^i)\}_{(u,v)\in\mathcal{G}} = p(t_\tau^i|\mathbf{z}_\tau^i) \approx p(t_\tau^i|\mathcal{C}) \tag{3}$$

We train the density estimator $\mathbf{D}_\omega^{\mathrm{d}}$ with a grid cross-entropy loss

$$\mathcal{L}_{\text{grid}} = -\sum_{(u,v)\in\mathcal{G}} \mathbf{1}\big[t_\tau^i \in \mathcal{B}_{u,v}\big] \log p_\omega^{(u,v)}(t_\tau^i|\mathbf{z}_\tau^i), \tag{4}$$

where $\mathbf{1}\big[t_\tau^i \in \mathcal{B}_{u,v}\big]$ is 1 if t_τ^i falls into bin $\mathcal{B}_{u,v}$ and 0 otherwise, resulting in a one-hot target distribution.

Note that the density estimator also receives only the point-wise latent $\mathbf{z}_\tau^i$ as input. Thus, $\mathbf{D}_\omega^{\mathrm{d}}$ is a decoder of the distribution encoded in the latent representation, learned through generative pre-training. Instead of estimating densities through Monte Carlo sampling, we decode them directly with a single ViT forward pass in less than 0.8 ms (s. Table 6) from the latent representation $\mathbf{z}_\tau^i$, enabling fast uncertainty estimation, uncertainty-aware planning, and interactive exploration of these motion distributions.

3.3 Interactive Exploration of Possibilities

At inference, the density decoder $\mathbf{D}_\omega^{\mathrm{d}}$ provides access to the distribution $p(t_\tau^i \mid \mathcal{C})$ in a single forward pass. This enables uncertainty quantification with GARFIELD by measuring the Shannon entropy of the predicted distribution

$$H(t_\tau^i) = -\sum_{(u,v)\in\mathcal{G}} p_\omega^{(u,v)}(t_\tau^i|\mathbf{z}_\tau^i) \log p_\omega^{(u,v)}(t_\tau^i|\mathbf{z}_\tau^i). \tag{5}$$

Thus, we can efficiently evaluate where future motion remains under-determined: high entropy indicates multiple plausible futures, while low entropy corresponds to collapsed regions of the possibility space. This allows users to identify and explore regions where the range of possibilities remains diverse, instead of wasting effort on collapsed low-variance regions of the kinematics distribution.

The density estimator enables interactive visualization and exploration of possible futures in real-time with no need for sampling multiple realizations from a generative model. Users can therefore explore possible futures and iteratively modify the constraints $\mathcal{C}$ in a feedback loop, enabling interactive planning by guiding and narrowing the set of feasible trajectories. Once the user is satisfied with the result, the latents $\mathbf{z}_\tau^i$ can be decoded *once* into a full, coherent trajectory realization with a generative model.

3.4 Sampling Coherent Realizations

For many downstream applications in motion planning, concrete trajectory realizations are required. Naively, GARFIELD could infer trajectories from the point-wise decoder by factorizing the distribution of scene kinematics as

$$p(T \mid \mathcal{C}) \approx \prod_i \prod_\tau p(t_\tau^i \mid \mathcal{C}). \tag{6}$$

However, if the space of possibilities is not fully collapsed and multimodal futures are possible, the point-wise decoder cannot resolve mutual dependencies between points within and between trajectories. Thus, a sample for timestep τ

might be drawn from one mode while the next timestep's sample $\tau + 1$ is drawn from another, resulting in temporally incoherent trajectories. The alternative of resolving these dependencies via auto-regressive updates of $\mathcal{C}$ would entail computational overhead and early greedy commitment to sampled points resulting in performance deterioration (Tab. D.3 in supplementary materials).

Instead, we train a full decoder $\mathbf{D}_\theta$ as a joint generative model conditioned on the set of all latents, which learns to sample

$$T \sim \mathbf{D}_\theta\Big(\big\{\mathbf{z}_\tau^i\big\}_{i,\tau}\Big) = \mathbf{D}_\theta(\mathbf{z}) = \mathbf{D}_\theta(\mathbf{E}_\varphi(\mathcal{C})) \approx p(T \mid \mathcal{C}) \tag{7}$$

using latents produced by the frozen encoder (Sec. 3.1). By sampling trajectories jointly, GARFIELD's full decoder correctly handles inter-dependencies.

By learning structured latents (Eq. (1)) with a point-wise generative decoder described in Sec. 3.1, GARFIELD enables access to density estimates through Eq. (3) and can sample coherent trajectories from $p(T \mid \mathcal{C})$ using Eq. (7).

4 Experiments

Our experiments evaluate GARFIELD's capabilities in goal-conditioned motion planning and its ability to predict distributions over future kinematics. We further show our method's usefulness as a foundation for downstream tasks.

4.1 General setup

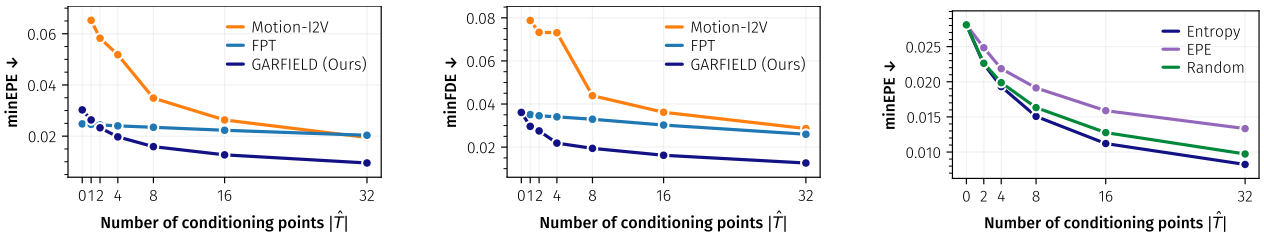
Data. For training and evaluation we annotate videos with off-the-shelf trackers [74] to obtain a pseudo-ground truth for supervision and metric calculation. We train on a dataset of 20M 224^2 px videos at 12 FPS that cover a wide range of subjects. Comparisons are performed on public OpenVid-1M [40] while some ablation studies are performed on a held-out validation set of the training data. We further evaluate on domain-specific data [12, 31, 45, 49, 61] with either human annotated trajectories or CoTracker3 [29] annotations following standard practice for application domains.

Implementation details. The encoder, full decoder, and density estimator of GARFIELD are implemented as transformer networks [17, 44]. We extract image features using DINOv2R-B [15, 41] and unfreeze DINO during training. We use 3D Axial RoPE [57], yet mask the starting position for the full decoder to avoid information leakage. We use g =20 for the density grid and append an out of bounds token. To ensure smoothness of $\mathbf{z}_\tau^i$ we apply a $\tanh$ function and add Gaussian noise with σ =1e-5 to $\mathbf{z}_\tau^i$ during training [52]. The encoder adopts static camera conditioning from FPT [8]. $\mathbf{D}_\psi^{\mathrm{p}}$ follows MAR [33] and is a 34M parameter MLP. More details are in Sec. A in the appendix.

Training details. The GARFIELD encoder $\mathbf{E}_\varphi$ is trained for 450k steps with the point-wise generative decoder and a linearly increasing goal sparsity $\widehat{T}/T$ from 0.5 to 0.01 in the first 50k steps. The density estimator $\mathbf{D}_\omega^{\mathrm{d}}$ and full decoder $\mathbf{D}_\theta$ are trained for 150k and 200k steps while keeping the encoder $\mathbf{E}_\varphi$ frozen. All training runs use a global batch size of 256, AdamW with a learning rate of 1e-4 (except for the density decoder with 1e-5), and bfloat16 precision. We generally model $i \in \{1 \dots 64\}$ tracks across $\tau \in \{1 \dots 32\}$ timesteps.

	Method	Text Free	Num. Goals	EPE ↓	PCK ↑ @10%	PCK ↑ @1%	FDE ↓	Latency↓ [s]
video	CogVideoX [71]	✗	n/a	0.027	0.952	0.466	0.032	178
	LTX [22]	✗	n/a	0.025	0.960	0.464	0.033	39
explicit motion	Motion-I2V [54]	✗	4	0.055	0.864	0.150	0.060	13.2
			16	0.033	0.957	0.263	0.037	
	Track2Act [10]	✓	n/a	0.041	0.931	0.091	0.046	4.10
	FPT [8]	✓	4	0.029	0.936	0.493	0.035	0.60
			16	0.026	0.943	0.523	0.032	
	GARFIELD (Ours)	✓	4	0.020	0.973	0.544	0.026	**0.40**
			16	**0.013**	**0.989**	**0.666**	**0.017**	

Table 1: Comparison on Motion Planning. GARFIELD achieves strong motion planning performance in open-set domains with reduced latency by modeling motion only for a subset of all scene elements and using spatio-temporally sparse goal conditioning.



(a): Average over time. **(b):** Final positional error. **(c):** Conditioning policy.

Figure 3: Impact of Goal Conditioning. GARFIELD is able to infer accurate motion in terms of **(a)** minEPE and **(b)** minFDE with limited conditioning. **(c)** Selecting conditioning based on entropy yields faster collapse than an oracle or random selection.

Metrics. For open-set evaluation, we measure motion accuracy using positional errors normalized to $[0, 1]$, where 1 corresponds to the image size. Our main metric is the *endpoint error* $\text{EPE} = \|\bar{t}^i_\tau - t^i_\tau\|_2$, averaged over all trajectories i and timesteps τ. We also report the *percentage of correct keypoints* $\text{PCK@}\alpha = \mathbb{E}[\text{EPE} < \alpha]$ with $\alpha \in \{10\%, 1\%\}$ relative to image size and the Final Distance Error (FDE), i.e., EPE at the final timestep. Calibration is evaluated using the Energy Score (ES), a proper scoring rule for probabilistic predictions. Domain-specific evaluations follow standard protocols from prior work. Appendix Sec. C provides more metric details.

4.2 Open-set Motion Modeling

Motion Planning. We compare the GARFIELD full decoder to open-set baselines for motion planning on the public OpenVid-1M [40] dataset. Models predict future motion from a start frame and sparse future constraints (text, goal images, or positions depending on the method). Because multiple futures may satisfy the constraints, we generate five samples per model and report the best. We evaluate both RGB video models and open-set trajectory models. For the former, we use TI2V models [22, 71] conditioned on the start frame and OpenVid prompt informing the model about temporal dynamics. Motion is extracted from generated videos using TapNext [74]. Motion-I2V [54] predicts optical flow from spatially sparse trajectories and interpolates temporally sparse signals. We generally run Motion-I2V with default settings, we compare to other configurations in Appendix Sec. E. Track2Act [10] generates trajectories on a dense grid conditioned on start and goal RGB frames. FPT [8] predicts the distributions of changes between a start and final timestep. To obtain full trajectories, we rerun FPT for multiple timesteps using only the spatio-temporally sparse conditioning points GARFIELD observed.

Tab. 1 shows GARFIELD achieves the best EPE, PCK, and FDE. Therefore, we are competitive with much larger and orders-of-magnitude slower video generation models. We also outperform Track2Act, which relies on a dense final RGB frame that is typically unavailable in open-set motion planning. Rerunning FPT for each timestep yields subpar results, highlighting the importance of modeling temporal dependencies. Motion-I2V receives the same spatio-temporally sparse points as GARFIELD and additional text prompts, yet performs worse and incurs higher latency, highlighting the advantage of modeling motion only for relevant scene elements and not having to temporally interpolate conditioning. Overall, GARFIELD provides state-of-the-art motion planning in open-set domains, which we attribute to explicit motion modeling of relevant scene elements and spatio-temporally sparse conditioning that provides a more precise control signal than text or RGB-based conditioning used in prior video [22, 71] and motion generation [10, 54] methods.

Goal Conditioning. Figure 3 shows the impact of adding more sub-goals to the conditioning set $\mathcal{C}$. GARFIELD is able to perform more accurate trajectory planning when using very sparse information, i.e. limited future positional information. GARFIELD, given only $|\hat{T}| = 4$, matches Motion-I2V using $|\hat{T}| = 16$ in EPE. Further, our approach is competitive with FPT [8] using only image conditioning and outperforms FPT consistently when using more than 2 goal positions.

Qualitative Evaluation. We present qualitative samples from GARFIELD with only $\hat{T}/T = 1\%$ known trajectory points on the left side of Figure 4. GARFIELD is able to predict complex trajectories for a range of objects and activities given minimal future knowledge. Fig. G.7 shows qualitative comparisons to baselines in the Supplementary Materials. The right side of Figure 4 shows that $\mathbf{z}^i_\tau$ can encode multiple possible futures given the same constraints $\mathcal{C}$. We show the zoomed-in densities obtained from the GARFIELD density estimator $\mathbf{D}^{\text{d}}_\omega$ as well as two samples from $\mathbf{D}_\theta$ for the same latent representation, highlighting GARFIELD's ability to capture the distribution of future outcomes.

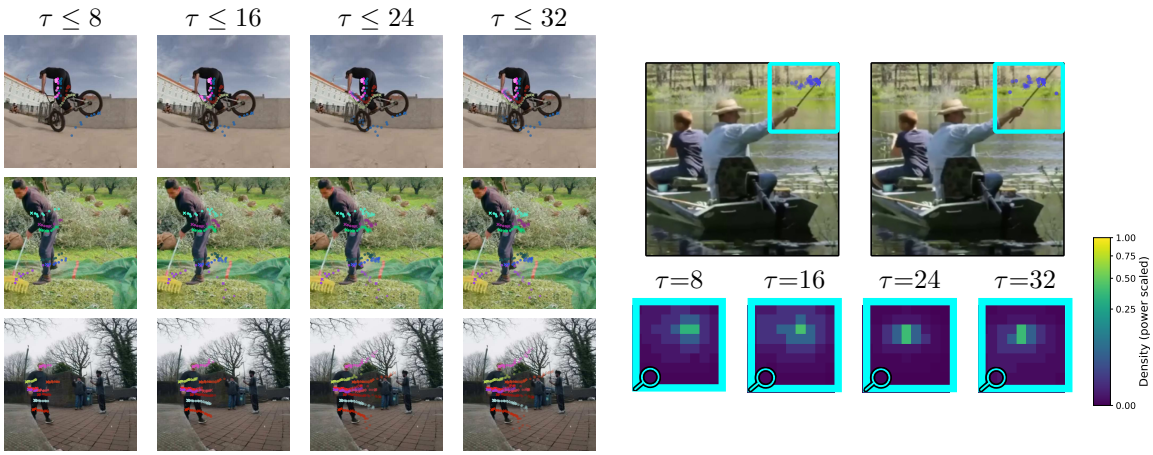


Figure 4: Qualitative open-set motion and uncertainty. (**left**) With sparse conditioning ($\widehat{T}/T = 1\%$), GARFIELD samples realistic motion. Dots denote predictions and crosses conditioning points. (**right**) Complex motion gives non-collapsed distributions.

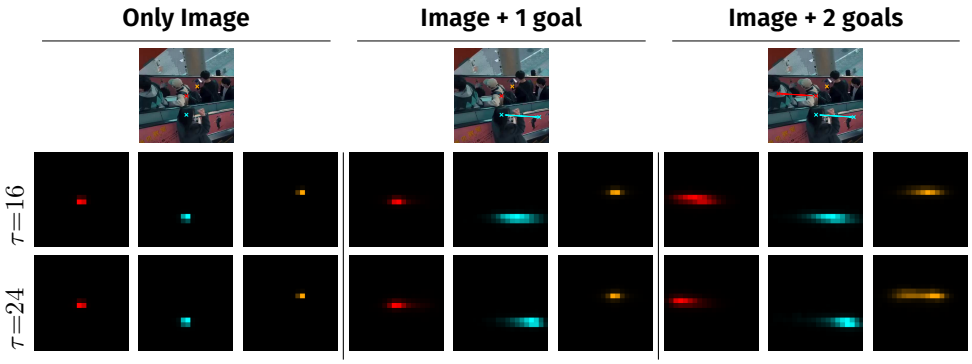


Figure 5: Distributions with minimal conditioning. Given $|\widehat{T}|$=0, GARFIELD assumes a static scene. With $|\widehat{T}|$=1 the model captures motion but produces inaccurate object relations. With $|\widehat{T}|$=2, GARFIELD remains uncertain only about exact speed.

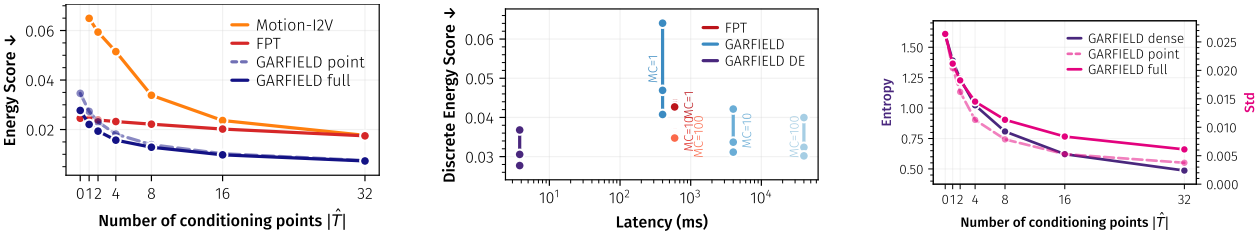
Figure 5 demonstrates qualitatively that the estimated densities reflect GARFIELD's improved motion understanding with more conditioning. While the model predicts a mostly static scene without future positions, and fails to account for object inter-dependencies with only one conditional position, it is able to infer complex motion distributions with as few as two conditioning points.

Distributions. We aim to verify our encoded latent captures meaningful uncertainty about scene dynamics. Figure 6a shows the energy score of our point-wise and full decoders. Compared to Motion-I2V [54] and FPT [8] our approach achieves superior energy score. Evaluating the density estimator $\mathbf{D}^{\mathrm{d}}_{\omega}$, we further compute a discretized energy score (Appendix Sec. C) and find that the density estimator achieves superior discrete energy scores while speeding up computation by two orders of magnitude. Additionally, Fig. 6c shows that providing more conditioning tightens the distributions. We provide further evaluation of calibration in Sec. E in the supplementary materials.

Automated interactive conditioning. We consider how we should perform interventions to collapse the distribution to the ground truth with minimal outside information (e.g. by prompting the user to provide further input). Using the

Method	BridgeData [61] ↑	RT1 [12]↑
Flow [69]	0.32	0.38
Track2Act [10]	**0.77**	0.75
GARFIELD (Ours)$_{\text{0-shot}}$	0.76	**0.77**

Table 2: Robotic Planning. Given final goals GARFIELD can infer robot arm motion with similar quality to prior art without finetuning. We extend [10]'s table.



(a): Energy Score. **(b):** Discretized energy scores. **(c):** Distribution tightening.

Figure 6: Evaluation of Distributions. GARFIELD achieves the best energy score. The density decoder $\mathbf{D}_\omega^{\mathrm{d}}$ attains superior discretized energy scores with orders-of-magnitude lower latency. Additional conditioning reduces entropy and variance.

GARFIELD density estimator we can quantify the model's uncertainty in each track position using entropy (Eq. (5)). We propose using entropy as an indicator for where to provide conditioning by finding the most uncertain trackpoint given the current constraints. Fig. 3c compares our entropy-based conditioning to an oracle-based variant that has access to the current EPE for each trackpoint and a random baseline. For automated evaluation, we insert ground-truth conditioning at either the most uncertain, the highest-error, or a random trackpoint. While using the error oracle performs even worse than random selection, we find that interactive entropy-based conditioning performs strongly. The entropy-based conditioning matches oracle-based conditioning with only half the conditional information. Note that no oracle is available in deployment while entropy-based conditioning selection does not rely on ground truth information. We additionally evaluate the robustness of entropy-based conditioning w.r.t. noisy user input in Appedix Sec. F.

4.3 Application Domains

Robotics. Planning is commonly used to evaluate motion models [10, 42, 56, 66]. We compare GARFIELD following the protocol from Track2Act [10] and sample trajectories for a robotic arm given start and goal states. While Track2Act uses dense RGB for start and end states, we use readily available positional information. We follow Track2Act and report *area under the curve* $\Delta = \frac{1}{A}\sum_{a=1}^{A} PCK@\frac{a}{S}$, where $A = 10$ is the highest PCK threshold in pixel units and $S = 256$ is the pixel resolution (more details: Appendix Sec. C). Tab. 2 highlights GARFIELD is competitive with Track2Act without training on robotics data. In comparison, Track2Act trains on splits from RT1 [12] and BridgeData [61]. This highlights fundamental understanding of motion and kinematics constraints learned by our model.

Pedestrian Trajectory Prediction. Pedestrian trajectory prediction important for autonomous driving and a long-standing benchmark. We compare zero-shot and finetuned performance of GARFIELD. As we require RGB input, we use only a subset of ETH/UCY. Further details are provided in Appendix Sec. E. Tab. 3 shows our zero-shot performance is competitive with established methods such as Social-LSTM [1] even though we *do not train for prediction*. With minimal finetuning (<15k iterations) GARFIELD achieves performance comparable to Social-GAN [21]. While we are outperformed by more recent domain-specific methods, our results underscore the utility of our representations.

Method	ETH [45]		HOTEL [45]		SDD [49]	
	ADE ↓	FDE ↓	ADE ↓	FDE ↓	ADE ↓	FDE ↓
Social-LSTM [1]	1.09	2.35	0.79	1.76	57.00	31.20
Social-GAN [21]	0.81	1.52	0.72	1.61	27.23	41.44
Trajectron++ [51]	0.39	0.83	0.12	0.21	8.98	19.02
IDM [35]	0.41	0.62	0.15	0.25	7.46	13.83
GARFIELD (Ours) 0-shot	1.09	1.79	0.41	0.71	36.77	49.86
GARFIELD (Ours) FT	0.84	1.44	0.29	0.56	19.92	34.98

Table 3: Pedestrian trajectory prediction. Errors are reported in pixels w.r.t. the original video resolution for Stanford Drone and meters for ETH/UCY. Baseline numbers are taken from IDM [35]. GARFIELD performs competitively.

Latent structure	EPE ↓	FDE ↓
Global	0.479	0.482
Entangled	0.509	0.510
Ours	0.011	0.015

Latent size	EPE ↓ $\frac{\widehat{T}}{T} = 1\%$	EPE (OOD) ↓ $\frac{\widehat{T}}{T} = 90\%$
16	0.012	0.009
64	0.012	0.008
256	0.012	0.019

Table 4: Embeddings. Using non-disentangled representations yields performance degradation. Smaller embeddings achieve superior performance in OOD settings. Evaluated on held-out validation set of training data.

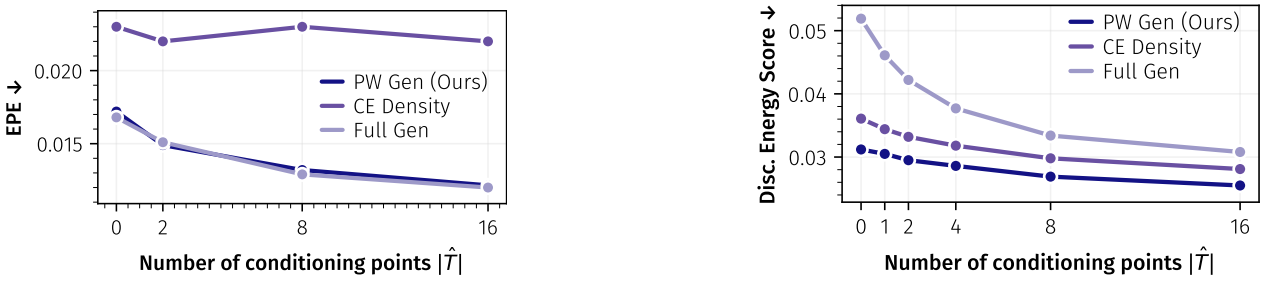
4.4 Ablations

Entangled and Global Representations. The left side of Tab. 4 shows training the point-wise decoder with a merged global token or without enforcing structure yields substantially degraded performance and does not allow decoding into accurate points. Intuitively, factorized $\mathbf{z}_\tau^i$ encode $p(t_\tau^i \mid \mathcal{C})$ and can be seamlessly decoded by the point-wise decoder and density estimator. In comparison, for non-localized embeddings the decoder has to identify relevant information by itself, resulting in a substantially harder task for $\mathbf{D}_\psi^{\text{p}}$ and $\mathbf{D}_\omega^{\text{d}}$. In comparison, our disentangled $\mathbf{z}_\tau^i$ enable high-quality track predictions and estimation of point-wise densities.

Latent Dimensionality. Tab. 4 further shows the performance on in-distribution samples ($\widehat{T}/T = 1\%$ conditioning) is almost independent of latent dimensionality. However, in OOD settings, e.g. when substantially more conditioning ($\widehat{T}/T = 90\%$) is given, smaller latents outperform larger variants, potentially due to over-adaptation. Unless noted otherwise, we use a latent dimensionality of 64 as a balance with strong performance on both in-domain and OOD tasks.

Pretraining with other Decoders. Fig. 7 compares pre-training the encoder $\mathbf{E}_\varphi$ with the point-wise decoder $\mathbf{D}_\psi^{\text{p}}$, full decoder $\mathbf{D}_\theta$, or density decoder $\mathbf{D}_\omega^{\text{d}}$. Fig. 7a shows motion planning performance is similar between directly training with the full decoder or first pretraining with a point-wise decoder. However, in Fig. 7b the density estimator fails to produce high-quality uncertainty estimates when the encoder is pre-trained with the full decoder. Also, pretraining the encoder with the density estimator and a grid Cross-Entropy objective from Eq. (4) leads to inferior performance of the full and density decoder. We conclude that (1) the density decoder's discretization limits the encoder's learning signal and (2) the density decoder $\mathbf{D}_\omega^{\text{d}}$ struggles to interpret entangled latents learned by pretraining the encoder with the full decoder $\mathbf{D}_\theta$ similar to Tab. 4. Thus, pretraining the encoder with our point-wise decoder $\mathbf{D}_\psi^{\text{p}}$ preserves both strong full trajectory sampling with a second stage decoder $\mathbf{D}_\theta$ and interpretability of $\mathbf{z}_\tau^i$ for the density decoder $\mathbf{D}_\omega^{\text{d}}$.

Full vs. Point-wise Decoder. Our point-wise decoder $\mathbf{D}_\psi^{\text{p}}$ is able to infer the positions of scene elements at a specific timestep given a latent $\mathbf{z}_\tau^i$. Full trajectories can be reconstructed from independent point-wise estimates. However, in case of multi-modal uncertainty, independent point-wise estimation breaks down as samples per timestep can be assigned to different modes. The full decoder $\mathbf{D}_\theta$ resolves these inter-dependencies. Tab. 5 verifies this assumption: in uncertain cases with limited $\mathcal{C}$ the full decoder produces higher quality estimates, while the point-wise model performs competitively when the distribution is collapsed.



(a): Full Decoder

(b): Density Estimator

Figure 7: Comparison of Training Protocol. We compare our pretraining with a point-wise generative decoder $\mathbf{D}_\psi^{\text{p}}$ (*PW Gen*) to training the encoder with the full decoder $\mathbf{D}_\theta$ (*Full Gen*) or the deterministic density estimator $\mathbf{D}_\omega^{\text{d}}$ (*CE Density*). The former achieves comparable track prediction performance but fails for density decoding, while the latter fails to produce a representation that can be used for motion planning. Evalauted on held-out validation set of training data.

Decoder	$\lvert\widehat{T}\rvert$	EPE ↓	FDE ↓
Point-wise $\mathbf{D}^{\mathrm{p}}_{\psi}$	2	0.035	0.060
	4	0.026	0.047
	16	0.014	0.024
Full sample $\mathbf{D}_{\theta}$	2	0.023	0.029
	4	0.019	0.024
	16	0.013	0.017

Table 5: Point-wise vs. Full Decoder. Given limited constraints the full decoder outperforms the point-wise model.

Component	Latency [ms]
Encoder $\mathbf{E}_{\varphi}$	3.0
Density $\mathbf{D}^{\mathrm{d}}_{\omega}$	0.8
Point-wise $\mathbf{D}^{\mathrm{p}}_{\psi}$	37.0
Full $\mathbf{D}_{\theta}$	330.4

Table 6: Latency. The density decoder predicts heatmaps faster than either generative decoder produces a sample.

Grid Resolution. Intuitively, higher g balance fine-granular density estimates with compute cost. Fig. 8 shows this trade-off empirically. As g influences the value range of the discrete energy score, we keep $g = 20$ fixed for all other evaluations and additionally report results with normalized grids ($g = 40$) for the ablation.

Alignment of Direct and MC Densities. The density decoder $\mathbf{D}^{\mathrm{d}}_{\omega}$, point-wise decoder $\mathbf{D}^{\mathrm{p}}_{\psi}$, and full decoder $\mathbf{D}_{\theta}$ decode the same latents. Thus, heatmaps from $\mathbf{D}^{\mathrm{d}}_{\omega}$ should align with MC density estimates from $\mathbf{D}_{\theta}$ and $\mathbf{D}^{\mathrm{p}}_{\psi}$. Figure 9 confirms this: both decoders achieve lower Jensen–Shannon divergence and Wasserstein distance to $\mathbf{D}^{\mathrm{d}}_{\omega}$ than 1-hot ground-truth heatmaps or baseline samples.

Component Latency. Tab. 6 shows runtimes for our individual components. The density estimator is able to decode a non-parametric distribution faster than either the point-wise or full decoder can produce a single example.

5 Conclusion

Reasoning about partially observable and uncertain environments requires modeling not only what will happen, but what *could happen*. We presented GARFIELD, a model that represents the space of possible scene kinematics through a structured probabilistic latent representation learned with a joint motion encoder and a point-wise diffusion decoder.

Conditioned on sparse goal positions, this representation captures the distribution of plausible future motions and progressively sharpens as additional constraints become available. From the same latent representation, we obtain both coherent trajectory samples and probability densities over future positions through an efficient deterministic density decoder. These densities provide direct access to uncertainty about where scene elements may appear, thus letting users identify underdetermined future regions and refine them through sparse conditioning. This enables interactive exploration and motion planning with minimal user intervention.

GARFIELD learns an open-set motion representation that performs competitively with large video models at substantially lower latency. The learned distributions collapse as additional information is given and transfer to downstream tasks such as robotic planning and pedestrian trajectory prediction. More broadly, we view GARFIELD as a step toward models that reason about the space of possible futures rather than single outcomes, enabling interpretable, controllable, and efficient probabilistic modeling of scene evolution.

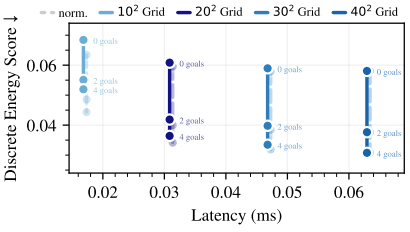


Figure 8: Grid size ablation. Grid size g trades quality for latency. Evaluated on held-out validation split of training data.

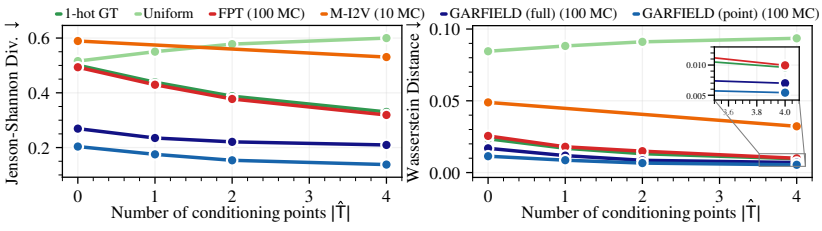


Figure 9: Alignment of MC samples and densities. (**left**) MC samples from $\mathbf{D}_{\theta}$ and $\mathbf{D}^{\mathrm{p}}_{\psi}$ best match $\mathbf{D}^{\mathrm{d}}_{\omega}$ in Jensen-Shannon divergence. (**right**) Their MC heatmaps also achieve the lowest Wasserstein Distance to $\mathbf{D}^{\mathrm{d}}_{\omega}$.

Acknowledgements

This project has been supported by 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 Stefan Baumann, Johannes Schusterbauer, Ming Gui, and Ulrich Prestel for helpful discussions and feedback, Nick Stracke, Frank Fundel, and Thomas Ressler-Antal for initial data work, Stefan Baumann for creating the LaTeX template, and Owen Vincent for continuous technical support.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. accessed 2025-11-18.

[2] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. accessed 2026-06-22.

[3] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2022. accessed 2024-09-17.

[4] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Thirty-eighth Conference on Neural Information Processing Systems*, 2024. accessed 2026-06-26.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. accessed 2026-03-08.

[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. accessed 2026-06-26.

[7] 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, Eloi Zablocki, Andrei Bursuc, Eduardo Valle, and Matthieu Cord. VaViM and VaVAM: Autonomous Driving through Video Generative Modeling, 2025. accessed 2025-07-10.

[8] 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. accessed 2026-06-22.

[9] Stefan Andreas Baumann, Jannik Wiese, Tommaso Martorella, Mahdi M Kalayeh, and Björn Ommer. Envisioning the future, one step at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6823–6836, 2026. accessed 2026-06-22.

[10] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, 2024. accessed 2026-06-26.

[11] 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. accessed 2026-06-26.

[12] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023. accessed 2026-06-26.

[13] 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. accessed 2026-06-22.

[14] Guangyi Chen, Junlong Li, Nuoxing Zhou, Liangliang Ren, and Jiwen Lu. Personalized trajectory prediction via distribution discrimination. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15560–15569, 2021. accessed 2025-07-21.

[15] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023.

[16] Patrick Dendorfer, Aljosa Osep, and Laura Leal-Taixé. Goal-gan: Multimodal trajectory prediction based on goal position estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. accessed 2026-06-22.

[17] 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*, 2020. accessed 2024-12-13.

[18] Markus Ebke. python-billiards, 2019. accessed 2026-06-23.

[19] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. accessed 2026-06-26.

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

[21] 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. accessed 2026-06-22.

[22] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion, 2024. accessed 2026-06-26.

[23] Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models, 2025. accessed 2026-06-26.

[24] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. accessed 2026-03-08.

[25] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics*, 39(6):1–14, 2020. accessed 2025-09-06.

[26] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision (ECCV)*, pages 289–305. Springer, 2024. accessed 2026-06-26.

[27] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981. accessed 2026-03-01.

[28] Ronny Hug, Wolfgang Hübner, and Michael Arens. Introducing probabilistic bézier curves for n-step sequence prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10162–10169, 2020. accessed 2025-09-06.

[29] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and Better Point Tracking by Pseudo-Labelling Real Videos, 2024. accessed 2025-09-07.

[30] Kihwan Kim, Dongryeol Lee, and Irfan Essa. Gaussian process regression flow for analysis of motion trajectories. In *2011 International Conference on Computer Vision*, pages 1164–1171, Barcelona, Spain, 2011. IEEE. accessed 2025-09-06.

[31] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, pages 655–664. Wiley Online Library, 2007. accessed 2026-06-26.

[32] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6150–6156. IEEE, 2019. accessed 2026-06-26.

[33] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, pages 56424–56445, 2024. accessed 2025-11-18.

[34] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2022. accessed 2026-06-26.

[35] Chen Liu, Shibo He, Haoyu Liu, and Jiming Chen. Intention-aware denoising diffusion model for trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2025. accessed 2026-06-26.

[36] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. accessed 2026-06-26.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. accessed 2026-06-26.

[38] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. VDT: General-purpose video diffusion transformers via mask modeling. In *The Twelfth International Conference on Learning Representations*, 2024. accessed 2026-06-26.

[39] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024. accessed 2026-06-26.

[40] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *The Thirteenth International Conference on Learning Representations*, 2025. accessed 2026-06-26.

[41] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[42] Karran Pandey, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, Niloy J. Mitra, and Paul Guerrero. Motion modes: What could happen next?, 2024. accessed 2026-06-26.

[43] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2013. accessed 2026-06-26.

[44] 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. accessed 2026-06-22.

[45] 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. accessed 2026-06-26.

[46] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2025. accessed 2026-06-26.

[47] Thomas Ressler-Antal, Frank Fundel, Malek Ben Alaya, Stefan Andreas Baumann, Felix Krause, Ming Gui, and Björn Ommer. Dismo: Disentangled motion representations for open-world motion transfer. In *Advances in Neural Information Processing Systems*, pages 158494–158534. Curran Associates, Inc., 2025. accessed 2026-06-26.

[48] Anthony Richardson and Felix Putze. Motion diffusion autoencoders: Enabling attribute manipulation in human motion demonstrated on karate techniques. In *Proceedings of the 27th International Conference on Multimodal Interaction*, pages 372–380, 2025. accessed 2026-06-26.

[49] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. accessed 2026-06-26.

[50] 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. accessed 2024-06-05.

[51] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data, 2021. accessed 2025-11-18.

[52] Johannes Schusterbauer, Jannik Wiese, Nick Stracke, Timy Phan, and Björn Ommer. Probabilistic precipitation nowcasting with rectified flow transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25742–25756, 2026. accessed 2026-06-26.

[53] Noam Shazeer. Glu variants improve transformer, 2020. accessed 2026-03-08.

[54] 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. accessed 2026-06-26.

[55] 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. accessed 2026-03-01.

[56] Nick Stracke, Kolja Bauer, Stefan Andreas Baumann, Miguel Angel Bautista, Josh Susskind, and Björn Ommer. Learning long-term motion embeddings for efficient kinematics generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 42581–42591, 2026. accessed 2026-06-22.

[57] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. accessed 2026-06-26.

[58] Didac Suris Coll-Vinent and Carl Vondrick. Representing spatial trajectories as distributions. In *Advances in Neural Information Processing Systems*, pages 13731–13744. Curran Associates, Inc., 2022. accessed 2026-06-26.

[59] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. accessed 2026-06-22.

[60] 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. accessed 2026-06-26.

[61] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *7th Annual Conference on Robot Learning*. PMLR, 2023. accessed 2026-06-26.

[62] 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, 2025. accessed 2025-11-18.

[63] Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8428–8437, 2025. accessed 2026-06-22.

[64] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pages 55–72. Springer, 2024. accessed 2026-06-22.

[65] Yuning Wang, Pu Zhang, Lei Bai, and Jianru Xue. Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1400–1409, 2023. accessed 2026-06-22.

[66] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2024. accessed 2025-07-02.

[67] Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. In *Advances in Neural Information Processing Systems*, pages 41051–41075. Curran Associates, Inc., 2024. accessed 2026-06-26.

[68] 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. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics. accessed 2026-06-26.

[69] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023. accessed 2026-06-26.

[70] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. accessed 2026-06-26.

[71] 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. accessed 2026-06-26.

[72] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. accessed 2026-06-26.

[73] Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27220–27230, 2025. accessed 2026-06-22.

[74] 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. accessed 2026-06-22.

[75] 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. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, 2025. accessed 2026-06-26.

Parameter	Encoder	Pointwise Decoder	Full Decoder	Density Estimator
Depth	24	3	28	12
Width	1024	1024	1152	768
Normalization	RMSNorm [72]	LayerNorm [5]	RMSNorm [72]	RMSNorm [72]
FFN expand factor	3	n/a	3	3
Activation	SwiGLU [53]	GELU [24]	SwiGLU [53]	SwiGLU [53]
Positional Encoding	3D RoPE [26, 57]	n/a	3D RoPE [26, 57]	2D RoPE [26, 57]
Static Scene Conditioning	AdaLN [44, 70]	n/a	n/a	n/a
Latent Conditioning	n/a	AdaLN [44, 70]	In-Context	In-Context
Image Encoder	DINOv2R-B [15, 41]	n/a	n/a	n/a
Batch size	256			
Optimizer	AdamW [37]			
Learning rate	$1e-4$	$1e-4$	$1e-4$	$1e-5$
LR schedule	Linear + Constant			
LR schedule steps	300 Linear warmup steps			
Betas	(0.9, 0.95)			
Conditioning sparsity	1% conditioning			
Sparsity schedule	Linear from 50% + Constant			Constant
Sparsity schedule steps	50k Linear warmup steps			n/a
Training steps	450k		200k	150k
Precision	$\mathrm{bfloat16}$			
Trainable Parameters	510M	34M	522M	97M
GPUs	16 Nvidia H200			
Training Time	4 days		1 day	1 day
Dataset	Open-Set Videos			
Number of clips	20M			
Tracker	TapNext [74]			
Tracker Position Seeding	1024 random positions			
Position Scale	[0,1]			
Image size	224×224			

Table A.1: Hyperparameters. We summarize GARFIELD and training configurations for all components of our method below.

A Implementation Details

Hyperparameters. The hyperparameters for all GARFIELD models used in evaluations and for qualitative results are summarized in Tab. A.1. We train the encoder model with the pointwise decoder for a total of 450k steps using the AdamW [37] optimizer with a linear learning rate warm-up of 300 steps and a final constant learning rate of 0.0001. We use a global batch size of 256 for all training runs. During encoder training, we start with $\widehat{T}/T = 50\%$ known positions and linearly decrease the amount of future knowledge to $\widehat{T}/T = 1\%$ over the first 50k training iterations. We find that this curriculum w.r.t. conditional sparsity results in better adherence to conditionings, while starting from more than $\widehat{T}/T = 50\%$ known positions results in training instabilities. The full decoder and density estimator are trained with a frozen encoder model using the same batch size, optimizer, and learning rate. We keep conditioning sparsity constant for decoder training and train these models for 150k (density decoder) and 200k (full decoder) iterations. All training is performed with $\mathrm{bfloat16}$ precision and parallelized across 16 Nvidia H200 GPUs for efficiency.

Starting from 20M video clips with high variety, we apply TapNext [74] to obtain our training data. The tracker is seeded with random space-time positions. During training, we apply a visibility filter to ensure all points our model observes are visible in the first frame I. We further utilize the tracker's predicted certainty to ensure high-quality tracks for training. We normalize positions w.r.t. the image size, such that visible tracks are in $[0, 1]$ range. Start frames are resized to 224×224 resolution before obtaining image features from the pre-trained foundational image feature extractor [55], which is finetuned during encoder training.

For tokenization, known conditional track positions $\widehat{T}$ are encoded as Fourier features, while unknown track positions are replaced with a learnable query token. For known positions, we concatenate image features from the corresponding position. Further, the encoder performs cross-attention to all image features. The resulting token sequence is passed to the encoder's transformer which is 24 layers deep and has a width of 1024. The encoder produces latents for all known and unknown positions. While we supervise all positions, we neglect conditional positions for evaluation. Encoder transformer tokens are finally downprojected to a fixed latent dimensionality of 64, and a $\tanh$ function is applied to obtain the final latent representation. We further add a small noise perturbation during training to ensure a locally

smooth latent space [52]. We use standard Llama-style transformers with RMSNorm [72], SwiGLU [53] activations, and RoPE [57] positional embeddings. The point-wise decoder is a three layer MLP with a width of 1024, using GELU [24] activations, and LayerNorm [5]. The point-wise decoder receives a noisy version of the sampled point as input and is conditioned on the diffusion timestep and latent via adaptive normalization [44, 70]. A final layer downprojects the hidden dimensionality to the point dimensionality. The full decoder is implemented as a DiT [44] with a similar architecture to the encoder. It is conditioned on the diffusion timestep via adaptive normalization, and conditioned on the full set of encoder latents by concatenation of noisy inputs and corresponding latent representations. The transformer tokens are finally downprojected to the data dimensionality. The density estimator is implemented as an S-sized ViT [17] with a depth of 12 and width of 768. It is conditioned on a single latent by prepending the latent to the sequence of grid tokens. These grid tokens are initialized as Fourier embedded grid positions. We further add a learnable query token for out of bounds positions. The density decoder transformer tokens are ultimately downprojected to a dimensionality of one for each grid cell and the out of bounds position, providing cross-entropy logits.

We pre-train the encoder and pointwise decoder with a point-wise flow matching objective (Eq. (B.5)). The full decoder is trained with a flow matching objective over all track positions. The density estimator is trained with a grid Cross Entropy loss (Eq. (4)). We do not enforce any attention masks in the encoder or decoder models and thus provide the model with the ability to leverage information about known parts of the future and track interdependencies for other track positions.

Conditioning. We condition decoders on encoder latents in different ways. For the full decoder $\mathbf{D}_\theta$ each trackpoint t^i_τ corresponds to exactly one specific processed token in the DiT and latent component $\mathbf{z}^i_\tau$. Hence, we enforce this structure by using *in-context conditioning*, concatenating noisy DiT inputs with corresponding latents before passing tokens to DiT layers.

For the pointwise decoder MLP $\mathbf{D}^{\text{p}}_\psi$, the conditioning signal is much stronger and easier to interpret, because it only receives a single latent $\mathbf{z}^i_\tau$ to denoise a single trackpoint t^i_τ. Thus, we resort to realizing the conditioning mechanism with *adaptive normalization* [44, 70]. We linearly project the latent with a learned matrix $Z^i_\tau = \text{proj}_Z(\mathbf{z}^i_\tau)$ and add projected latents to embeddings of the diffusion timestep κ. The combined conditioning $c = \text{proj}_c(\{Z^i_\tau, \kappa\})$ is then used to compute scale $\gamma(c)$ and shift $\beta(c)$ with learned linear projections. We then modify the standard normalization and compute

$$\text{AdaNorm}(x, \gamma(c), \beta(c)) = \gamma(c)\text{Norm}(x) + \beta(c). \tag{A.1}$$

We apply adaptive normalization before each MLP layer, conditioning each non-linear projection on the latent.

Similarly to the pointwise model, the density estimator $\mathbf{D}^{\text{d}}_\omega$ processes only a single latent $\mathbf{z}^i_\tau$ corresponding to a single trackpoint t^i_τ. Yet, similar to the full decoder, the density estimator processes a sequence of tokens enabling stronger conditioning. However, the density estimator is not a generative diffusion model, but a deterministic network. We opt for a variant of *in-context conditioning* where the latent $\mathbf{z}^i_\tau$ is projected to the model dimensionality $Z^i_\tau = \text{proj}_{dens}(\mathbf{z}^i_\tau)$ and prepended to the sequence of tokens corresponding to grid cells. Thus, the density estimator can utilize latent information via self-attention without the additional complexity introduced by cross-attention or adaptive normalizations.

Positional Embedding for Motion Modeling. We use 3D Axial RoPE [26, 57] for positional embedding in our Transformer models. However, we repurpose the spatial components of 3D RoPE to mitigate information leakage and foster trajectory coherence. Thus, for our encoder model, we use ground truth timestep information, anchoring each grid latent to a specific timepoint, yet we use only the ground truth position for the first timestep $\tau = 0$ for spatial positioning. We assume that the position at $\tau = 0$ is available through observation, while future positions are generally unknown. Thus, the input to RoPE for each token is a position pos given by

$$\text{pos} = (x_{\tau=0}, y_{\tau=0}, \tau), \tag{A.2}$$

which informs the token about its spatial origin and temporal position. Note that each point t^i_τ on the trajectory t^i for scene element i shares the same 2D start position $(x_{\tau=0}, y_{\tau=0})$, thus enabling the model to identify which points lie on the same trajectory.

Image tokens use the same coordinate system as spatial trajectory positions and are thus positionally embedded with the same positional embedding, setting $\tau = 0$. This eases the cross-attention of trajectory tokens to the corresponding image regions by aligning their rotation.

For the full decoder, we further prevent information leakage by choosing random positions $\tilde{x}_{\tau=0}, \tilde{y}_{\tau=0} \sim [0, 1]$ while

still using true temporal information τ. Thus, we retain the ability to infer which points lie on the same trajectory through shared spatial positions, yet true start positions have to be obtained from the latents, preventing any information leakage. The density estimator uses standard 2D Axial RoPE to embed positions of grid cells, while the point-wise decoder model does not require any positional embedding.

Static Camera Conditioning. We reuse the static camera detection from FPT [8]. Thus, we consider the camera to be static if less than a fraction α of all movements in the scene exhibit motion magnitudes greater than m,

$$\begin{aligned} \text{static}_{\alpha,m} =& \frac{1}{(H-1)N} \sum_{i=0}^{N} \sum_{\tau=0}^{H-1} \mathbf{1}_{\|t^i_{\tau+1} - t^i_\tau\|_2 > m} < \alpha \\ \text{where, } \mathbf{1}_c =& \begin{cases} 1, & \text{if } c \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{A.3}$$

where N is the number of trajectories and H is the time horizon. We manually tune $\alpha = 0.45$ and $m = 0.0002$ on our training dataset, maximizing the $F1$ score of correct detections on 100 manually labeled training clips with TapNext annotations. We then embed $c = \text{emb}(\text{static})$ and apply Adaptive Normalization [44, 70] defined in Eq. (A.1) in the encoder, where scale $\gamma(c)$ and shift $\beta(c)$ are determined by projections of the static camera embedding c, thus $(\gamma(c), \beta(c)) = \text{proj}(\text{c})$. This enables training on mixed data with both moving and static cameras, while fixing the camera for interpretable inference results. Therefore, the latent embedding encodes the information as to whether the scene has a static or moving camera.

Robotics planning data processing. To match Track2Act's dense motion prediction, we select the 256 tracks with the highest motion magnitude per video sample to evaluate our method and dismiss the rest as static background. Because our model is trained with $i \in \{1 \dots 64\}$ tracks (s. Sec. 4.1), we split the 256 tracks into 4 disjoint subsets of 64 tracks which serve as the GT for our evaluation.

B General Flow Matching

Generative models aim to produce samples from an unknown data distribution. We build upon the framework of flow matching [2, 3, 34, 36], which demonstrated strong mode coverage and high sample quality in complex domains such as image and video generation [19, 39, 46, 50].

Flow matching generates samples from a data distribution by transferring samples from a simple prior distribution $\mathbf{x}_0 \sim p_0$ to samples from the data distribution $\mathbf{x}_1 \sim p_1$. This transformation is defined through a continuous time-dependent process $\mathbf{x}_\kappa = \alpha_\kappa \mathbf{x}_1 + \sigma_\kappa \mathbf{x}_0$, where α_κ and σ_κ are functions that control the interpolation process over diffusion time $\kappa \in [0, 1]$. Training a model to predict the *velocity field*

$$\mathbf{v}(\mathbf{x}_\kappa, \kappa) = \frac{\mathrm{d}x}{\mathrm{d}\kappa} = \dot{\alpha}_\kappa \mathbf{x}_1 + \dot{\sigma}_\kappa \mathbf{x}_0 \tag{B.4}$$

allows sampling from the data distribution by numerically integrating this process over diffusion time.

We train $\mathbf{v}_\theta(\mathbf{x}_i, i)$ using the standard flow matching loss

$$\mathcal{L}_\mathbf{v}(\theta) = \int \mathbb{E}[\|\mathbf{v}_\theta(\mathbf{x}_\kappa, \kappa) - (\dot{\alpha}_\kappa \mathbf{x}_1 + \dot{\sigma}_\kappa \mathbf{x}_0)\|^2] \, \mathrm{d}\kappa. \tag{B.5}$$

We use the Rectified Flow interpolant [36] defined by $\alpha_\kappa = \kappa$ and $\sigma_\kappa = (1 - \kappa)$.

The density of the data distribution p_1 can be estimated via Monte Carlo (MC) sampling by generating N_{MC} samples from the model starting from different prior samples. In theory, this estimate converges to the true density as $N_{MC} \to \infty$. However, in practice, accurate density estimation remains prohibitively expensive for large models.

C Metric Details

End-Point Error. We measure the end-point error (EPE) as the average distance from predicted to ground truth trajectories. Consider a set of trajectories $T = \{t_\tau^i\}$ for N scene elements i and H timesteps τ. Given a prediction $\tilde{T} = \{\tilde{t}_\tau^i\}$ the endpoint error for this example is computed as

$$\mathrm{EPE}(T, \tilde{T}) = \frac{1}{H \cdot N} \sum_{i=0}^{N} \sum_{\tau=0}^{H-1} d_\tau^i, \tag{C.6}$$

where

$$d_\tau^i = \|t_\tau^i - \tilde{t}_\tau^i\|_2^2. \tag{C.7}$$

For a fair comparison, we evaluate EPE only for predicted trajectory points $\bar{T} = T \setminus \hat{T}$ and ignore conditional positions $\hat{T}$.

As multiple futures may align with the same constraints, we compute an ensemble of predictions $\tilde{E} = \{\tilde{T}_n\}$ for $n \in [1, N_{MC}]$ and take the minimum value

$$\mathrm{minEPE}(T, E) = \min_{\tilde{T} \in E} \mathrm{EPE}(T, \tilde{T}). \tag{C.8}$$

Thus, each method has the opportunity to make a range of possible predictions. Note that minEPE is equivalent to the commonly used minADE metric. Unless noted otherwise, EPE refers to minEPE averaged over multiple scenes, simplifying notation for the sake of readability. For pedestrian trajectory prediction evaluations on ETH/UCY in Tab. 3 we follow standard practice and average minADE over observed agents instead of scenes.

Final Distance Error. Following standard practice, we additionally report the Final Distance Error (FDE), the L2-distance between predicted and ground truth final positions. Therefore, for trajectories $T = \{t_\tau^i\}$ for N scene elements i and H timesteps τ the FDE is given by the average error of predicted trajectories $\tilde{T} = \{\tilde{t}_\tau^i\}$ at the last timestep $H - 1$:

$$\mathrm{FDE}(T, \tilde{T}) = \frac{1}{N} \sum_{i}^{N} \|t_{H-1}^i - \tilde{t}_{H-1}^i\|_2^1. \tag{C.9}$$

Similar to EPE, we report FDE only for predicted trajectory points $\bar{T} = T \setminus \hat{T}$ and use the minimum FDE over an ensemble of predictions averaged over multiple scenes. For ETH/UCY we again average over agents instead of scenes, following common practice.

Percentage of Correct Keypoints. We report the Percentage of Correct Keypoints (PCK) as an accuracy metric for trajectory prediction given pre-defined thresholds. Given trajectories $T = \{t_\tau^i\}$ for N scene elements i and H timesteps τ as well as predictions $\tilde{T} = \{\tilde{t}_\tau^i\}$, we first compute the per-trackpoint distances $D = \{d_\tau^i\}$ (s. Eq. (C.7)). Then, for threshold α the $\mathrm{PCK@}\alpha$ is given by

$$\begin{aligned} \mathrm{PCK@}\alpha(D) = \mathrm{PCK@}\alpha(\{d_\tau^i\}) &= \frac{1}{H \cdot N} \sum_{\tau=0}^{H-1} \sum_{i=0}^{N} \mathbf{1}_{d_\tau^i < \alpha} \\ \text{where, } \mathbf{1}_c &= \begin{cases} 1, & \text{if } c \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{C.10}$$

Therefore, $\mathrm{PCK@}\alpha$ measures the fraction of predictions that land with-in a distance α of the round truth.

We define α relative to the image size and report $\mathrm{PCK@}10\%$ and $\mathrm{PCK@}1\%$ only for the predicted points, ignoring the points provided as conditional information. We again average PCK values over multiple scenes and take the best prediction from an ensemble of N_{MC} samples by taking the maximum.

Area Under the Curve. For our robotic planning evaluation we follow Track2Act [10] and report the *area under the curve* over PCK thresholds. Effectively, we estimate the integral over PCK (s. Eq. (C.10)) with a sum over ten α

thresholds:

$$\Delta = \frac{1}{A}\sum_{a=1}^{A} \mathrm{PCK@}\frac{a}{S}(D) \approx \int_0^A \mathrm{PCK@}\frac{a}{S}(D)\,\mathrm{d}a, \tag{C.11}$$

where D are L2-distances between track points, $A = 10$ is the highest threshold in pixel units and $S = 256$ is the pixel resolution such that $0.39\,\% \leq \alpha \leq 3,91\,\%$.

Energy Score. [20] (ES) is a proper scoring rule for evaluating probabilistic forecasts of multivariate variables. Given a predictive distribution P over $X \in \mathbb{R}^d$ and an observed value y the energy score is computed as

$$\mathrm{ES}(P, y) = \mathbb{E}_{X\sim P}\big[\|X - y\|\big] - \frac{1}{2}\mathbb{E}_{X,X'\sim P}\big[\|X - X'\|\big]. \tag{C.12}$$

The first term measures the expected distance of samples from the distribution to the observation, while the second term rewards diversity in the predictive distribution. Therefore, ES balances closeness to the observation with internal diversity. Lower values indicate more calibrated predictions.

If the distribution P is represented by an ensemble of N_{MC} individual samples $\{x_1, \ldots, x_{N_{MC}}\}$ the expectations can be approximated empirically, yielding the *ensemble energy score*

$$\mathrm{ES}\Big(y, \{x_n\}_{n=1}^{N_{MC}}\Big) = \frac{1}{N_{MC}}\sum_{n=1}^{N_{MC}} \|x_n - y\| - \frac{1}{2{N_{MC}}^2}\sum_{n=1}^{N_{MC}}\sum_{m=1}^{N_{MC}} \|x_n - x_m\|. \tag{C.13}$$

Note that, if the ensemble is overly dispersed, the second term reduces the score less than the first term increases, penalizing underconfident predictions.

Discrete Energy Score. To evaluate calibration of a discretized probability distribution over a grid, we interpret each grid cell as a position by taking the grid cell center normalized to the full image size and ignore out-of-bounds positions. Therefore, the discretized probability distribution is interpreted as a distribution over $g \times g$ positions $s_{u,v} \in [0, 1]^2$ with grid indices $u, v \in \{1, \ldots, g\}$. The probability $P(X = s_{u,v})$ is given by the discrete probability $p_{u,v}$ assigned to the grid cell (u, v).

With this, the multivariate energy score for this distribution can be computed exactly (i.e. without the need for empirical ensemble approximation) as

$$\begin{aligned}\mathrm{ES}(P, y) = &\sum_{u=1}^{g}\sum_{v=1}^{g} p_{u,v}\,\|s_{u,v} - y\|_2 \\ &- \frac{1}{2}\sum_{u=1}^{g}\sum_{v=1}^{g}\sum_{u'=1}^{g}\sum_{v'=1}^{g} p_{u,v}p_{u',v'}\,\|s_{u,v} - s_{u',v'}\|_2.\end{aligned} \tag{C.14}$$

Note that discretization introduces different value ranges compared to the continuous energy score. However, discrete energy scores for ensemble-based methods can be computed by counting the relative frequencies of samples landing in grid cell regions and normalizing the resulting probabilities so that $\sum_{u=1}^{g}\sum_{v=1}^{g} p_{u,v} = 1$. Thus, discretization enables comparison of ensemble-based methods and methods that directly produce non-parametric probability heatmaps.

Latency. Latency measurements are performed on a single Nvidia H200 GPU with a batch size of one and reported only for computing a single realization, not the full ensemble. We warm-up model runs before computing latency measurements and use compilation where possible. For fairness, the time for embedding the start frame is also included in the latency measurement. We use the pipelines in diffusers [60, 68] for LTX [22] and CogVideoX [71] and the official repositories for Motion-I2V [54], Track2Act [10], and FPT [8]. Note that for FPT we use two auto-regressive loops, resulting in increased latency: auto-regressive updates of the conditioning pokes to achieve consistent predictions, and independent sampling of each timestep to obtain full tracks instead of single-step estimates.

Method	Conditioning			Prediction		
	Independent of appearance	Localized motion	Allows sub-goals	Spatially sparse	Temporally dense	Sampling-free uncertainty
Track2Act [10]	✗	✗	✗	✓	✓	✗
Motion-I2V [54]/ Motion Modes [42]	✓	✓	✓	✗	✓	✗
FPT [8]	✓	✓	✗	✓	✗	✓
What happens next [11]	✓	✗	✗	✗	✓	✗
GARFIELD (Ours)	✓	✓	✓	✓	✓	✓

Table C.2: Conceptual comparison. Unlike related approaches, we aim to model the distribution of motion densely over time with spatially sparse trajectories while also allowing single-NFE uncertainty estimation without expensive diffusion sampling.

D Conceptual Differences to Prior Motion Models

For clarification, we elaborate on conceptual differences between GARFIELD and previous open-set probabilistic motion models. We provide an argumentation, explaining the advantages of our formulation, for each difference. We separate differences in conditioning the predicted motion (i.e. inputs) from differences in what is predicted (i.e. outputs). Tab. C.2 summarizes our comparison.

D.1 Conditioning

Future Appearance Independence. Prior methods such as Track2Act [10] condition on the appearance of the expected result. They require not only a start frame from the present but also a goal frame from the end in full RGB detail. We argue (i) while appearance differences include all information necessary for motion planning, the information about goals is not directly accessible to the motion model, (ii) goals in the form of images contain spurious additional information that should not influence motion planning (e.g. lighting), (iii) information about the final appearance of a system is often hard to obtain, while positions can be more easily provided by users, and (iv) while changes in positions can be inferred from start and goal images by estimating semantic correspondances, inferring RGB details from a start frame and changes in positions is non-trivial, highlighting the increased generality of GARFIELD over appearance-based conditioning. Thus, we opt for appearance-free conditioning based only on positional information.

Localized Conditioning. Textual descriptions used by *What happens next?* [11] and goal images [10] can inform the model about the future. Yet, text is inherently coarse and ambiguous whereas full images are overly detailed. Thus, for both input modalities models require additional capacity to extract localized information about which scene element should move where. In comparison, we directly provide localized goals associated with specific scene elements, focussing modeling capacity on actual trajectory planning instead of goal understanding.

Sub-Goal Conditioning. In addition, prior approaches such as Track2Act [10] and FPT [8] only condition the final goal positions. GARFIELD additionally enables specification of sub-goals that should be passed on the way to the final goal. By enabling spatially and temporally sparse, yet arbitrarily distributed conditioning on future positions, our conditioning is maximally flexible. This enables users to provide exactly the information and assumptions about the future they have available and feel confident about, if any.

D.2 Prediction

Spatially Sparse Motion. Most prior open-set motion models such as Track2Act [10], *What happens next?* [11], and Motion-I2V [42, 54] predict the future evolution of the scene densely. They predict either optical flow [42, 54] or point trajectories with points sampled on a dense grid [10, 11]. Inspired by canonical baselines in closed-domain trajectory prediction [1, 21, 51] and the more recent open-set FPT [8] we argue (i) modeling motion for a spatially sparse subset of relevant points is sufficient to obtain meaningful estimates of the future. Additionally, (ii) modeling motion for spatially sparse points is computationally cheaper, as we focus on what is important instead of wasting effort to model static scene elements.

Method	Goals	EPE ↓	Latency ↓
AR ($\mathbf{D}^{\mathrm{p}}_{\psi}$)	4	0.076	71.4
	16	0.040	
GARFIELD (Ours) $\mathbf{D}_{\theta}$	4	0.014	**0.40**
	16	**0.012**	

Table D.3: AR vs. joint. Joint prediction achieves better trajectory quality and lower latency than AR rollouts. Evaluated on held-out validation set of training data.

We empirically find that in open-set scenes a large portion of trajectories is static and visualize a motion histogram in Fig. D.1. Further, sampling motion on a grid similar to the one used by *What happens next?* [11] incurs a $10\times$ latency increase with our model. Therefore, focusing on salient, moving, off-grid scene elements allows substantially faster inference.

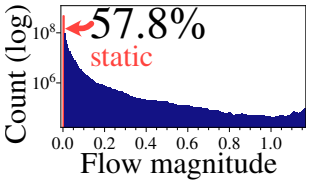


Figure D.1: Motion Magnitude Histogram. Most open-set flow magnitudes are small or static.

Temporally Dense Motion. The FPT model [8] provides spatially sparse predictions. However, they only model motion between two discrete points in time. We postulate (i) downstream tasks such as robotic planning benefit from a sequence of small changes rather than big changes between relatively far apart timesteps. Furthermore, (ii) single-step models such as FPT have to account for all interdependencies and interactions internally, limiting their performance in complex multi-object interaction scenarios (see Tab. F.11). GARFIELD predicts temporally dense trajectory estimates for a spatially sparse set of relevant objects.

Sampling-free Uncertainty. Probabilistic motion models based on generative diffusion [10, 11, 42, 54] provide implicit access to the underlying distribution of future motion through repeated sampling. However, estimating Likelihoods and probabilities of future events as well as quantifying the uncertainty about what might happen in a scene, is costly as it requires Monte-Carlo approximation. In contrast we propose learning a structured embedding of probability distributions over the future, that can be decoded into probability estimates in less than a millisecond, enabling orders of magnitude faster density estimation. Further, in our experiments direct density estimation consistently outperforms Monte-Carlo sampling.

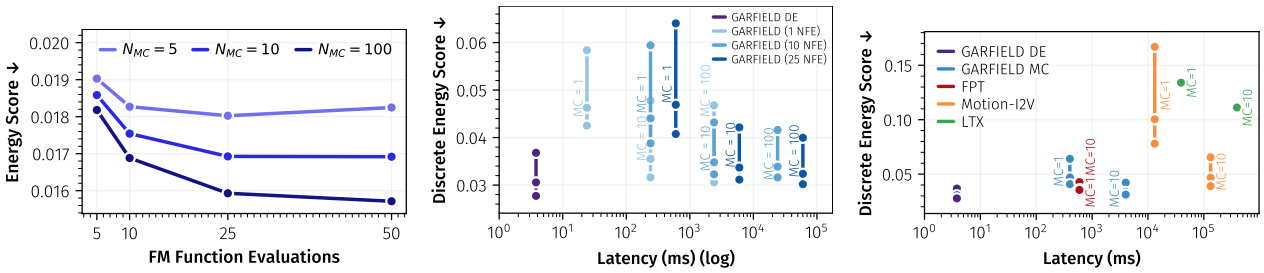
Joint Modeling. FPT [8], which is conceptually closest to our work predicts only a marginal distribution per scene element per timestep and maintains spatial coherence via autoregression (AR). Extending FPT's AR to multi-step horizons (similar to MYRIAD. [9]) requires early commitment which factorizes spatio-temporal inter-dependencies into marginals. By using a bidirectional encoder, our latents model the joint distribution of possible kinematics where the uncertainty of each point is informed by the whole scene without greedy commitment. From these latents, we sample full trajectories and estimate densities over future locations. Empirically, Tab. D.3 shows, that our GARFIELD joint full decoder improves sample quality and efficiency over a diffusion-based FPT-style AR variant extended to multiple steps, and sub-goal conditioning.

E Extended Evaluations

Test-time Scalability. Fig. E.2a demonstrates how more function evaluations and larger Monte Carlo ensembles impact the energy score. For large ensembles, there is a consistent benefit of investing more compute, in the form of NFEs, into each individual member. For small ensembles, however, energy score diverges slightly for high NFE. We attribute this to error accumulation in the sampling process. The same trend applies to discretized energy scores visualized in Fig. E.2b. The point-wise decoder requires three orders of magnitude more compute to produce ensembles that come close to the performance of the discrete model. In general, sampling more and higher quality realizations results in better motion planning results, providing the users with the ability to balance this trade-off according to their current needs.

Extended Comparison of Discretized Energy Scores. We extend Fig. 6b with more expensive baselines in Fig. E.2c to further emphasize the benefit of our proposed density estimator $\mathbf{D}^{\mathrm{d}}_{\omega}$. Our density decoder produces better predictions than large video generation models with five orders of magnitude reduced latency.

Additional Calibration Metrics. To further validate the calibration of densities decoded by $\mathbf{D}^{\mathrm{d}}_{\omega}$, we additionally compute Negative log-Likelihoods of GT trajectories (NLL), reliability curves (with $|\hat{T}| = 4$), and coverage vs. nominal

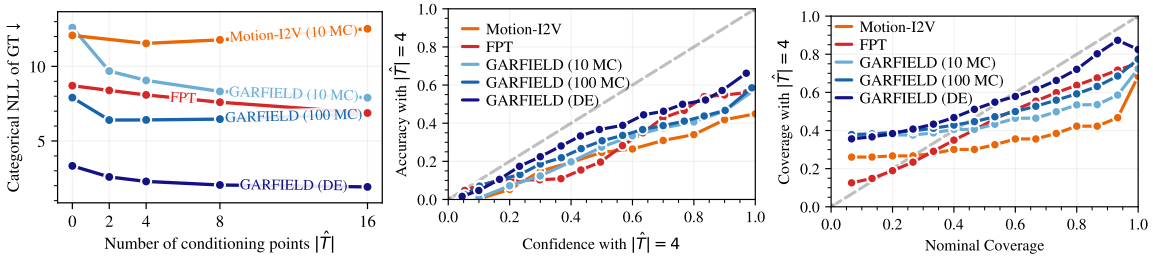


(a): Energy Score vs. NFE and MC samples for point-wise $\mathbf{D}_\psi^{\mathrm{p}}$.

(b): Discrete Energy Score vs. NFE and MC samples.

(c): Discrete Energy Score vs. Latency for more baselines.

Figure E.2: Calibration trade-offs. We visualize trade-offs between well-calibrated density estimation, indicated by low (discrete) energy score, and the compute latency to estimate these densities. While more NFEs and larger ensembles improve motion prediction, they also induce additional latency. In **(b)** and **(c)**, we also show how adding conditionals ($|\hat{T}| \in \{0, 1, 2\}$) further lowers the discrete energy score for the same model and ensemble size.



(a): NLL of GT trajectory.

(b): Reliability Diagram.

(c): Coverage vs. Nominal Coverage.

Figure E.3: Calibration of Densities. Densities from the density decoder $\mathbf{D}_\omega^{\mathrm{d}}$ consistently outperform Monte-Carlo sampling and outperforms strong baselines in terms of negative log-Likelihood of the ground truth trajectory. Evaluated on held-out validation set of training data.

coverage curves (with $|\hat{T}| = 4$). Note that while (discrete) energy scores and NLL are proper scoring rules, reliability and coverage can improve under variance inflation and should therefore be considered only as supplementary to the other metrics. Fig. E.3a shows that our density decoder $\mathbf{D}_\omega^{\mathrm{d}}$ outperforms not only baseline approaches but also MC estimates from the full decoder $\mathbf{D}_\theta$ and the point-wise decoder $\mathbf{D}_\psi^{\mathrm{p}}$ in NLL. This highlights the advantage of direct density decoding over estimating densities via Monte-Carlo sampling. Additionally, Fig. E.3b and Fig. E.3c show that our density decoder $\mathbf{D}_\omega^{\mathrm{d}}$ performs competitively with strong baselines for reliability and coverage while consistently outperforming MC sampling.

Additional Comparison to FPT. The FPT [8] model is designed to predict flow between the conditional image I and a single timestep in the future, which differs from GARFIELD that models motion densely over time. In particular, FPT can only be fed future knowledge about the exact timestep τ it is supposed to predict while our method allows conditioning and prediction at different points in time. As we considered the task of motion planning, we evaluated FPT in Tab. 1 by rerunning it for different time horizons with spatio-temporally sparse conditioning, matching our method. However, GARFIELD can also be applied in the setting FPT was trained for, conditioning only a single timestep and predicting remaining points for that step. Tab. E.4 demonstrates that GARFIELD is competitive in this setting. While FPT achieves slightly superior performance, our method achieves similar performance without training for such motion interpolation tasks.

Additional Comparison to Motion-I2V. Motion-I2V [54] predicts optical flow to animate images based on text prompts and drags which correspond to our goals. However, Motion-I2V is designed to accept full tracks over time as input which is inherently different from our spatio-temporally sparse conditioning. To make the baseline comparable, we follow the protocol for sparse conditioning outlined in the Motion-I2V paper [54] and linealy interpolate sparse known goals to full tracks for the comparison in Tab. 1. However, Motion-I2V performs noticeably better in its native setting with full tracks as conditioning (s. Tab. E.5), but this requires dense knowledge of where the scene element will be at every timestep.

Method	FDE with $\lvert\widehat{T}\rvert$ End-Point Goals					
	0	1	2	4	8	16
FPT [8]	0.017	0.017	0.015	0.014	0.013	0.009
GARFIELD (Ours)	0.022	0.020	0.019	0.018	0.014	0.013

Table E.4: End-point motion interpolation. We compare our model's ability to infer where scene elements end up given end-point for other elements in the scene. Evaluated on held-out validation set of training data.

Further, the Motion-I2V repository recommends a cfg scale of 7.0 for inference, which we use throughout our main evaluation aligning. However, we find performance is improved when using cfg 2.0, which we attribute to higher variance predictions leading to better exploration of the search space. Tab. E.5 shows our method remains superior on all metrics for the sparse conditioning case. However, combining dense conditioning with insights from a cfg sweep, Motion-I2V approaches the quality of our motion planning.

More Details on Pedestrian Trajectory Prediction. We expand on the evaluation of pedestrian trajectory prediction in Tab. 3. We report zero-shot performance as well as performance after minimal finetuning. For finetuned results we train the encoder with the pointwise decoder model for 4k iterations with a batch size of 128. We modify our masking to always provide the initial eight positions as conditions, and train the model to predict the following twelve positions without future knowledge, matching the inference setting. We further mask the loss to only supervise prediction. We then finetune the full decoder by substituting the finetuned encoder and training the full decoder for 4k iterations in the same prediction setting. For the Stanford Drones dataset [49] we use the official train-test-split, while we use a leave-one-out finetuning for the ETH/UCY [31, 45] scenes. As GARFIELD requires RGB input we only use the ETH, Hotel, Univ, Zara01, and Zara02 scenes for finetuning and evaluation, matching the evaluation performed by IDM [35].

Tab. E.7 shows results for more baselines on the Stanford Drones Dataset (SDD) [49]. Without training for prediction, GARFIELD achieves zero-shot performance comparable with Social-LSTM [1]. After finetuning, GARFIELD outperforms Social-GAN [21] and is competitive with CGNS [32] and Goal-GAN [16]. Although we do not outperform recent domain-specific methods, this result highlights the general world knowledge learned by GARFIELD.

Similarly, Tab. E.6 shows results for more baselines and more subsets of the ETH/UCY datasets. Again, GARFIELD's zero-shot performance without ever training for prediction matches established approaches. While even our finetuned model does not outperform recent state-of-the-art baselines, the ADE of GARFIELD is below 1 m still providing useful information for use in downstream applications such as autonomous driving.

Note that the methods we compare against in Tab. E.7 and Tab. E.6 are specifically designed for pedestrian trajectory prediction and fail to predict the motion of other scene elements. Therefore, the usefulness of these methods in application is limited compared to GARFIELD's general understanding of kinematics.

Alternative Training on Public Data. In the main paper, we present a model trained on a diverse in-house dataset. To ensure reproducibility, we also train GARFIELD with the same configuration on a subset of the public Koala-36M [63] dataset. Table E.8 compares our main model to the model trained on public data. We find negligable differences in

Method	Num. Goals	CFG	EPE ↓	PCK ↑ @10%	PCK ↑ @1%	FDE ↓
Motion-I2V [54]	4	7.0	0.055	0.864	0.150	0.060
	16	7.0	0.033	0.957	0.263	0.037
	4	2.0	0.022	0.982	0.428	0.029
	16	2.0	0.016	0.992	0.577	0.030
	$4 \times H$	7.0	0.051	0.886	0.166	0.053
	$16 \times H$	7.0	0.027	0.981	0.312	0.030
	$4 \times H$	2.0	0.021	0.986	0.447	0.026
	$16 \times H$	2.0	0.013	0.996	0.602	0.016
GARFIELD (Ours)	4	1.0	0.020	0.973	0.544	0.026
	16	1.0	0.013	0.989	0.666	0.017

Table E.5: Extended comparison to Motion-I2V [54] When fed full trajectories with horizon H as input, Motion-I2V performs noticeably better at motion planning.

Methods	ETH		Hotel		Univ		Zara1		Zara2	
	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
Social-LSTM	1.09	2.35	0.79	1.76	0.67	1.40	0.47	1.00	0.56	1.17
Social-STGCN	0.64	1.11	0.49	0.85	0.44	0.79	0.34	0.53	0.30	0.48
Causal-STGCN	0.64	1.00	0.38	0.45	0.49	0.81	0.34	0.53	0.32	0.49
STAR	0.36	0.65	0.17	0.36	0.31	0.62	0.26	0.55	0.22	0.46
Expert-GMM	0.37	0.65	**0.11**	**0.15**	**0.20**	0.44	**0.15**	0.31	0.12	0.26
PCCSNET	**0.28**	0.54	**0.11**	0.19	0.29	0.60	0.21	0.44	0.15	0.34
Social-GAN	0.81	1.52	0.72	1.61	0.60	1.26	0.34	0.69	0.42	0.84
Goal-GAN	0.59	1.18	0.19	0.35	0.60	1.19	0.43	0.87	0.32	0.65
Social-BiGAT	0.69	1.29	0.49	1.01	0.55	1.32	0.30	0.62	0.36	0.75
MG-GAN	0.47	0.91	0.14	0.24	0.54	1.07	0.36	0.73	0.29	0.60
CGNS	0.62	1.40	0.70	0.93	0.48	1.22	0.32	0.59	0.35	0.71
PECNET	0.54	0.87	0.18	0.24	0.35	0.60	0.22	0.39	0.17	0.30
Trajectron++	0.39	0.83	0.12	0.21	**0.20**	0.44	**0.15**	0.33	**0.11**	**0.25**
LB-EBM	0.30	**0.52**	0.13	**0.20**	0.27	0.52	0.20	0.37	0.15	0.29
MID	0.39	0.66	0.13	0.22	0.22	0.45	0.17	0.30	0.13	0.27
IDM	0.41	0.62	0.15	0.25	**0.20**	**0.42**	0.17	**0.28**	0.12	**0.25**
GARFIELD (Ours)$_{\text{0-shot}}$	1.09	1.79	0.41	0.71	0.60	1.12	0.72	1.37	0.55	1.05
GARFIELD (Ours)$_{\text{FT}}$	0.84	1.44	0.29	0.56	0.58	1.19	0.33	0.65	0.33	0.67

Table E.6: ETH/UCY Pedestrian Prediction. We show an extended variant of Tab. 3 with all scenes from ETH/UCY [31, 45] and more baselines. Errors are in meters. Our method achieves competitive performance while enabling open-set motion modeling.

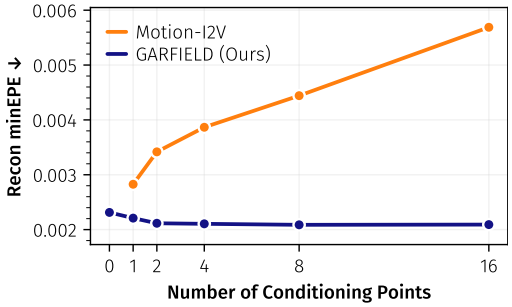


Figure F.4: Reconstruction Error GARFIELD produces accurate reconstructions of conditional points, showcasing its ability to serve as a representation for given points beyond a planning method.

performance. We provide checkpoints and results for the model trained on internal data, as we hope the diverse training data introduces more general world knowledge useful as a foundation in downstream domains.

F Extended Ablations

Reconstruction Quality. Fig. F.4 showcases reconstruction quality for points $\hat{T}$ provided as a part of $\mathcal{C}$ to the encoder. We find near-perfect reconstructions ($EPE < 0.003$; < 1 px on a 256^2 image) with GARFIELD across the entire range of conditioning density. Therefore, GARFIELD serves not only as a powerful motion planning method, but is also able to encode given trajectories with minimal loss of information, which is not possible with previous approaches such as Motion-I2V [54].

Performance beyond Tracker Artifacts. GARFIELD is trained and evaluated using a pseudo-ground truth obtained by running the off-the-shelf point tracker TapNext [74] on video data. Experiments using human annotated data [31, 45, 49] and CoTracker3 [29] in Tab. 2 and Tab. 3 partially show that our model generalizes beyond its original training setting. Still, tracker artifacts in supervision and evaluation targets may obstruct evaluation of the true motion understanding. Consequently, we ablate GARFIELD's motion understanding by testing a model trained using TapNext [74] annotations on TapNext as well as CoTracker3 [29] annotated open-set clips from OpenVid-1M in Tab. F.9a. We find that GARFIELD performs similarly well when using either tracker as the prediction target. Furthermore, both TapNext and CoTracker3 estimate similar tracks when given the same videos and are seeded with the same tracking queries (s. Tab. F.9b). While we do not inspect the biases of each tracking method further, this result validates that our model learns

Method	minADE-20	minFDE-20
Social-LSTM	57.00	31.20
Expert+GMM	10.67	14.38
SimAug	10.27	19.71
PCCSNET	8.62	16.16
Y-Net	8.97	14.61
Social-GAN	27.23	41.44
Goal-GAN	12.20	22.10
MG-GAN	13.60	25.80
CGNS	15.60	28.20
Trajectron++	8.98	19.02
PECNET	9.96	15.88
LB-EBM	8.87	15.61
MID	7.61	14.30
IDM	**7.46**	**13.83**
GARFIELD (Ours) 0-shot	36.77	49.86
GARFIELD (Ours) FT	19.92	34.98

Table E.7: Stanford Drone Dataset We show an extended variant of Tab. 3 for the Stanford Drone Dataset [49]. Errors are in pixels w.r.t. the original video resolution. Our method achieves comparable performance, without domain-specific adaptations.

(a): Point-wise $\mathbf{D}_{\psi}^{\mathrm{p}}$

$\lvert\hat{\mathbf{T}}\rvert$	EPE ↓		FDE ↓	
	Ours	Koala	Ours	Koala
0	0.017	0.015	0.023	0.020
4	0.013	0.012	0.017	0.016
8	0.011	0.012	0.015	0.015
16	0.011	0.011	0.015	0.015

(b): Full $\mathbf{D}_{\theta}$

$\lvert\hat{\mathbf{T}}\rvert$	EPE ↓		FDE ↓	
	Ours	Koala	Ours	Koala
0	0.017	0.016	0.022	0.021
4	0.014	0.014	0.018	0.018
8	0.013	0.013	0.016	0.016
16	0.012	0.012	0.015	0.015

Table E.8: Public vs Private Dataset We compare the effect of training our model with two different video datasets: an in-house dataset and a subset of Koala-36M [63]. Training on publicly available data achieves comparable performance, ensuring reproducibility of results. Evalauted on a subset of OpenVid-1M [40] data.

motion concepts beyond the artifacts produced by one tracking method.

Density Estimator Objective. We train our density estimator using a grid-Cross Entropy loss Eq. (4). Alternatively, we could distill the distribution learned by the point-wise decoder into a deterministic model by performing Monte-Carlo sampling with the point-wise decoder, constructing a heatmap, and training the density estimator with an MSE loss to regress the heatmap. With this, the alternative loss for the density estimator would become

$$\mathcal{L}_{\omega}^{\text{alt}} = \frac{1}{g^2} \sum_{(u,v)\in\mathcal{G}} \| p_{\omega}^{(u,v)}(t_{\tau}^{i} \mid \mathbf{z}_{\tau}^{i}) - p_{\psi,MC}^{(u,v)}(t_{\tau}^{i} \mid \mathbf{z}_{\tau}^{i}) \|_2^2, \tag{F.15}$$

where $p_{\omega}^{(u,v)}(t_{\tau}^{i} \mid \mathbf{z}_{\tau}^{i})$ is the predicted probability for grid cell (u, v) by the density estimator $\mathbf{D}_{\omega}^{\mathrm{d}}$ and $p_{\psi,MC}^{(u,v)}(t_{\tau}^{i} \mid \mathbf{z}_{\tau}^{i})$ is obtained by sampling N_{MC} realizations from the pointwise decoder $\mathbf{D}_{\psi}^{\mathrm{p}}$ conditioned on latent $\mathbf{z}_{\tau}^{i}$ and counting the relative frequency of a point-wise sample landing in grid cell (u, v).

Training with this objective introduces a fundamental trade-off between accurate targets $p_{\psi,MC}^{(u,v)}(t_{\tau}^{i} \mid \mathbf{z}_{\tau}^{i})$ and the time required to sample N_{MC} realizations, as $p_{\psi,MC}^{(u,v)}(t_{\tau}^{i} \mid \mathbf{z}_{\tau}^{i}) \longrightarrow p^{(u,v)}$ if $N_{MC} \longrightarrow \infty$. Furthermore, Tab. F.10 shows that a distilled density estimator achieves substantially worse calibration, resulting in a detrimental increase in discrete energy scores. We conclude that the distilled density estimator overfits to artifacts produced by the point-wise decoder when N_{MC} is too small, while the cross-entropy variant has a more stable supervision target in the ground-truth position.

Robustness of Entropy-based Conditioning Signal. We evaluate the robustness of using entropy as a signal for where to provide additional conditioning. For this purpose we perturb the additional conditioning provided with random noise, mimicking uncertainty of human annotators about correct constraints. Fig. F.5 reveals that entropy-based conditioning is

(a): Inference performance

$\lvert\hat{\mathbf{T}}\rvert$	EPE ↓		FDE ↓		PCK@1% ↑	
	TN	CT	TN	CT	TN	CT
0	0.017	0.017	0.022	0.022	0.731	0.725
2	0.015	0.015	0.018	0.019	0.786	0.771
8	0.013	0.013	0.016	0.016	0.807	0.799
16	0.012	0.012	0.015	0.015	0.821	0.812

(b): Tracker Agreement

EPE ↓	0.003
PCK@10% ↑	0.995
PCK@1% ↑	0.942

Table F.9: Alternative Tracker Pseudo-Ground Truth. We evaluate the performance of a TapNext-trained model on TapNext (TN) [74] and CoTracker (CT) [29] annotated data and find similar performance. We further find high agreement between TapNext and CoTracker. Therefore, our model generalizes beyond artifacts of a specific tracker, learning useful motion representations. Evaluated on a subset of OpenVid [40] evaluation data.

$\lvert\hat{\mathbf{T}}\rvert$	Discrete Energy Score ↓	
	Cross Entropy	Distillation
0	0.061	0.202
2	0.043	0.199
8	0.030	0.198
16	0.026	0.197

Table F.10: Density Estimator Objective Training with a cross-entropy (CE) objective (Eq. (4)) results in a lower discrete energy score (dES; Eq. (C.14)) than training with a distillation objective (Eq. (F.15)) across any number of conditioning points $|\hat{T}|$. Evalauted on held-out validation set of training data.

substantially more robust than EPE oracle-based conditioning. Our conditioning variant not only achieves consistently better performance at the same perturbation level but also diverges more slowly under stronger perturbation.

Multi-Object Interactions. To test force-driven dynamics beyond kinematics, we train our model and FPT [8] on data from a synthetic Billiard simulation [18] and compare prediction accuracy with up to 16 balls. We select a time horizon of one second (40 steps, $\Delta t = 0.025\,s$) as we find this horizon is sufficient for multiple ball-to-ball interactions. Note that this requires FPT to internally model many-ball interactions, while GARFIELD is able to reason about step-by-step interactions. Tab. F.11 showcases GARFIELD achieves relatively stable performance when introducing more balls and thus forcing more ball interactions to occur. While performance slightly deteriorates when moving from two to eight balls, performance remains stable when increasing to 16 interacting scene elements. In comparison, FPT achieves substantially worse performance across the entire range of ball counts.

Fig. G.13 qualitatively shows Billiard interactions over time given the starting velocity and end-point conditioning. GARFIELD is able to correctly infer interactions at intermediate timesteps. This highlights GARFIELD's ability to understand and reason about multiple interacting entities.

Alignment of Samples with Density. As an additional verification of alignment between sampled trajectories by $\mathbf{D}_\theta$ and $\mathbf{D}^{\mathrm{p}}_\psi$ and decoded distributions from $\mathbf{D}^{\mathrm{d}}_\omega$ we compute the negative log-Likelihood (NLL) of samples under distributions from $\mathbf{D}^{\mathrm{d}}_\omega$. Fig. F.6 shows that samples from either $\mathbf{D}_\theta$ or $\mathbf{D}^{\mathrm{p}}_\psi$ are more likely under $\mathbf{D}^{\mathrm{d}}_\omega$ than the ground truth trajectory or trajectories from an alternative model (Motion-I2V [54]). This further strengthens the evidence for alignment between models that decode the same latent possibility space.

Single-shot Performance. Given only partial future knowledge multiple outcomes can be equally plausible. Therefore,

Method	EPE ↓			
	2 Balls	4 Balls	8 Balls	16 Balls
FPT [8]	0.260	0.237	0.225	0.218
GARFIELD (Ours)	0.140	0.182	0.196	0.186

Table F.11: Multi-object Interactions. GARFIELD shows consistent performance with more balls, highlighting it's ability to model interactions between many objects.

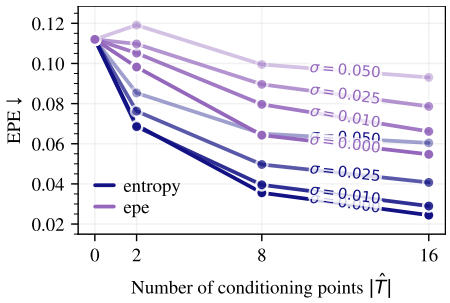


Figure F.5: Robustness of Entropy-based Conditioning. Selecting conditioning based on entropy is more robust to perturbed conditioning than EPE-based conditioning. Evaluated on held-out validation set of training data.

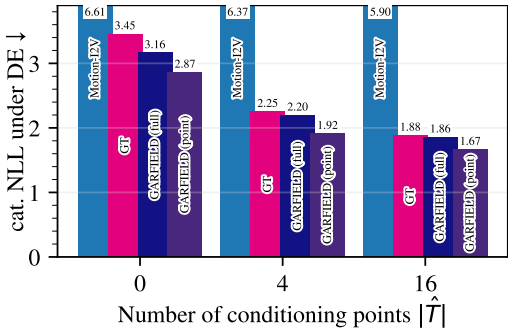


Figure F.6: NLL of samples under decoded density. The samples from $\mathbf{D}_\theta$ or $\mathbf{D}^{\mathrm{p}}_\psi$ are more likely under $\mathbf{D}^{\mathrm{d}}_\omega$ than the ground truth trajectories or samples from an alternative model (Motion-I2V [54]). This strengthens the assumption that all our decoders decode a shared possibility space.

trajectory prediction performance is typically evaluated in a best-of-N setting (cf. [1, 9, 11, 21, 51]). However, selecting the best trajectory might be impossible in application settings. Yet, Tab. F.12 highlights that GARFIELD remains highly competitive in the single-shot setting.

G Additional Qualitative Samples

For animated qualitative examples, we refer to the attached video samples summarized in `video_samples/index.html`.

We qualitatively compare our model to open-set baselines in Fig. G.7, qualitatively supplementing Tab. 1. For the TI2V models, we show tracks extracted from generated videos using TapNext [74] matching the motion planning evaluation setting. CogVideoX [71], LTX [22], predict realistic motion, however due to the lack of appropriate conditioning, their samples are dominated by unwanted camera motion and fail to match the ground truth in speed and direction. FPT [8] is unable to predict consistent trajectories due to modeling only the flow between two points in time, requiring independently rerunning the model for each horizon. Motion-I2V [54] and Track2Act [10] tend to predict low motion magnitudes, limiting their applicability to samples with complex motion. GARFIELD is able to sample motion which has the closest alignment to the target with as few as $|\widehat{T}| = 4$ known positions.

Furthermore, we show examples of GARFIELD's density estimation with $\mathbf{D}^{\mathrm{d}}_\omega$ in Fig. G.10, Fig. G.9 and Fig. G.8. GARFIELD becomes more certain when τ is close to 0 (the present) and/or more conditioning $\widehat{T}$ is given. Even when

Method	Goals	EPE↓
CogVideoX [71]	n/a	0.020
Motion-I2V [54]	4	0.036
FPT [8]	4	0.123
GARFIELD (Ours)	4	**0.015**

Table F.12: Single-Shot performance. GARFIELD remains competitive with baselines in the single-shot setting. Evaluated on a subset of the full OpenVid [40] validation data.

no future knowledge is provided, the ground-truth is within or close to the regions which GARFIELD considers likely and GARFIELD becomes more confident in the correct regions when conditioned on more knowledge about the future.

In Fig. G.11 and Fig. G.12, we visualize predicted trajectories for robotics and pedestrians. When given only the final goal position of each trajectory in the robotics setting, GARFIELD is able to plan reasonable motion for robotics without any domain-specific training. After fine-tuning GARFIELD, it can predict realistic trajectories of pedestrians in traffic scenes.

GARFIELD is able to handle and predict dynamics, e.g. multi-object inter-actions in Billiard simulations as shown in Fig. G.13. It is able to predict and resolve multiple collisions based on the final positions and initial velocities of each ball, which demonstrates its understanding of physical dynamics.

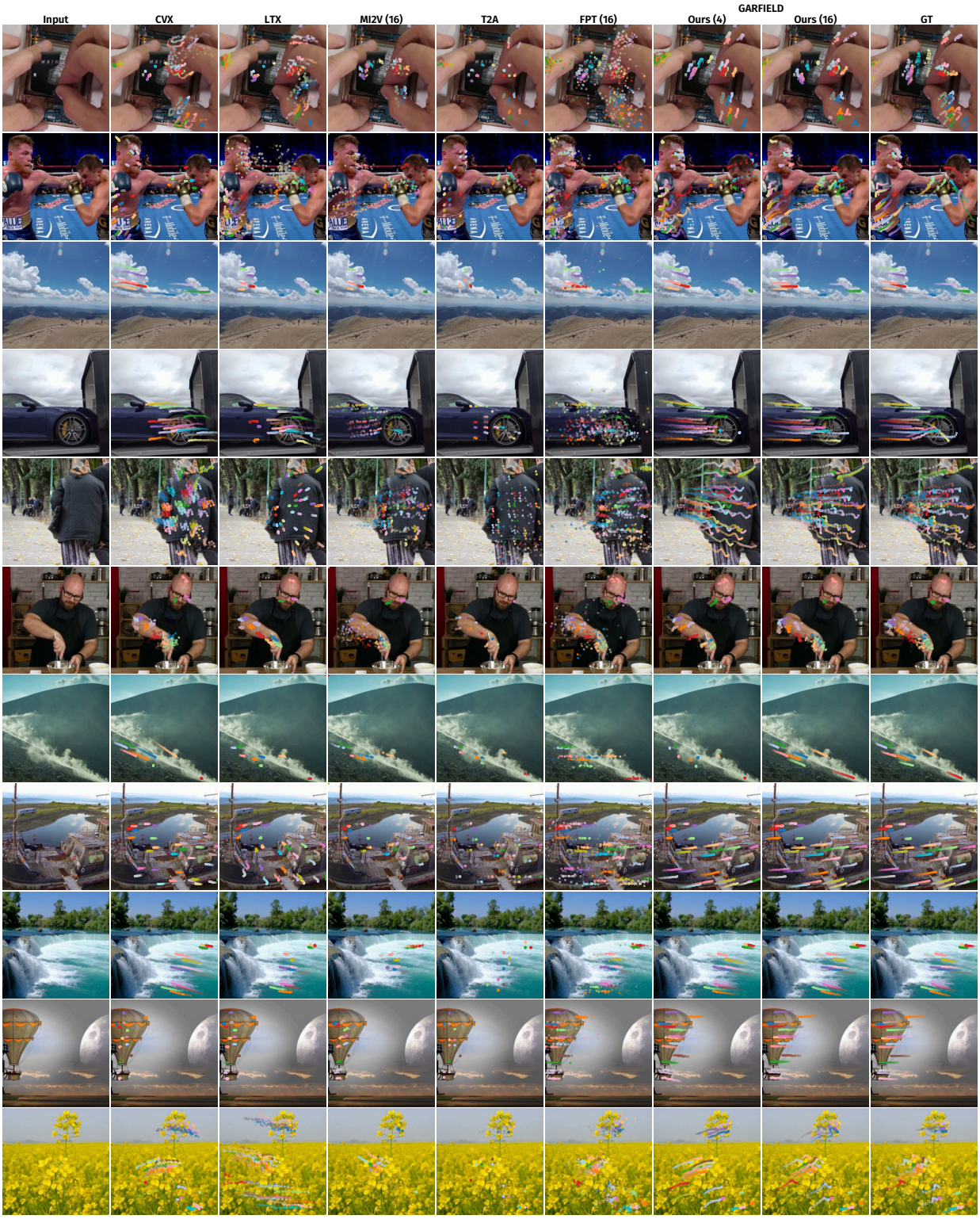


Figure G.7: Qualitative Comparison on OpenVid-1M [40]. We select samples with high motion magnitude and ignore static and nearly static points for visualization. Similar to Tab. 1 we compare CogVideoX [71](CVX), LTX [22], Motion-I2V [54] with 16 conditioning points (MI2V (16)), Track2Act [10] (T2A), and the Flow Poke Transformer [8] with 16 conditioning points (FPT (16)) to GARFIELD.

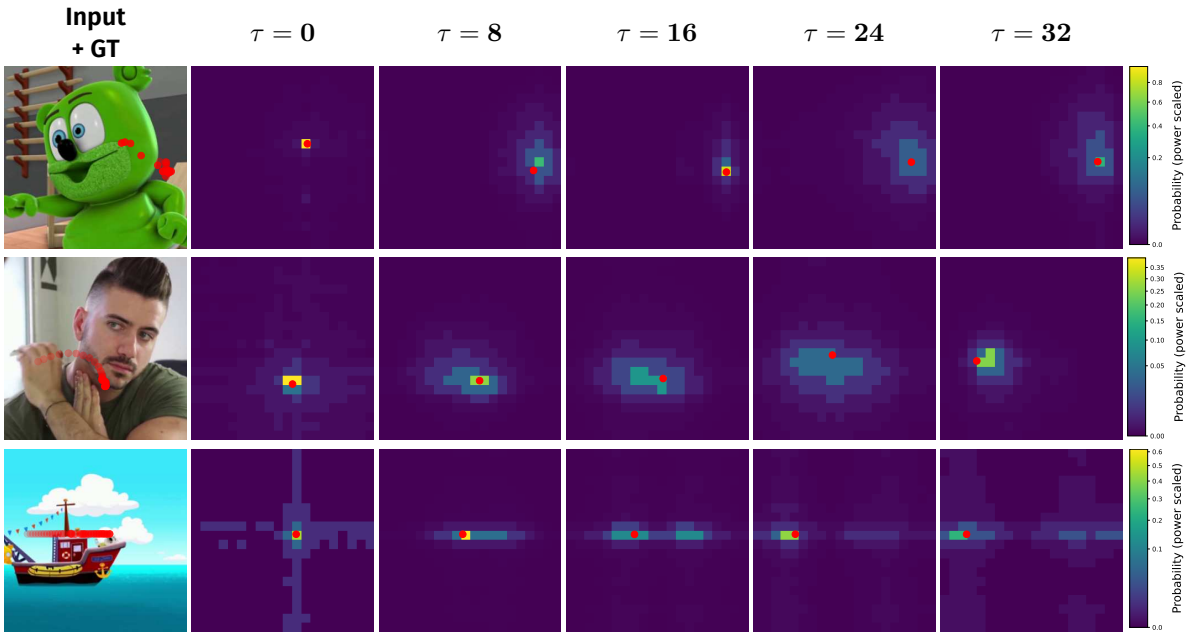


Figure G.8: Density Samples when conditioned on $|\widehat{T}| = 4$ future positions. GARFIELD consistently predicts high certainty close to the ground-truth, but remains reasonably uncertain about exact speed.

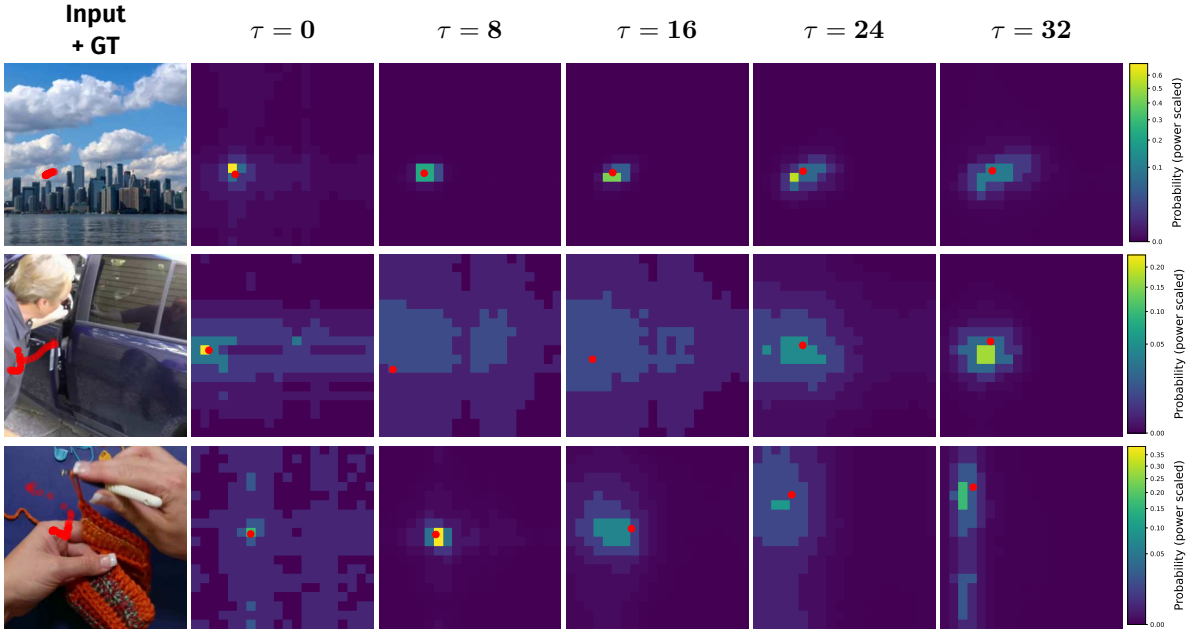


Figure G.9: Density Samples when conditioned on $|\widehat{T}| = 2$ future positions. While GARFIELD remains uncertain at some points in the spatio-temporal volume, motion tends to align with the observed direction.

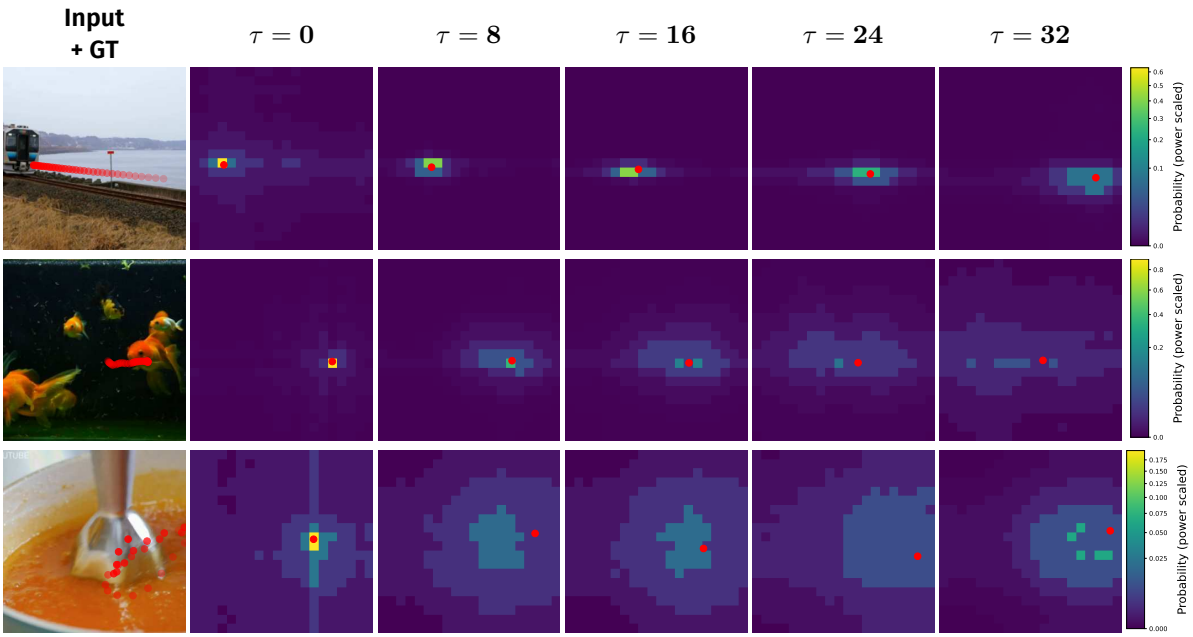


Figure G.10: Density Samples without future conditioning. Without future knowledge GARFIELD remains uncertain about exact motion that occurs but is able to estimate the correct area, based on it's general world knowledge.

Figure G.11: Qualitative Robotic Planning. Samples from GARFIELD on examples in the RT1 [12] (first three) and BridgeData [61] (last three) datasets. We show only trajectories exceeding a motion threshold and further subsample trajectories if necessary to achieve an interpretable visualization.

Figure G.12: Qualitative Pedestrian Trajectory Prediction. Trajectories predicted with GARFIELD for ETH/UCY [31, 45] (top row) and SDD [49]. Crosses ($\times$) indicate given positions, dots ($\cdot$) are predictions. Trajectories are subselected to trajectories that are visible in the first and last frame to ensure proper visualization.

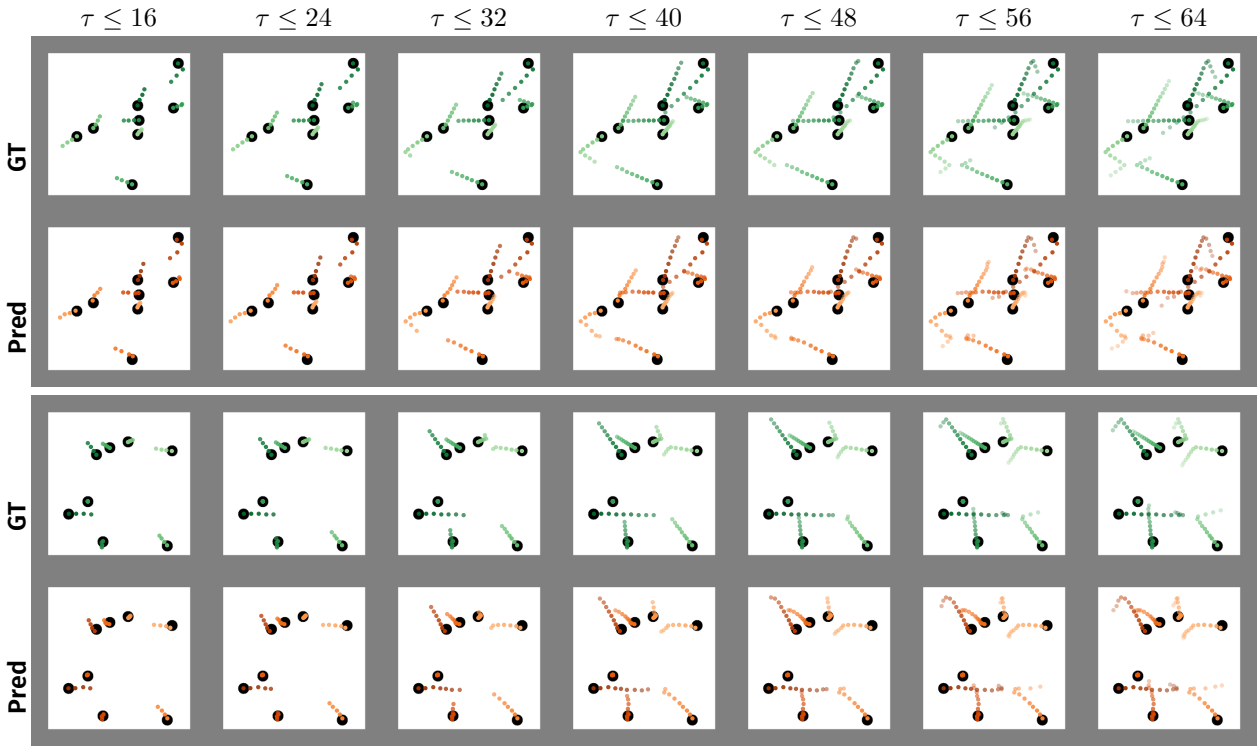


Figure G.13: Multi-object Interactions. GARFIELD is able to resolve dynamics of multiple objects colliding given conditioning on final positions and initial velocity.

H Copyright

The style used for this paper is adapted from the arXiv preprint *Discrete Flow Matching* (Gat et al., 2024), licensed under CC BY 4.0. Throughout the paper and figures/plots, we use Fira Sans (licensed under the OFL v1.1) for bold text.