What If: Understanding Motion Through Sparse Interactions

Stefan Andreas Baumann* Nick Stracke* Timy Phan* Björn Ommer

CompVis @ LMU Munich
Munich Center for Machine Learning (MCML)

Abstract

Understanding the dynamics of a physical scene involves reasoning about the diverse ways it can potentially change, especially as a result of local interactions. We present the Flow Poke Transformer (FPT), a novel framework for directly predicting the distribution of local motion, conditioned on sparse interactions termed "pokes". Unlike traditional methods that typically only enable dense sampling of a single realization of scene dynamics, FPT provides an interpretable directly accessible representation of multi-modal scene motion, its dependency on physical interactions and the inherent uncertainties of scene dynamics.

We also evaluate our model on several downstream tasks to enable comparisons with prior methods and highlight the flexibility of our approach. On dense face motion generation, our generic pre-trained model surpasses specialized baselines. FPT can be fine-tuned in strongly out-of-distribution tasks such as synthetic datasets to enable significant improvements over in-domain methods in articulated object motion estimation. Additionally, predicting explicit motion distributions directly enables our method to achieve competitive performance on tasks like moving part segmentation from pokes which further demonstrates the versatility of our FPT. Code and models are publicly available at compvis.github.io/flow-poke-transformer.

1. Introduction

A key feat of human visual intelligence is motion understanding, our ability to understand and predict the various ways the world around us *could potentially* change at a given point in time (see Fig. 1). Our cortex is not creating a mental video, focusing on how the colors of individual pixels change. Rather, we are constantly making predictions about the various ways individual objects or parts thereof could potentially move and deform [30]. We do not perceive the future as an unambiguously deterministic sequence of events but as a vast space of possibilities.

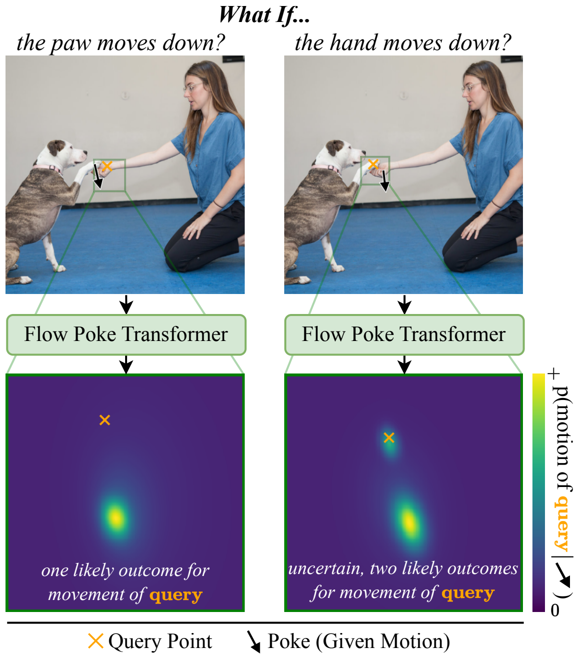


Figure 1. *What If:* Our Flow Poke Transformer directly models the uncertainty of the world by predicting *distributions* of how objects (✕) may move conditioned on some input movements (pokes, →). We see that whether the hand (below paw) or the paw (above hand) moves downwards directly influences the other's movement. Left: the paw pushing the hand down, will force the hand downwards, resulting in a unimodal distribution. Right: the hand moving down results in two modes, the paw following along or staying put.

It is natural to focus selectively on parts of a scene, infer how they might evolve, and reason about the underlying physical properties and interactions that drive change. This selective, probabilistic, and multimodal reasoning is rooted in the perceived inherent stochastic nature of the world. It is governed by the stochastic physical properties of complex systems and further compounded by the presence of agents with a complex, inaccessible internal state, lead by free will or other, from the outside often unapproachable causes.

This inherent uncertainty makes dense, deterministic predictions of future motion both impractical and ill-posed to represent real-world dynamics. The prediction of pixel-

[*] Equal Contribution.

perfect and long-term sequences [28] requires models to commit to one trajectory and, in so doing, ignore the rich multimodality of real-world outcomes. At best, such a prediction biases the scenario towards a single plausible future; at worst, it produces photorealistic frames that show limited understanding of physical processes, interactions, and constraints. In many situations, like autonomous agent systems, robotics, and automated planning, the ability to predict and process multiple possible outcomes of a given situation is more valuable than the naive assumption that events play out according to a single trajectory.

To address these issues, we propose a framework for representing the distribution of possible motions of parts of a scene. To control the degree of uncertainty, a human observer can interact with or perturb a scene with local "*pokes*", by nudging an object or applying a force. By repeating similar interactions, the multimodal nature of potential outcomes can be observed. Similarly, we allow conditioning the motion distribution on such sparse, localized pokes. Compared to traditional dense approaches, our method operates at a higher level of abstraction, predicting localized *distributions* of motion rather than committing to a single outcome. This approach aligns more closely with real-world dynamics where uncertainty and multimodality are intrinsic and actionable insights often emerge from reasoning about sparse, local changes rather than exhaustive dense predictions. These include aspects like the inherent interpretability of explicitly predicted distributions, such as identifying modes and quantifying uncertainty directly.

By avoiding dense (video) prediction, our model reframes motion prediction as a problem of capturing potential dynamics, directly predicting motion distributions. For instance, a poke applied to an unstable stack of blocks might cause it to topple in multiple ways, remain stable, or shift slightly without collapsing. We capture this variability, avoiding the pitfalls of dense video models that have to commit to one specific sample in the set of potential outcomes. This also addresses the impracticality of dense or long-term predictions, where compounding uncertainty renders dense outputs increasingly arbitrary.

Applications of our proposed model include (sparse) interactive simulation, where pokes guide scene exploration and the multimodal distributions of possible motions are directly captured, moving part segmentation, but also classic sampling of dense motion predictions. As opposed to optical flow estimation and tracking, where the future is given via a future frame, from which motion is estimated, we predict what future motion might be from only a single frame.

Overall, we present a step toward a more efficient, flexible, and detailed understanding of scene dynamics. We focus on the vast distribution of what could happen—optionally conditioned on sparse interactions instead of rendering specific futures. Our framework is not only efficient and scalable, but also conceptually aligned with the inherent uncertainties our human perception and reasoning is facing when dealing with our changing environment.

Our main contributions are as follows:

- Multimodal Distribution Prediction: we directly predict full distributions of potential motion instead of just enabling sampling from them, providing increased flexibility in applications over previous approaches, such as directly estimating uncertainties.
- Sparse Kinematics Modeling: our method reasons about sparse, local motion distributions across the scene. This balances efficiency with expressive power by focusing computational resources where they matter most.
- Generalizability: our method can learn a generic motion understanding from unstructured web videos, generalizing effectively to diverse, open-world data.
- Efficiency: our approach of sparsely modeling interactions enables sparse predictions with our method in 25ms and throughputs of more than 160k parallel predictions per second on a single modern GPU which is promising for real-time applications.

2. Related Work

Estimation of plausible motion for a given image or scene has been approached in various ways over the years. What makes this task particularly challenging is that it requires the model to have a physical understanding of how objects move in general, how they can be manipulated, and how they relate to each other. Many approaches directly predict a video from still images which makes it harder to access and leverage the underlying motion understanding of the model. Other approaches first predict a dense flow map which they use to later warp the images. However, more complex scenes can have multiple instantiations of realistic motion depending on the given conditioning, which we aim to model directly. In the following, we review various methods which have been studied in the literature.

Motion-based Editing A field that has recently gained attention is image editing using diffusion models by providing a set of pokes that indicate how specific parts of the image should move. [31] takes a GAN [13] generated image and warps it using motion supervision based on user-provided pokes. InstantDrag [36] on the other hand first predicts dense optical flow using a GAN and uses that as conditioning for a diffusion model to generate the final warped image.

Similar approaches have been used in video editing [5, 22, 45] that extend base models with ControlNets [54] or LoRAs [15] to condition the model on the desired motion. The goal is then to move entire objects according to a specific poke by selecting the object with a bounding box or entity representation [46]. Unlike our method, this general direction neglects an understanding of physical and realistic motion in exchange for precise adherence to the poke guidance and realistic inpainting of occluded regions.

Motion Generation Various works learn to hallucinate motion for static images [2, 11, 24, 33, 35, 43]. MoVideo [24] and Motion-I2V [35] use diffusion models to predict dense flow sequences given a start frame and use them to synthesize videos. Motion-I2V specifically allows conditioning on sparse movement information using "motion drags", similar to pokes, which is an improvement upon DragNUWA [50] that directly synthesizes dense RGB video from drags. The latter makes the actual motion prediction substantially less accessible because it needs to be estimated with an additional model like RAFT [40] or COTR [17], a property also shared by other methods [19, 20]. [43] predicts discrete bins of optical flow for static images with a classification loss. This enables them to model multiple flow fields for a single image. Im2Flow [11] predicts a single realization of continuous optical flow for an image and combines that with the image to boost action classification performance.

Learning how objects move and behave together can also be used as a general pretext task to build physical scene understanding. [2] introduced the concept of pokes as sparse motion conditioning to indicate how the poked object should move and directly synthesize dense RGB videos on limited-domain datasets. This is an unspecified problem as movement information is only available for a small number of poked pixels and the model needs to learn how the remainder of the scene moves. DragAPart [20] and the follow-up work PuppetMaster [19] focus on modeling the movement of individual parts of objects for a closed-domain, synthetic dataset (part-level motion). While these works focus on building a more fine-grained physical understanding, they directly predict the result of the poke(s) in RGB space. This makes the underlying motion representation harder to access and requires e.g. optical flow estimation between frames. Additionally, they do not provide any uncertainty estimation in the form of an underlying motion distribution, but simply render a single possible sample of the result space.

Other approaches focus on directly predicting a specific physical representation of motion. Generative Image Dynamics [23] learns oscillatory dynamics as commonly found in nature using Fourier-based motion representations. PhysDreamer [56] and PhysGaussian [49] extend the work of [23] from 2D to 3D scenes. While these approaches work well in their limited domains, they lack the flexibility to model the vast, often non-oscillatory motion space of the real world and are thus limited in the amount of general motion understanding they can obtain though training.

Generative Models in Computer Vision Generative models have recently become a cornerstone in computer vision, as they allow modeling tasks through full conditional distributions $p(\mathbf{y}|\mathbf{x})$ instead of reducing predictions to single-point estimates, such as the expectation $\mathbb{E}[\mathbf{y}|\mathbf{x}]$ often used in discriminative models. Major paradigms include GANs [13], diffusion models [14, 37], and autoregressive (AR) models [42]. GANs and diffusion models enable sampling from the modeled distribution but provide limited direct insight into its structure. Diffusion models, in particular, have demonstrated scalability to general data distributions and billions of parameters [32], whereas GANs are typically constrained to closed-set distributions. AR models, extensively applied in NLP [42] and increasingly adopted for vision tasks [3, 9, 51], directly model probability mass functions (PMFs) for discrete distributions and scale well to hundreds of billions of parameters [8]. However, the discrete nature of PMFs limits their applicability to real-valued problems, which are prevalent in vision tasks. Recent advances such as GIVT [41] and [21] have extended AR transformers to continuous-valued outputs while retaining the scalability of AR transformer models [10]. Specifically, [21] employ a diffusion model to sample from autoregressively predicted feature vectors, while GIVT directly parameterizes distributions as Gaussian Mixture Models (GMMs) with diagonal covariances, enabling direct access to the probability density function (PDF) for downstream applications. Our implementation builds upon the latter, while extending it to non-diagonal covariances to enable accurate modeling of motion distributions.

3. Method

3.1. Problem Setting

Given an image $\mathcal{I}$, we aim to model the movement of all visible points in the image and their interdependencies. To this end, our goal is to model the conditional distribution $p(\mathbf{f}(\mathbf{q})|\mathcal{P}, \mathcal{I})$ of the movement $\mathbf{f}(\cdot) \in \mathbb{R}^2$ of arbitrary *query* points $\mathbf{q} \in \mathbb{R}^2$ in the image conditioned on a set of N_p *pokes* $\mathcal{P} = \{(\mathbf{p}_i, \mathbf{f}(\mathbf{p}_i))\}_{i=1}^{N_p}$, each specifying the movement $\mathbf{f}(\mathbf{p}_i)$ at locations $\mathbf{p}_i \in \mathbb{R}^2$. Explicit conditioning on movement information given at specific points is crucial to enable the exploration of interactions and controlling movement prediction in the scene. Here, the movement $\mathbf{f}(\cdot)$ of a point describes its change in position from the current time t to a future time $t + \Delta t$, also referred to as *forward flow*.

3.2. Flow Poke Transformer

To model the movement distribution $p_\theta(\mathbf{f}(\mathbf{q})|\mathcal{P}, \mathcal{I})$, we use a transformer-based architecture, denoted as p_θ. Transformers [42] are especially well-suited for this task, as they are well-capable of working with sparse sequences due to token interactions only being implemented via the attention mechanism. We show a high-level overview in Fig. 2. We view each poke $(\mathbf{p}_i, \mathbf{f}(\mathbf{p}_i)) \in \mathcal{P}$ and each query point $\mathbf{q}_j \in \mathcal{Q}$ as

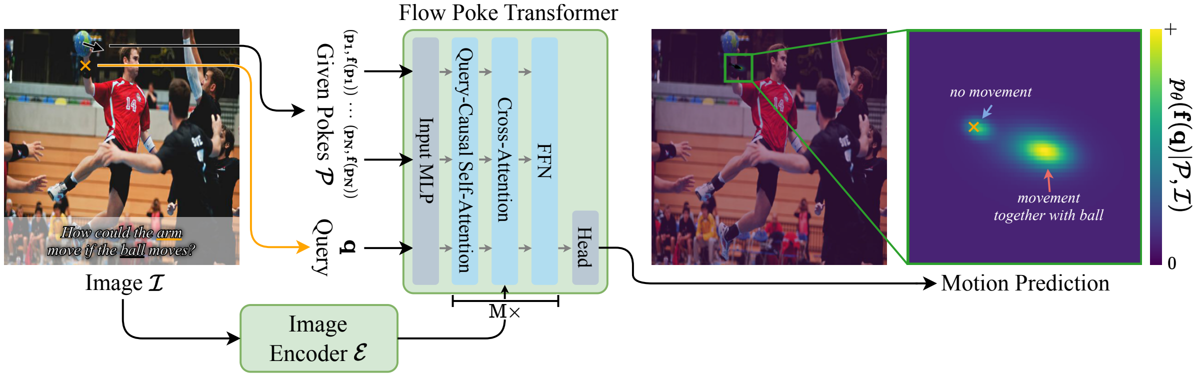

Flow Poke Transformer
How could the arm move if the ball moves?
Image $\mathcal{I}$
$(\mathbf{p}_1, \mathbf{f}(\mathbf{p}_1)) \dots (\mathbf{p}_N, \mathbf{f}(\mathbf{p}_N))$
Given Pokes $\mathcal{P}$
$\mathbf{q}$ Query
Input MLP
Query-Causal Self-Attention
Cross-Attention
FFN
Head
M×
Image Encoder $\mathcal{E}$
no movement
movement together with ball
Motion Prediction
+
0
$p_\theta(\mathbf{f}(\mathbf{q})|\mathcal{P},\mathcal{I})$

Figure 2. **High-level Model Architecture Overview**. Given an image $\mathcal{I}$, a set of given pokes $\mathcal{P}$ (visualized as arrows →), and query positions $\mathbf{q}$ (×), our model directly predicts an explicit distribution of the movement at each query position. The flow poke transformer cross-attends to features from a jointly trained image encoder to incorporate visual information. Crucially, our architecture represents movement at individual points $\mathbf{q}$ (enabling sparse & off-grid motion processing) and directly predicts continuous, multimodal output distributions.

individual tokens. Each poke's movement $\mathbf{f}(\mathbf{p})$ is encoded at the input using a Fourier embedding, while query tokens are set to a learned embedding. Positional encoding is implemented using relative positional embeddings [38], allowing positions to be set with arbitrary precision without needing to conform to any grid. This is important, as it enables training the model with high-quality but sparse and off-grid flow obtained via optical tracking. During self-attention, queries only attend to themselves and pokes, not to other queries. This enables evaluating the distribution $p_\theta(\mathbf{f}(\mathbf{q})|\mathcal{P},\mathcal{I})$ for multiple queries $\mathbf{q}_j$ in parallel, which is crucial for efficient dense flow predictions. The image $\mathcal{I}$ is encoded separately using a vision transformer, resulting in a set of spatial encoded image tokens $\mathcal{E}(\mathcal{I})$. The poke and query tokens then cross-attend to the image tokens, with spatial information again encoded using relative positional embeddings [38].

To obtain the movement distribution $p_\theta(\mathbf{f}(\mathbf{q})|\mathcal{P},\mathcal{I})$, a projection head at the transformer's output directly predicts a Gaussian Mixture Model (GMM), enabling real-valued distributional predictions following GIVT [41]. The distribution being directly accessible in this manner enables a range of additional capabilities, such as directly capturing multi-modal distributions in a single forward pass or enabling the fine-grained quantification of uncertainty. Unlike [41], we parametrize each component n using a full covariance matrix $\mathbf{\Sigma}^{(n)} \in \mathbb{R}^{2\times2}$ instead of a purely diagonal one, greatly increasing the prediction's degrees of freedom. The positive semi-definiteness of the covariance matrix is ensured by the model predicting a lower triangular matrix $\mathbf{L}^{(n)} \in \mathbb{R}^{2\times2}$ with a positive diagonal (by soft-clipping to a lower threshold), from which the covariance matrix is computed as $\mathbf{\Sigma}^{(n)} = \mathbf{L}^{(n)}(\mathbf{L}^{(n)})^\top$. Overall, this results in the predicted N-component GMM

$$p_\theta = \textstyle\sum_{n=1}^{N} \pi^{(n)} \cdot \mathcal{N}(\boldsymbol{\mu}^{(n)}, \mathbf{\Sigma}^{(n)}), \tag{1}$$

with component mixture coefficients $\pi^{(n)}$ and means $\boldsymbol{\mu}^{(n)}$.

A specific challenge with learning motion understanding from open-world web videos is that camera movement can dominate the overall motion distribution of a frame. Only training on videos with static cameras is not viable, as it would limit potential training data too much. We address this by replacing the typical normalization layers in the transformer with adaptive normalization layers [16], using which we condition the model on whether the camera is static, which we detect by whether a significant fraction of the scene's content is static. This allows us to learn motion prediction on general videos.

Training Objective We directly train our model to minimize the negative log-likelihood (NLL) of a ground truth flow $\mathbf{f}(\mathbf{q})$ of the random query point $\mathbf{q}$, conditioned on a random set of flow pokes $\mathcal{P}$

$$\begin{aligned}&\mathcal{L}(\mathbf{f}(\mathbf{q}), \mathcal{P}, \mathcal{I}; \theta) = -\log p_\theta(\mathbf{f}(\mathbf{q})|\mathcal{P},\mathcal{I})\\&= -\log\Big(\textstyle\sum_{n=1}^{N} \pi^{(n)} \mathcal{N}(\mathbf{f}(\mathbf{q})|\boldsymbol{\mu}_\theta^{(n)}(\mathcal{P},\mathcal{I}), \mathbf{\Sigma}_\theta^{(n)}(\mathcal{P},\mathcal{I}))\Big).\end{aligned} \tag{2}$$

Specifically, we compute the loss for image $\mathcal{I}$ conditioned on random sets of pokes $\mathcal{P}^{(i)}$ of length $|\mathcal{P}^{(0)}| = 0, \dots, |\mathcal{P}^{(N_p)}| = N_p$. Predicting the flow distribution at N_q different random query positions $\mathbf{q}$ per set of pokes. To enable efficient training, we introduce a variation of teacher forcing [39], typically used to train autoregressive transformers [42]. We select the set of random pokes such that $|\mathcal{P}^{(0)}| \subset |\mathcal{P}^{(1)}| \subset \dots \subset |\mathcal{P}^{(N_p)}|$. We then use a causal attention mask on the poke tokens and let the queries for each set of pokes individually attend to all the pokes in their respective set. We call the resulting attention pattern *query-causal attention* (see Fig. A for visualizations). As opposed to training with independent sets of pokes and full self-attention for all sets of pokes and queries, this reduces the computational complexity from $\mathcal{O}(N_p^2 \cdot N_q^2)$ to $\mathcal{O}(N_p^2 + N_p \cdot N_q)$ for the same number of trained predictions.

Since N_p can be large during training, this substantially improves performance and enables efficient training.

3.3. Downstream Applications

Our method's primary goal is to enable efficient and interpretable modeling of multimodal movement distributions of different parts in scenes. We achieve this by modeling the conditional distribution of the movement of query points $\mathbf{q}$ in the image $\mathcal{I}$ given the movement of any number of pokes $\mathcal{P}$ to condition on. As our model directly makes probability density functions for each query's movement accessible and captures its multi-modality (c.f., Fig. 3), the distribution of potential movements can be directly interpreted. Besides modeling this relation of movement of different parts in a scene, this also enables other direct and indirect downstream tasks and applications, which we describe in this section.

Dense Motion Prediction The conditional distribution our method learns to model can also be used to predict a dense grid of queries $\mathcal{Q}$. This prediction can be done both purely conditioned on the reference image or given reference pokes $\mathcal{P}$. This motion can be obtained by predicting the pointwise flow distributions in parallel or via autoregressive sampling. To sample from the joint distribution $p(\mathbf{f}(\mathcal{Q})|\mathcal{P},\mathcal{I})$ that models the precise interactions between all points in the image, we employ autoregressive sampling. Iteratively, we predict the flow distribution for a random query $\mathbf{q}_i \in \mathcal{Q}$, sample a flow instance from the conditional distribution $\mathbf{f}(\mathbf{q}_i) \sim p_\theta(\mathbf{f}(\mathbf{q}_i)|\mathcal{P},\mathcal{I})$, and add the query to the set of pokes $\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathbf{q}_i\}$. This results in individual coherent samples $\mathbf{F}_{\text{sample}} \sim p_\theta(\mathbf{f}(\mathcal{Q})|\mathcal{P},\mathcal{I})$ from the distribution of possible dense flows. We show qualitative examples of sampled dense flow in Fig. 6. For parallel sampling, we compute the mean dense flow $\mathbf{F}_{\text{mean}} = \mathbb{E}\left[\mathbf{f}(\mathcal{Q})|\mathcal{P},\mathcal{I};\theta\right]$ in a pointwise manner for all queries $\mathbf{q}_i \in \mathcal{Q}$ in parallel. This provides efficient, high-quality flow predictions (see, e.g., appendix Fig. F), but also results in mode averaging.

Segmenting Moving Parts Segmenting parts that move together is a task introduced in [20] and is useful for various applications such as predicting affordances. Given a poke $(\mathbf{p},\mathbf{f}(\mathbf{p}))$, the aim is to segment the image into parts that would move in response to it. Unlike [20], we do not need to rely on involved methods for extracting and comparing internal feature activations of our model for this task. Instead, our model enables direct quantification of the effect a movement $\mathbf{f}(\mathbf{p})$ of a point $\mathbf{p}$ has on another point $\mathbf{q}$ by measuring the relative entropy between the conditional and unconditional distribution, i.e., how much conditioning on $\mathbf{p}$ changes the movement distribution of $\mathbf{q}$. This is done using the Kullback-Leibler (KL) divergence

$$D_{\mathrm{KL}}(p_\theta(\mathbf{f}(\mathbf{q})|(\mathbf{p},\mathbf{f}(\mathbf{p})),\mathcal{I}) \parallel p_\theta(\mathbf{f}(\mathbf{q})|\mathcal{I})). \tag{3}$$

Specifically, if the movements of $\mathbf{q}$ and $\mathbf{p}$ are independent, the conditional distribution $p(\mathbf{f}(\mathbf{q})|(\mathbf{p},\mathbf{f}(\mathbf{p})),\mathcal{I})$ is equal to the marginal distribution $p(\mathbf{f}(\mathbf{q})|\mathcal{I}))$, and thus, the KL divergence in Eq. (3) is zero. Otherwise, it quantifies the change in movement distribution, and, thus, the motion interdependencies of different parts of the scene. We efficiently approximate the KL divergence using the matched bound approximation [12]. This can then be computed over all points $\mathbf{q}$ in the image $\mathcal{I}$ in parallel and directly quantifies the effect the given movement of $\mathbf{p}$ has on each point $\mathbf{q}$.

4. Experiments

4.1. Dataset and Implementation Details

For general pretraining, we train on a random 3.8M video clip subset of WebVid [1]. The wide variety of concepts present in WebVid enables our model to learn a general representation for motion instead of being limited to a specific domain, such as face-only videos. We train our model with flow from optical tracks using CoTracker3 [18] for a random 48-frame interval from each clip using a uniform 48^2 grid from the respective start frames.

The image encoder and the poke transformer are ViT-Base transformers [7] for a total parameter count of 220M. We use RoPE [4, 38] both for self-attention between flow tokens and for cross-attention to image tokens. We initialize the image encoder with DINOv2-R [6, 29] to make training more efficient but keep the weights unlocked. Jointly training the full model is essential, as DINOv2 does not have good instance segmentation capabilities (see Sec. B for additional details), which are essential for our task. We pass images to the vision encoder at a resolution of 448^2 to obtain a 32^2 grid of visual embedding tokens. The model is trained in bfloat16 precision for 800k steps using AdamW [26] with a learning rate of 5e-5, at a global batch size of 32 images, which is increased to 128 after 250k steps. Per image, we sample sets of random pokes of sizes $0, 1, \dots, 128$, and compute losses on $N_q = 15$ random query points per set of pokes. This results in a global batch size of 61,440 (245,760) queries. Overall, training this model took 7 days on 2 Nvidia H200s. We also train a second model on a dataset of 5M open-set video clips we collected, with optical tracks obtained using TAPNext [57]. Here, we simplify the training setup to use a batch size of 128 across the whole training, and add a cosine decay [25] for the learning rate after an initial warmup. A further optimized training setup allows us to train this model to 1M steps in 24h on 8 Nvidia H200s. We provide additional details and ablations in appendix Secs. A and B, and explore the potential for extension to 3D motion in Sec. C.

Without inference optimizations such as quantization or 8-bit inference, a single conditional movement distribution prediction for query in an image can be obtained in less than 25ms of delay on a single H200. This makes our model

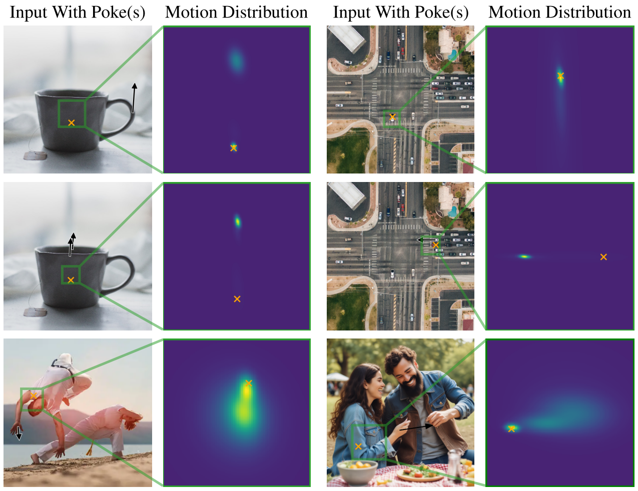


Figure 3. **Multimodal Motion Distribution Prediction**. We condition on one or multiple pokes (→) and then query the motion distribution of specific points (×). Our model's predictions capture the multi-modal nature of motion and exhibit understanding of interactions, such as only lifting the cup by its handle not necessarily causing the whole cup to move upwards, while grabbing it at stable points does. It also demonstrates prior understanding from scenes, such as a car in an intersection being more likely to move forwards than backward and cars in traffic likely moving together.

applicable for real-time applications. Throughput (with parallel predictions) is about 160k predictions per second per image, thanks to our query-causal attention implementation.

We generally evaluate predicted motion at a resolution of 64^2 unless specified otherwise. We primarily rely on endpoint error $\text{EPE} = \|\hat{\mathbf{f}}(\mathbf{q}) - \mathbf{f}_{\text{GT}}(\mathbf{q})\|_2$, which measures the difference between the true motion $\mathbf{f}_{\text{GT}}(\mathbf{q})$ and the predicted motion $\hat{\mathbf{f}}(\mathbf{q})$. Additionally, we also compute the percentage of correct keypoints $\text{PCK} = \mathbb{E}[\|\hat{\mathbf{f}}(\mathbf{q}) - \mathbf{f}_{\text{GT}}(\mathbf{q})\|_2 < \alpha]$, with $\alpha = 1\text{px}$ unless specified otherwise.

4.2. Evaluation of FPT's Key Abilities

FPT's Ability to Predict Movement Distributions We observe our model's predicted distributions $p_\theta(\mathbf{f}(\mathbf{q})|\mathcal{P}, \mathcal{I})$ given an image $\mathcal{I}$ of a scene and conditioned on a sparse set of pokes $\mathcal{P}$ in Fig. 3 (see appendix Fig. E for additional samples). Qualitatively, our model exhibits an understanding of physical phenomena and interactions, predicting realistic movement distributions for the given pokes in the context of the respective scenes. Most importantly, it captures the multi-modality of potential movements in different circumstances and their variability/uncertainty. Our approach is trained in an open-world setting, being not limited to individual object categories, but, nevertheless, captures fine-grained details of specific objects' potential motion.

One can also draw samples from the joint distribution of movement of the whole scene $p_\theta(\mathbf{f}(\mathcal{Q})|\mathcal{P}, \mathcal{I})$ by autoregressive sampling. We show examples of such unconditional motion generations in Fig. 6. They successfully show diverse but realistic global scene motion.

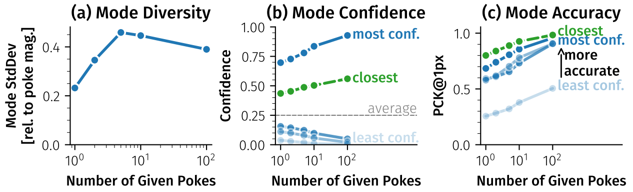


Figure 4. **Predicted Mode Analysis.** Values are computed at a resolution of 64^2. **(a)** Diversity of predicted modes is high, with mode variation covering a large fraction of poke magnitude. **(b)** One mode typically has a substantially higher confidence than others, which increases with given poke count. The mode **closest** to the ground truth consistently has a higher-than-average confidence. **(c)** More confident modes are more accurate as measured by PCK.

FPT predicts meaningful multimodal motion distributions. One important property that differentiates FPT from common motion modeling approaches is that it *directly* predicts the multimodality of possible future motions. For these multimodal predictions to be valuable, they should cover the diverse modes of possible motion and have meaningful predicted confidences $\pi^{(n)}$. We analyze these properties in Fig. 4. Generally, we find the modes to be highly diverse (Fig. 4a, cf. Figs. 3 and E), with them covering substantially different movements. As expected, the multimodal predictions reduce to primarily unimodal predictions when enough conditioning information is available to reduce the stochastic uncertainty of the future and discern one clear correct mode (Fig. 4b). Importantly, the confidence of the mode closest to the ground truth motion is consistently substantially higher than the average. This indicates that the model's confidence predictions are meaningful. Similarly, analyzing the modes' accuracy (Fig. 4c) shows that the model assigns higher confidences to modes more likely to be correct. Still, secondary and tertiary predicted modes are also meaningful, as indicated by the accuracy of the mode closest to the ground truth exceeding that of the most confident one.

FPT's predicted distributions accurately model uncertainty. We evaluate the predictive quality of our model's predicted uncertainty w.r.t. true prediction error in Fig. 5. Specifically, we investigate the relation between the predicted distribution's standard deviation $\text{Std}[\mathbf{f}(\mathbf{q})|\mathcal{P}, \mathcal{I}; \theta]$ and the motion estimation error as measured with the endpoint error (EPE). We find that the predicted motion's error strongly correlates to the predicted uncertainty. This capability is independent of the approach to derive the single motion prediction from the predicted distribution. Sampling from the predicted distribution (Pearson $\rho = 0.66$), using its mean ($\rho = 0.64$), or using the most confident mode ($\rho = 0.62$) all lead to high predictive accuracy of the true prediction error compared to the ground truth.

Method	Trained On	1 Poke			2 Pokes			5 Pokes			10 Pokes			100 Pokes		
		EPE ↓	PCK ↑	LPIPS ↓	EPE ↓	PCK ↑	LPIPS ↓	EPE ↓	PCK ↑	LPIPS ↓	EPE ↓	PCK ↑	LPIPS ↓	EPE ↓	PCK ↑	LPIPS ↓
InstantDrag [36]	Faces	9.24	**0.193**	0.18	9.12	**0.196**	0.17	8.82	**0.197**	0.17	8.39	**0.198**	0.16	7.29	0.212	0.15
Motion-I2V [35]	**Generic (WebVid-10M, Zero-Shot)**	29.08	0.029	0.35	27.40	0.031	0.34	24.22	0.030	0.32	20.90	0.048	0.30	n/a	n/a	n/a
Ours	**Generic (WebVid-3.8M, Zero-Shot)**	**7.64**	0.150	**0.16**	**6.87**	0.154	**0.15**	**5.32**	0.167	**0.13**	**4.20**	0.183	**0.12**	**2.51**	**0.264**	**0.10**
	Generic (Open Set-5M, Zero-Shot)	7.99	0.150	0.17	7.02	0.154	0.16	5.50	0.158	0.14	4.17	0.182	0.12	2.44	0.285	0.10

Table 1. **Face Motion Generation Evaluation**. We evaluate the accuracy of predicted motion on TalkingHead-1KH [44] given a starting frame and one or more pokes (partially) defining the head movement. Our method performs substantially better in a zero-shot comparison to Motion-I2V, which was also trained in a generic setting. Compared to InstantDrag, which was trained for specifically this setting, our method achieves a substantially better endpoint error (EPE) but slightly worse PCK for low poke counts, highlighting our model's capability to perform competitively with purpose-trained methods while being generic. It can also make more efficient use of the available information, achieving greater accuracy gains from additional pokes compared to other methods. When using the predicted motions to warp the source image, our method consistently outperforms others.

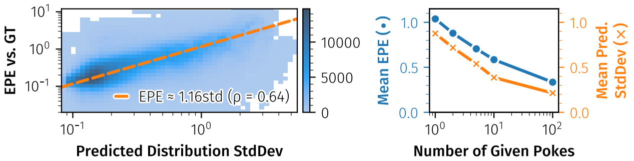


Figure 5. **Uncertainty Calibration.** We find that the motion prediction error measured by EPE strongly correlates with the predicted uncertainty (Pearson $\rho = 0.64$). This relationship holds for low & high numbers of given pokes.

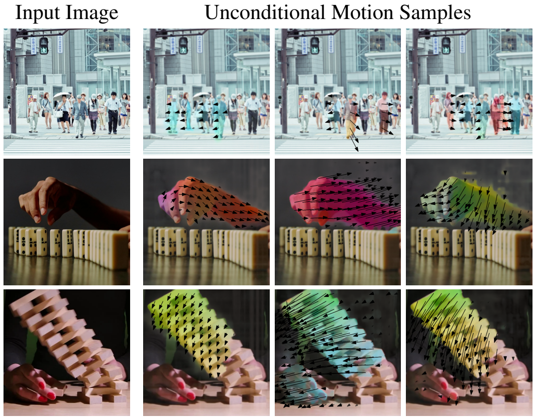


Figure 6. **Unconditional AR Motion Sampling**. We show samples of generated flow without prior motion conditioning on pokes. Our model can generate a wide variety of realistic motions.

4.3. Comparisons on Motion Prediction

To enable quantitative comparisons of our model's motion understanding with existing methods, we evaluate on predicting dense flow from sparse flow pokes, given a starting frame. We compare against baseline methods in the setting they have been trained on to enable fair comparisons. To evaluate against DragAPart [20] and PuppetMaster [19], which only generate images/videos based on drags, we extract flow from the source to the generated image using RAFT [40].

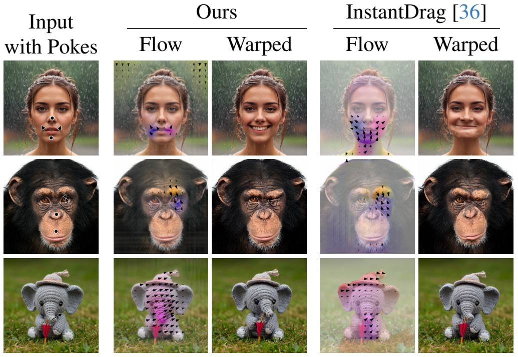


Figure 7. **Fine-grained Face Motion Control**. We show fine-grained zero-shot poking results on faces and compare against InstantDrag [36], which was trained for this task. We further visualize the predicted motion as warps using I-D's face warping model.

Face Motion Estimation We compare against InstantDrag [36], which was trained on CelebV-Text [52] in their evaluation setting – poke-conditioned motion prediction on aligned faces on TalkingHead-1KH [44]. The qualitative results (see Fig. 7) show that our model tends to predict more accurate and localized motion. This can also be observed when visualizing the motion by warping the image. For quantitative evaluations, we extract chunks of length 0.8s (following [36]) and use CoTracker3 [18] at a grid size of 128^2 to obtain the target motion from the start to the end frame. Then, we condition on $N \in \{1, 2, 5, 10, 100\}$ pokes $\mathcal{P}$, where the first poke is chosen to be the one with the largest flow magnitude, and the others are sampled randomly, and compare the dense predicted motion to the target downsampled by a factor of two. We also compare quantitatively with [35], which was trained generically. We compare favorably to both methods in EPE independent of the number of given pokes, indicating further that our predicted flow is more precise (Tab. 1). The PCK is slightly worse than the non-generically trained InstantDrag for low poke counts but catches up with more conditioning. When using the respective method's generated motion to perform image warping using InstantDrag's warping stage, our method consistently outperforms both others, as measured by the LPIPS [55] distance to the ground truth images.

Articulated Objects DragAPart [20] and PuppetMaster [19] were explicitly trained on part-level object motion on a synthetic dataset, Drag-A-Move [20] (DAM). To enable comparisons with them, we evaluate on the test set of DAM, which contains synthetic images of furniture with

Method	Trained On	**(a)** Motion Est.		**(b)** Moving Part Segm.
		EPE ↓	PCK ↑	mIoU ↑
Motion-I2V [35]	**Generic (Zero-Shot)**	33.27	0.043	0.073
DragAPart [20]	Objects (DAM)	9.69	0.514	0.273†
PuppetMaster [19]	Objects (DAM + OAHQ)	9.62	0.472	0.112
Ours	**Generic (WV, Zero-Shot)**	12.74	0.191	0.287
Ours (fine-tuned)	Generic → DAM	**3.57**	**0.834**	**0.572**

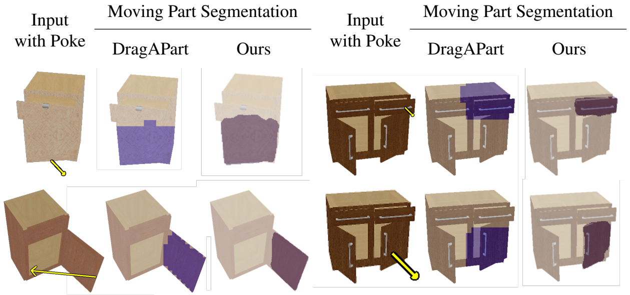
†taken from original publication, our evaluation yields 0.228.

Table 2. **Articulated Object Motion Estimation.** We compare motion (flow) estimation (a) and moving part segmentation performance (b) on Drag-A-Move [20] (DAM). On zero-shot motion estimation, our model substantially outperforms the other zero-shot method M-I2V, while not being much worse than specifically trained methods. When adapted, our method significantly outperforms previous approaches. In moving part segmentation, even our generic model outperforms other, in-domain models.

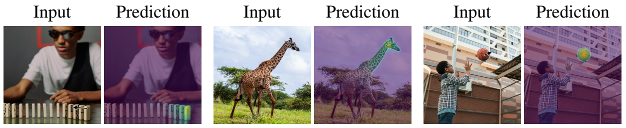
one or more pokes $\mathcal{P}$ that define the movement of one or more articulated parts of that object. It also provides ground truth dense flow at a resolution of 512^2, which we compare against. As DAM is significantly dissimilar from our train set, we evaluate our model both in a zero-shot and fine-tuned setting. For the fine-tuned setting, we fine-tune our model on DAM for 30k steps at a batch size of 128 with an exponentially decaying learning rate of 5e-7 that halves every 10k steps. Our quantitative evaluation (see Tab. 2, for additional qualitative samples see appendix Fig. D) shows that, in a zero-shot setting, our model's predicted motion is substantially more accurate than Motion-I2V [35] but less accurate than the specifically trained DragAPart and PuppetMaster. This demonstrates that our model generalizes to out-of-distribution data comparatively well, but, expectedly, falls short of models explicitly trained for this OOD domain. Once fine-tuned, our model outperforms even the purpose-made DragAPart and PuppetMaster by a wide margin. We also show qualitative examples in Fig. D. These find Motion-I2V to hallucinate substantial motion independently from the given pokes and struggle with matching the precise flow direction and magnitude for the given pokes. The extracted flow from DragAPart also seems to face struggles in complex multi-poke situations, which our model handles better.

4.4. Segmenting Moving Parts

We perform moving part segmentation with our method by thresholding the KL divergence between the pointwise unconditional motion distribution and the pointwise motion distribution conditioned on a specific poke (Eq. (3)). We show qualitative results in Fig. 8 and compare quantitatively against other methods, similarly thresholding the flow magnitude in Tab. 2b. Here, we find that our method, especially when finetuned in-domain, outperforms DragAPart [20], which introduced this benchmark, and the other methods by a wide margin. A large part of this gain can be attributed to our divergence-based score, which leverages FPT's unique property of directly predicting distributions. Without it, our method achieves an mIoU of 0.415 – still outperforming the previous state-of-the-art by a wide margin, but less so. Our method is also robust to threshold choice – halving/doubling it leads to mIoUs of 0.54/0.52 respectively.



(a) We directly replicate Fig. 7 from DragAPart [20] with our method. Our method provides spatially continuous predictions and makes fewer critical mistakes like segmenting the furniture body with the drawer (top right).



(b) Open-set moving part dependency visualization. The degree to which the movement of each part is influenced by the poke (→) is visualized as a heatmap, where brighter color means a higher degree of influence.

Figure 8. **Segmenting Moving Parts**. We show qualitative results for moving part segmentation, as introduced in [20], both on the Drag-A-Move dataset (a) and in a generic, open-set setting (b).

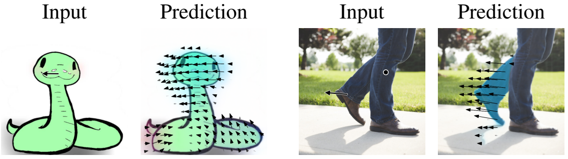


Figure 9. **Common Failure Cases**. Our model, generically pretrained on primarily realistic videos, does not generalize well to cartoons, causing parts of the background to be moved together with objects. Additionally, our model sometimes, but not consistently, jointly predicts the movement of shadows together with objects, which can be problematic for downstream use cases.

5. Conclusion

We introduce the Flow Poke Transformer (FPT), a novel framework for motion understanding that captures real-world dynamics' multi-modal and stochastic nature through interpretable distributions of local motion, conditioned on targeted interactions (*pokes*). Contrary to previous motion prediction approaches, FPT directly models the probabilistic distribution of possible outcomes, providing insights into the effects of physical interactions and inherent uncertainties.

Our evaluations demonstrate FPT's versatility across different domains and its generalization capabilities. Despite being designed and trained for sparse, general-purpose motion understanding, it also offers competitive performance in established tasks such as dense motion generation on faces or articulated objects. Importantly, while valuable for comparison, these evaluations do not fully reflect our method's primary strength – its ability to provide directly interpretable and useable predictions of motion distributions in interactive environments, bridging the gap between physical plausibility and efficiency. Furthermore, capabilities such as moving part segmentation directly emerge from our method's design. Overall, these results show our method's strength as a versatile, interpretable, general-purpose motion model. We envision this work as a foundation for more probabilistic, generalizable, and actionable approaches to motion understanding, paving the way for deeper insights into complex physical dynamics and future advancements in handling ambiguous and extreme out-of-distribution scenarios (Fig. 9).

Acknowledgment

We would like to thank Mahdi M. Kalayeh for helpful discussions and feedback. We would further like to thank Jannik Wiese, Kim-Louis Barwig, Enrico Shippole, Ming Gui, Thomas Ressler-Antal, Olga Grebenkova, Paul Hofman, Owen Vincent, and the anonymous reviewers. This work was supported in part by a research grant from Netflix. The authors thank Netflix for its support. This project was also supported by the Federal Ministry for Economic Affairs and Energy within the project "NXT GEN AI METHODS - Generative Methoden für Perzeption, Prädiktion und Planung", the project "GeniusRobot" (01IS24083) funded by the Federal Ministry of Research, Technology and Space (BMFTR), the bidt project KLIMA-MEMES, and the Horizon Europe project ELLIOT (GA No. 101214398). 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.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021. 5, 12

[2] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. ipoke: Poking a still image for controlled stochastic video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14707–14717, 2021. 3

[3] Chenjie Cao, Yuxin Hong, Xiang Li, Chengrong Wang, Chengming Xu, Yanwei Fu, and Xiangyang Xue. The image local autoregressive transformer. *Advances in Neural Information Processing Systems*, 34:18433–18445, 2021. 3

[4] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Proceedings of the 41st International Conference on Machine Learning*, pages 9550–9575. PMLR, 2024. 5, 12

[5] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Fine-grained open domain image animation with motion guidance. *arXiv preprint arXiv:2311.12886*, 2023. 2

[6] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023. 5

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 5

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 3

[9] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. 2021 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[10] Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens. *arXiv preprint arXiv:2410.13863*, 2024. 3

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

[12] Goldberger and Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Proceedings Ninth IEEE International conference on computer vision*, pages 487–493. IEEE, 2003. 5

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 2, 3

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

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 4, 12

[17] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for

matching across images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6207–6217, 2021. 3

[18] 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. 5, 7, 12

[19] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Puppet-master: Scaling interactive video generation as a motion prior for part-level dynamics. *arXiv preprint arXiv:2408.04631*, 2024. 3, 7, 8, 14

[20] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dragapart: Learning a part-level motion prior for articulated objects. In *Computer Vision – ECCV 2024*, pages 165–183, Cham, 2025. Springer Nature Switzerland. 3, 5, 7, 8, 14

[21] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *NeurIPS 2024*, 2024. 3

[22] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Yuexian Zou, and Ying Shan. Image conductor: Precision control for interactive video synthesis. *arXiv preprint arXiv:2406.15339*, 2024. 2

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

[24] Jingyun Liang, Yuchen Fan, Kai Zhang, Radu Timofte, Luc Van Gool, and Rakesh Ranjan. Movideo: Motion-aware video generation with diffusion models. In *European Conference on Computer Vision*, pages 0000–0000, 2024. 3

[25] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5, 12

[27] 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. *arXiv preprint arXiv:2407.02371*, 2024. 14

[28] OpenAI. Sora, 2024. 2

[29] 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. 5, 13

[30] Stephen E. Palmer. *Vision Science: Photons to Phenomenology*. The MIT Press, 1999. 1

[31] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2

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

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

[34] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 12

[35] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. *SIGGRAPH 2024*, 2024. 3, 7, 8, 13, 14

[36] Joonghyuk Shin, Daehyeon Choi, and Jaesik Park. Instantdrag: Improving interactivity in drag-based image editing. *arXiv preprint arXiv:2409.08857*, 2024. 2, 7, 13

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 3

[38] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: enhanced transformer with rotary position embedding. corr abs/2104.09864 (2021). *arXiv preprint arXiv:2104.09864*, 2021. 4, 5, 12

[39] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 4

[40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3, 7

[41] Michael Tschannen, Cian Eastwood, and Fabian Mentzer. Givt: Generative infinite-vocabulary transformers. In *European Conference on Computer Vision*, pages 292–309. Springer, 2024. 3, 4, 13

[42] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 3, 4

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

[44] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, 2021. 7

[45] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[46] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting

Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2025. 2

[47] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xin Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *ICCV*, 2025. 13

[48] Yuxi Xiao et al. Spatialtracker: Tracking any 2d pixels in 3d space. In *CVPR*, 2024. 13

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

[50] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023. 3

[51] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. Featured Certification. 3

[52] Jianhui Yu, Hao Zhu, Liming Jiang, Chen Change Loy, Weidong Cai, and Wayne Wu. Celebv-text: A large-scale facial text-video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14805–14814, 2023. 7

[53] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 12

[54] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 2

[55] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[56] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *European Conference on Computer Vision*, pages 388–406. Springer, 2025. 3

[57] 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. *arXiv preprint arXiv:2504.05579*, 2025. 5, 12

A. Implementation Details

The hyperparameters for the base model used for all evaluation and qualitative examples are reported in Tab. B. We train the WebVid model for a total of 800k steps with a learning rate of 5.0×10^{-5} using the AdamW [26] optimizer and a linear warmup of 5000 steps. The first 250k steps are trained with a batch size of 32; for the remainder, we set the batch size to 128. For our second model, keep the batch size constant and significantly increase warmup time. We also add a cosine LR decay and increase training time to 1M steps. Training time differences are due to an improved trainer setup from the first to the second model.

As described in Sec. 4.1, we randomly sample flow pokes and their corresponding positions from the given trajectories. We enforce that all flow values are in $[-1, 1]$ by applying a $\texttt{tanh}$ mapping and obtain a sinusoidal embedding for the x and y components of the flow. We then find the corresponding image features using the pokes positions and concatenate them with the flow features and local image features extracted by the image feature extractor $\mathcal{E}(\mathcal{I})$. Finally, we project them using a mapping network. These embeddings are then combined with query tokens. The query tokens represent the locations in the image for which we want to predict a flow distribution and are realized by a learnable token and the corresponding positional encoding.

The flow pokes and query tokens are fed to the transformer, which is 12 blocks deep and has a width of 768. The self-attention uses our *query-causal attention* mask as introduced in Sec. 3.2. We visualize it in Fig. A. In the cross-attention, the pokes and queries attend to the image features, enabling them to learn a global understanding of the scene. In both attention mechanisms, we use 2D Axial RoPE [4, 38] to model the spatial relationships of tokens. The FNNs expand the internal feature dimension by a factor of three, use SwiGLU [34] as an activation function, and are conditioned on whether or not the camera is static using AdaRMSNorms. On the output, we observe that directly using the output of the transformer for GMM parameter prediction sometimes produces unreasonable distributions and thus bad performance. Therefore, we use a simple MLP to project the transformer's output, alleviating that problem. For further details, we refer to our reference implementation[1], which contains extensive further comments in context.

Whether the camera is static or not is detected using a simple heuristic: we consider it to be static if a significant fraction of the scene's content is static. This information can be directly derived from the training tracks. Specifically, we find that considering a camera static once 40% of the frame move by at most 3px (at our training resolution of 448^2) works well on our training data. For the second training, we use 1% of the frame side length as the threshold instead.

[1] https://github.com/CompVis/flow-poke-transformer

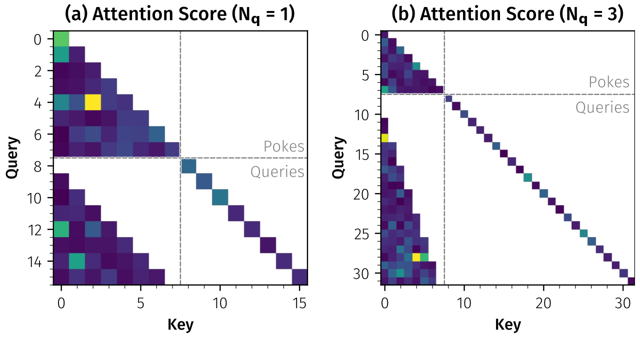


Figure A. **Query-Causal Attention Pattern Visualization**. We show the resulting attention patterns for our query-causal attention for different numbers of queries per poke count. We put poke tokens first, followed by query tokens. **(a)** In the simplest setting, with one query per set of pokes, there is one query token per set of pokes, with each query token attending to one more poke token than its predecessor. **(b)** For $N_q > 1$, the poke attention does not change, but there are multiple query tokens per poke set. Here, even query tokens for the same poke set do not attend to each other to enable parallel evaluation during inference.

Parameter	Value	
Dataset	WebVid [1] Subset	Open-Set Videos
Number of clips	3.8M	5M
Batch size	32→128	128
Optimizer	AdamW [26]	AdamW [26]
Peak learning rate	5.0×10^{-5}	5.0×10^{-5}
Learning rate schedule	constant	cosine decay to 10^{-8}
Betas	$(0.9, 0.99)$	$(0.9, 0.99)$
Warm-up steps	5k	100k
Total Steps	800k	1M
Precision	bfloat16	bfloat16
Total Parameters	230M	230M
GPUs	2 Nvidia H200	8 Nvidia H200
Training Time	7 days	1 day
Tracker	CoTracker3 [18]	TAPNext [57]
Tracker position seeding	48×48 grid	1024 random
Flow scale	$[-1, 1]$	$[-1, 1]$
Image size	448×448	448×448
Mixtures	4	4
Covariance	Full	Full
Given pokes	128	128
Query factor	15	15
Depth	12	12
SA width	768	768
CA width	768	768
Normalization	RMSNorm [53]	RMSNorm [53]
FFN expand factor	3	3
Activation	SwiGLU [34]	SwiGLU [34]
Positional encoding	2D Axial RoPE [4, 38]	2D Axial RoPE [4, 38]
Static scene conditioning	Adaptive Norm [16]	Adaptive Norm [16]

Table B. Hyperparameters for our main model across both datasets. Ablation models use the same parameters as the first one, but only train for 250k steps.

Method	Trained On	EPE@1 ↓	EPE@2 ↓	EPE@5 ↓	EPE@10 ↓	EPE@100 ↓
InstantDrag [36]	Faces	9.24	9.12	8.82	8.39	7.29
Motion-I2V [35]	**Generic (Zero-Shot)**	29.08	27.40	24.22	20.90	n/a
Ours (full training)	**Generic (Zero-Shot)**	**7.64**	**6.87**	**5.32**	**4.20**	**2.51**
Vision Feature Extractor Initialization						
Jointly Trained Pre-Trained (**Ours**)	**Generic (Zero-Shot)**	**8.08**	**6.96**	**5.38**	3.99	2.33
Frozen Pre-Trained	**Generic (Zero-Shot)**	8.30	7.22	5.44	**3.78**	**2.22**
Trained from Scratch	**Generic (Zero-Shot)**	8.15	7.51	6.14	4.73	2.57
GMM Component Count						
1 Component	**Generic (Zero-Shot)**	**7.60**	**6.87**	5.42	4.18	2.59
2 Components	**Generic (Zero-Shot)**	8.98	7.87	5.83	4.08	2.34
4 Components (**Ours**)	**Generic (Zero-Shot)**	8.08	6.96	**5.38**	3.99	2.33
8 Components	**Generic (Zero-Shot)**	8.23	7.19	5.57	4.01	**2.29**
16 Components	**Generic (Zero-Shot)**	8.41	7.26	5.44	**3.90**	2.34
GMM Covariance Parametrization						
Full Covariance, 4 Components (**Ours**)	**Generic (Zero-Shot)**	**8.08**	**6.96**	**5.38**	3.99	2.33
Diagonal, 4 Components	**Generic (Zero-Shot)**	8.13	7.09	5.40	**3.98**	**2.24**

Table A. Extension of Tab. 1 including our ablations. The experiment is identical to the original one. The ablation models have been trained for 250k steps compared to 800k for the full training. EPE@N refers to the endpoint error given N pokes.

B. Ablations

We show an extended version of Tab. 1 that includes quantitative results for hyperparameter ablations, which were used to motivate the choices mentioned in the main paper, in Tab. A. All comparison models follow the original training recipe but are only trained until 250k steps due to compute constraints.

First, we ablate whether to initialize the vision encoder with pre-trained weights and whether to continue training them. We find that initializing with pre-trained weights gives a performance boost in this setting. We hypothesize that, for very long trainings, both versions might end up performing similarly well. As noted in Sec. 4.1, freezing the feature extractor when initializing with DINOv2 [29] empirically results in a model with reduced instance segmentation capabilities compared to the unlocked version. We show a qualitative example of this in Fig. B. This behavior can also be observed in the quantitative evaluations, where, for low poke counts, the jointly trained model performs better than the one with a frozen feature extractor. At high poke counts, instance segmentation capabilities likely become less relevant, as the movement of all instances is likely already given explicitly via the conditioning.

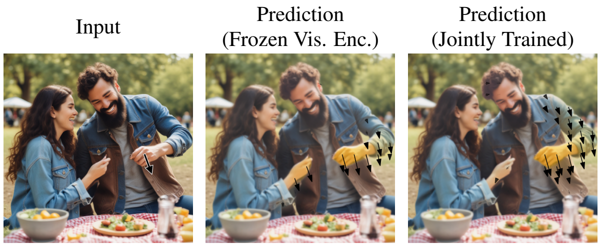


Figure B. Jointly training the vision encoder is important. We train a model with a frozen pretrained vision encoder. The model struggles with instance-specificity, predicting the same movement for the woman's hand as it does for the man's hand. When jointly training the vision encoder with the flow poke transformer, the man's hand's movement does not directly influence the woman's hand.

When ablating the number of GMM components, we find that adding too many components reduces the model's performance when conditioned on low numbers of given pokes. Only predicting a single component results in better quantitative performance at low given poke counts, but, obviously prevents the model from predicting multimodal distributions (see, e.g., Figs. 1, 3 and E), omitting a central property of our model. Parametrizing the covariance matrices as a pure diagonal matrix as done in GIVT [41] results in slightly reduced performance on average. Qualitatively, it also prevents angled distributions that our model successfully uses to express directional uncertainty (see, e.g., Fig. 1).

C. Extension to 3D Motion

In this paper, we evaluated the Flow Poke Transformer in the two-dimensional setting, meaning that the model only reasons in the image plane. However, the architecture itself is not limited to this setting and can trivially be extended to higher dimensions if desired. We show qualitative motion prediction results from such a version in Fig. C, where the model also successfully predicts reasonable out-of-plane motion in full 3D. This model was obtained by continued training from a 2D FPT checkpoint with 3D trackers obtained using SpatialTrackerV2 [47, 48] on a subset of Open-

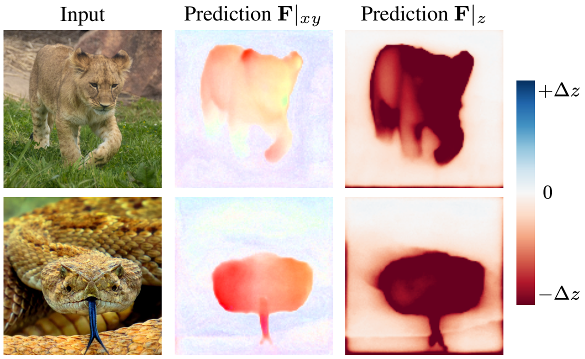


Figure C. **3D Motion Estimation**. We show unconditional 3D motion estimation samples from an FPT variant fine-tuned on 3D track data. The in-plane motion prediction $\mathbf{F}|_{xy}$ resembles that of a 2D FPT model, while this version can also successfully predict plausible out-of-plane motion $\mathbf{F}|_{z}$.

Vid1M [27]. The model is capable of predicting movement towards and away from the camera without any pokes, as shown by the near-static background and the clearly segmented motion of the animals in all three dimensions. When combining the predicted flow in Z-direction with a depth estimation of the input image, it is possible to tell which parts in the image will be occluded in the future.

D. Additional Qualitative Examples

For additional context for our quantitative results in Tab. 2, we show visualizations of some samples from that experiment in Fig. D. We also show additional qualitative examples for and pointwise motion predictions in Fig. E and samples for dense motion estimation in Fig. F.

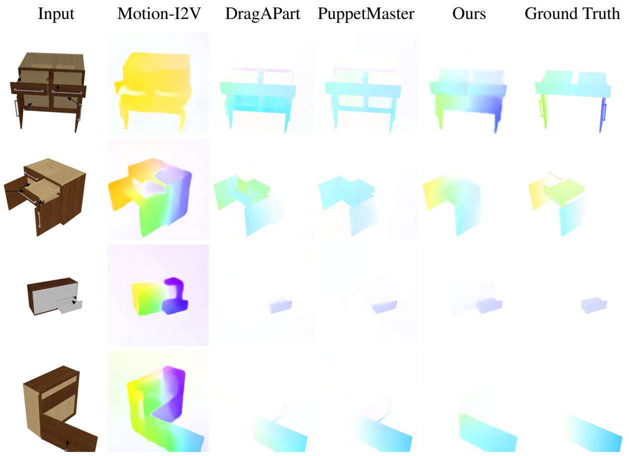
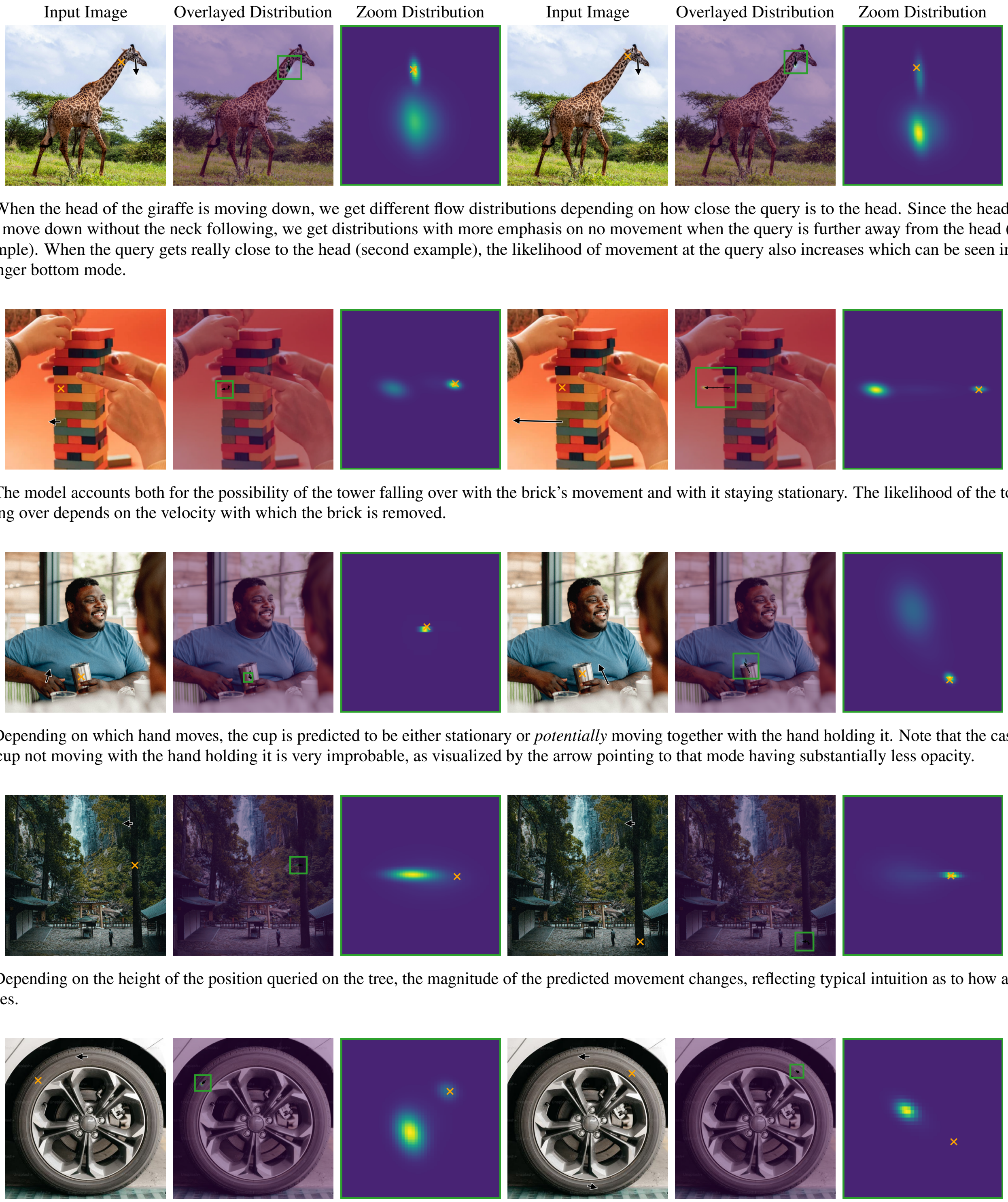


Figure D. **Qualitative Results for Articulated Object Motion Estimation**. We compare on Drag-A-Move [20] with Motion-I2V [35], DragAPart [20], and PuppetMaster [19]. Our model is qualitatively more capable of capturing complex conditioning with multiple different pokes than DAP and PM in this setup. Motion-I2V often fails to accurately follow the conditioning locally.



(a) When the head of the giraffe is moving down, we get different flow distributions depending on how close the query is to the head. Since the head can also move down without the neck following, we get distributions with more emphasis on no movement when the query is further away from the head (first example). When the query gets really close to the head (second example), the likelihood of movement at the query also increases which can be seen in the stronger bottom mode.

(b) The model accounts both for the possibility of the tower falling over with the brick's movement and with it staying stationary. The likelihood of the tower falling over depends on the velocity with which the brick is removed.

(c) Depending on which hand moves, the cup is predicted to be either stationary or *potentially* moving together with the hand holding it. Note that the case of the cup not moving with the hand holding it is very improbable, as visualized by the arrow pointing to that mode having substantially less opacity.

(d) Depending on the height of the position queried on the tree, the magnitude of the predicted movement changes, reflecting typical intuition as to how a tree moves.

(e) The model is capable of understanding the effect of rotational movements.

Figure E. Visualization of flow distribution for different pokes on the same image. The overlayed distribution visualizes the potential movement in the overall images, with the opacity of arrows denoting how likely each mode is (the more likely a mode, the less transparent the arrow).

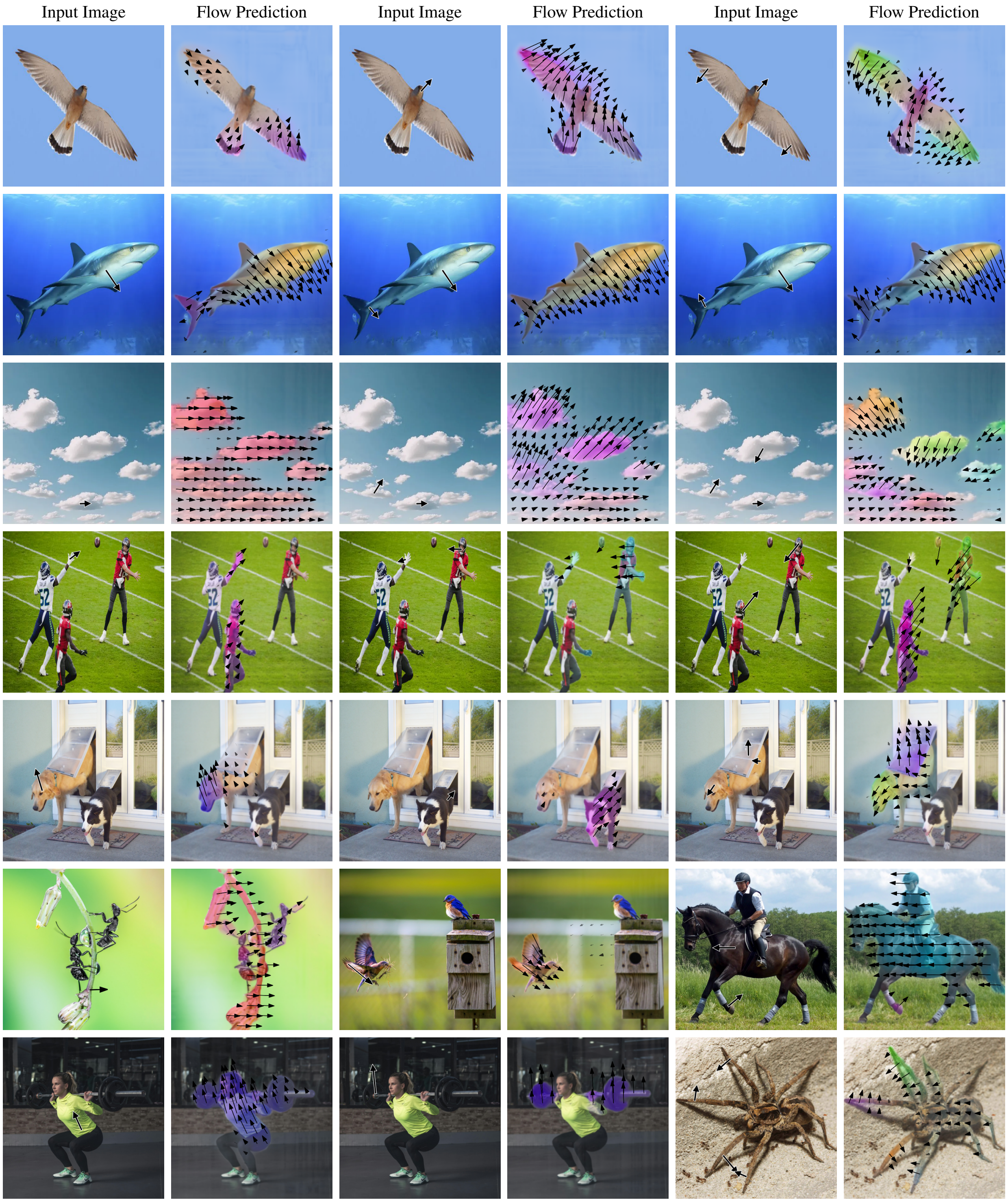


Figure F. Qualitative samples visualizing motion predictions inferred from a single image and (optionally) pokes.