POSEDIFF: A UNIFIED DIFFUSION MODEL BRIDGING ROBOT POSE ESTIMATION AND VIDEO-TO-ACTION CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

We present PoseDiff, a conditional diffusion model that unifies robot state estimation and control within a single framework. At its core, PoseDiff maps raw visual observations into structured robot states—such as 3D keypoints or joint angles—from a single RGB image, eliminating the need for multi-stage pipelines or auxiliary modalities. Building upon this foundation, PoseDiff extends naturally to video-to-action inverse dynamics: by conditioning on sparse video keyframes generated by world models, it produces smooth and continuous long-horizon action sequences through an overlap-averaging strategy. This unified design enables scalable and efficient integration of perception and control. On the DREAM dataset, PoseDiff achieves state-of-the-art accuracy and real-time performance for pose estimation. On Libero-Object manipulation tasks, it substantially improves success rates over existing inverse dynamics modules, even under strict offline settings. Together, these results show that PoseDiff provides a scalable, accurate, and efficient bridge between perception, planning, and control in embodied AI.

1 INTRODUCTION

Estimating robot pose from a single RGB image is fundamental in robotics, with applications in human-robot interaction (Yang et al., 2021; Christen et al., 2023; Zhu et al., 2023), multi-robot collaboration (Rizk et al., 2019; Papadimitriou et al., 2022; Li et al., 2022; Xun et al., 2023), hand-eye calibration (Taunyazov et al., 2020), and robot navigation (Rana et al., 2022; Bultmann et al., 2023). The task predicts 3D keypoints or joint angles from an RGB image (Figure 1a), which is challenging due to high degrees of freedom and variable poses. Existing multi-stage methods like RoboPose (Labbé et al., 2021), RoboPEPP (Goswami et al., 2025), HoRoPose (Ban et al., 2024), and RoboKeyGen (Tian et al., 2024b) rely on intermediate modalities, limiting end-to-end accuracy and speed.

Diffusion models have shown strong performance in robotics (Wang et al., 2024; Ren et al., 2024; Chandra et al., 2025; Lu et al., 2025; Chi et al., 2023), e.g., Diffusion Policy (Chi et al., 2023). Building on this, we propose **PoseDiff**, a conditional diffusion model that estimates 3D robot poses from a single RGB image using FiLM-modulated Conditional U-Net and DDPM denoising (Perez et al., 2018; Ho et al., 2020). On the DREAM dataset (Lee et al., 2020), **PoseDiff** achieves state-of-the-art accuracy and inference speed.

With embodied world models (Gao et al., 2024; Chi et al., 2024; Chi et al.; Rigter et al., 2024; Tian et al., 2024a), videos of action sequences are generated from RGB images and instructions but require inverse dynamics networks for execution, leading to high latency and sparse trajectories. Crucially, both pose estimation and inverse dynamics share the need to accurately extract robot states from visual observations—either as 3D poses from single images or as executable action sequences from video keyframes. This close connection motivates a unified approach.

Leveraging this insight, We extend **PoseDiff** to inverse dynamics by taking keyframe pairs as input and outputting continuous action sequences, using an overlap-averaging strategy (Figure 1b). On the Libero-Object dataset with off-the-shelf world models like AVDC (Ko et al., 2023), **PoseDiff** produces smooth, dense action sequences offline, outperforming baseline inverse dynamics methods and RoboPEPP adaptations.

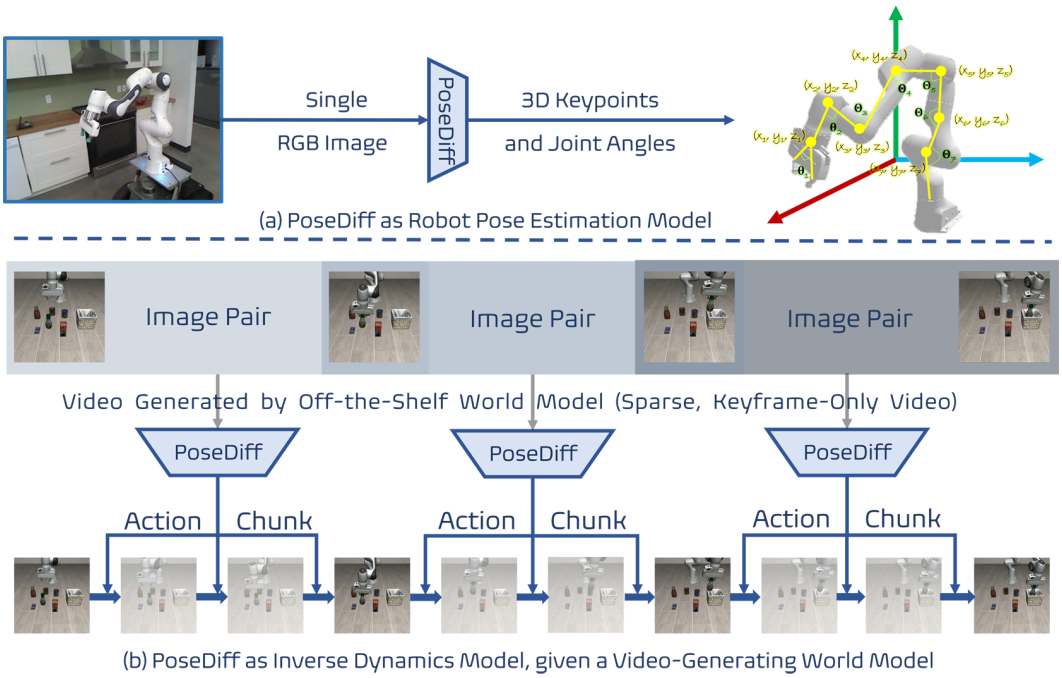

Figure 1: Overview of PoseDiff. (a) As a robot pose estimation model, PoseDiff predicts 3D keypoints or joint angles from a single RGB image. (b) As an inverse dynamics model, PoseDiff builds on the same robot-state-estimation capability to convert sparse keyframe pairs into dense, smooth action sequences for long-horizon robot control.

In summary, our contributions are:

- We propose **PoseDiff**, the first end-to-end diffusion model for robot pose estimation from a single RGB image, achieving SOTA accuracy and real-time speed.
- We are the first to exploit the intrinsic link between pose estimation and inverse dynamics, unifying them in a single framework that maps sparse keyframe videos to smooth long-horizon actions, substantially improving task success rates.

2 Related Work

2.1 Robot Pose Estimation

Early robot pose estimation relied on fiducial markers such as AuUco (Garrido-Jurado et al., 2014), AprilTag (Olson, 2011), and Artag (Fiala, 2005), using camera intrinsics and joint feedback for pose computation. Deep learning methods later improved accuracy and robustness: DREAM (Lee et al., 2020) combined CNNs with PnP, PoseFusion (Han et al., 2024) used multi-scale feature fusion, GSAM (Zhong et al., 2023) handled occlusions, CTRNet (Lu et al., 2023) leveraged self-supervised sim-to-real transfer, and SGTAPose (Tian et al., 2023) exploited temporal attention. However, most require real-time joint feedback, limiting sensorless applications.

Without joint angles, methods rely on rendering, task decomposition, or multi-stage pipelines. RoboPose (Labbé et al., 2021) is slow (1.8 FPS), SPDH (Simoni et al., 2022) needs depth maps, HoRoPose (Ban et al., 2024) and RoboKeyGen (Tian et al., 2024b) use multi-stage networks, and RoboPEPP (Goswami et al., 2025) employs a two-stage self-supervised framework. These approaches are computationally heavy and complex. PoseDiff overcomes these limitations with a single end-to-end diffusion model that estimates robot state directly from one RGB image—without joint priors, camera intrinsics, task decoupling, or auxiliary inputs—achieving higher accuracy and real-time inference.

2.2 Embodied World Model and Inverse Dynamics Model

World models predict future environment states from current observations and have shown strong adaptability. Early work focused on low-dimensional dynamics (Lesort et al., 2018; Ferns et al., 2004), latent-space planning (Nasiriany et al., 2019), and neural-network-based observation/action prediction (Finn & Levine, 2017). Modern model-based RL approaches (Hafner et al., 2020; Micheli et al., 2022) integrate dynamics and action generation in latent spaces for more effective planning. Video prediction is a key instantiation: Seer (Gu et al., 2023) generates future frames, while large-scale video models (Yang et al., 2023a; Valevski et al., 2024) extend this to interactive settings. In robotics, world models enable embodied planning and control, e.g., RoboDreamer (Zhou et al., 2024), UniSim (Yang et al., 2023b), EVA (Chi et al., 2024), and FLIP (Gao et al., 2024).

However, most existing works focus on video generation without translating frames into executable actions. Some add inverse dynamics models, e.g., UniPi (Du et al., 2023) maps frames to robot parameters, and FLIP (Gao et al., 2024) uses diffusion policies. These pipelines are iterative, slow, and produce sparse, discontinuous actions. Importantly, both pose estimation and inverse dynamics rely on extracting accurate robot states from visual inputs, yet prior methods treat them in isolation. **PoseDiff** is the first to exploit this connection, providing a fully offline inverse dynamics model: given sparse world-model-generated videos, it directly outputs dense, continuous action sequences executable on real robots, eliminating online inference and improving efficiency and stability.

3 PoseDiff as Robot Pose Estimation Model

3.1 Framework Comparison between PoseDiff and other Methods

As shown in Figure 2, PoseDiff has a much simpler architecture than existing robot pose estimation methods. Competing approaches rely on intermediate modalities: HoRoPose (Ban et al., 2024) predicts depth maps then converts them to 3D keypoints; RoboPEPP (Goswami et al., 2025) uses joint masks with an encoder-decoder; RoboKeyGen (Tian et al., 2024b) generates 2D keypoints then lifts them to 3D with a diffusion model.

In contrast, PoseDiff directly consumes an RGB image and outputs 3D keypoints end-to-end, avoiding modality conversion or auxiliary supervision. This reduces complexity and improves efficiency: RoboPEPP (Goswami et al., 2025) needs 23 ms per image, HoRoPose (Ban et al., 2024) 44 ms, and RoboKeyGen (Tian et al., 2024b) 54 ms, while PoseDiff achieves 14 ms, enabling real-time deployment.

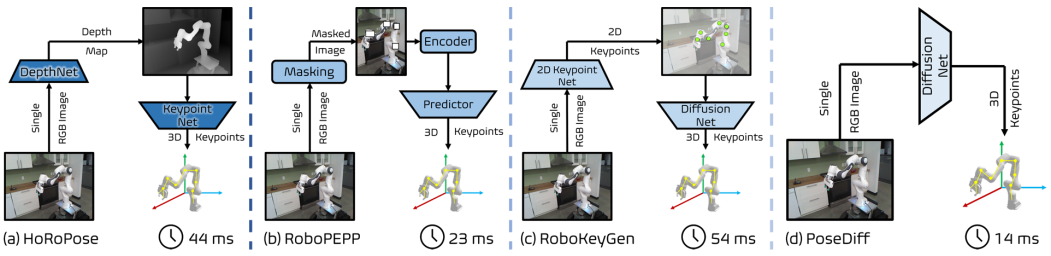

Figure 2: Framework comparison between PoseDiff and existing robot pose estimation methods. HoRoPose requires depth prediction followed by keypoint estimation (44 ms per image); RoboPEPP relies on joint masking and an encoder-decoder predictor (23 ms); RoboKeyGen first estimates 2D keypoints and then lifts them to 3D via a diffusion model (54 ms). In contrast, PoseDiff directly maps a single RGB image to 3D robot keypoints in an end-to-end manner, achieving faster inference (14 ms).

3.2 Diffusion Process

PoseDiff's core algorithm is a denoising diffusion probabilistic model (DDPM) that directly models the robot pose as a D-dimensional vector $\mathbf{x}_0 \in \mathbb{R}^D$. The model performs a forward (noising) process

that gradually corrupts $\mathbf{x}_0$ with Gaussian noise, followed by a learned reverse (denoising) process that recovers $\mathbf{x}_0$ from a noise sample.

Formally, the forward diffusion is defined as

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t;\ \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\big), \tag{1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, and β_t is a predefined noise schedule, and $\mathbf{I}$ denotes the D-dimensional identity matrix. This formulation allows sampling of $\mathbf{x}_t$ at any step t without iterating through all previous steps.

The reverse denoising process is parameterized as

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, c) = \mathcal{N}\big(\mathbf{x}_{t-1};\ \boldsymbol{\mu}_\theta(\mathbf{x}_t, t, c),\ \Sigma_\theta(t)\big), \tag{2}$$

where c denotes image features from a visual encoder. In practice, the network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)$ predicts the injected Gaussian noise. This prediction can be equivalently used to reconstruct the mean through

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t, c) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)\right), \tag{3}$$

so that $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)$ and $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t, c)$ are two equivalent parameterizations of the reverse process. Following the standard DDPM setting, we fix the variance as $\Sigma_\theta(t) = \beta_t\mathbf{I}$, determined by the noise schedule.

Therefore, PoseDiff is trained by minimizing

$$\mathcal{L}(\theta) \ = \ \mathbb{E}_{\mathbf{x}_0,\, t,\, \boldsymbol{\epsilon}}\Big[\ \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)\|_2^2\Big], \tag{4}$$

where $\mathbf{x}_t$ is sampled from the forward process in Eq. 1, and $\boldsymbol{\epsilon}$ denotes the Gaussian noise added during this process. This loss encourages accurate prediction of the injected noise at each timestep, thereby enabling the model to denoise robot poses conditioned on the visual context.

3.3 Model Design

PoseDiff adopts a Conditional U-Net architecture as the core noise predictor. The network is responsible for estimating the noise at each denoising step of the diffusion process, conditioned on both the current noisy state and external information. The overall architecture is illustrated in Figure 3.

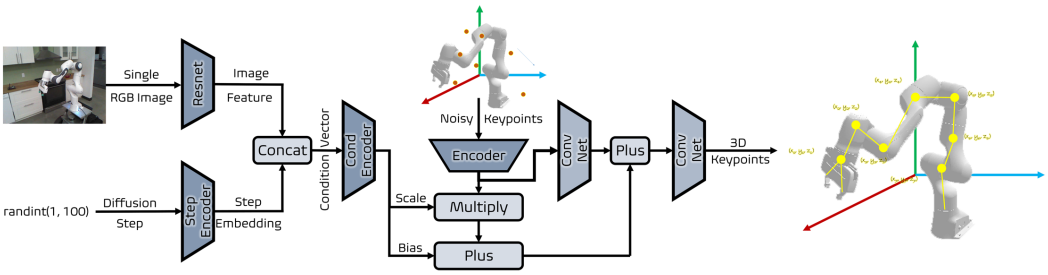

Figure 3: PoseDiff architecture. Visual features from a ResNet and timestep embeddings are fused via a condition encoder with FiLM modulation, guiding the denoising of noisy 3D keypoints into clean pose estimates.

Specifically, a single RGB image is first passed through a ResNet encoder to extract an image feature representation. During training, a randomly sampled diffusion timestep t is separately embedded via a step encoder, producing a timestep embedding. The image feature and timestep embedding are concatenated to form a condition vector, which is then fed into a dedicated condition encoder. To ensure that this condition vector effectively guides the prediction of the robot pose, we adopt Feature-wise Linear Modulation (FiLM). The condition encoder outputs two modulation vectors, denoted as $\mathbf{s}$ (scale) and $\mathbf{b}$ (bias).

In parallel, the noisy 3D keypoint coordinates are input to another encoder that produces a keypoint feature $\mathbf{f}_{\text{kp}}$. FiLM modulation is then applied to $\mathbf{f}_{\text{kp}}$ using the outputs of the condition encoder:

$$\mathbf{f}_{\text{FiLM}} = \mathbf{s} \odot \mathbf{f}_{\text{kp}} + \mathbf{b}, \tag{5}$$

where $\odot$ denotes element-wise multiplication. To further enhance representational capacity, we incorporate a residual module: the original keypoint feature is processed by a convolutional layer and then added to $\mathbf{f}_{\text{FiLM}}$. Finally, the combined feature is passed through a convolutional network to predict the denoised 3D keypoints.

For clarity, Figure 3 only depicts the core structure of the model. In practice, the concatenation branch after the condition encoder extends into multiple layers of the Conditional U-Net, which jointly form a full encoder-decoder backbone. This hierarchical design allows multi-scale integration of visual context and diffusion timesteps, enabling robust noise prediction and effective robot pose estimation.

4 PoseDiff as Inverse Dynamics Model Built on Pose Estimation

Figure 4 illustrates how PoseDiff operates as an inverse dynamics model in collaboration with an embodied world model to accomplish tasks. Given a third-person *init RGB image*, which reflects both the initial robot state and the initial state of the manipulated objects, a textual instruction describing the target task is sent to the world model. Based on this instruction, the world model generates a complete video that visually depicts a successful execution of the task.

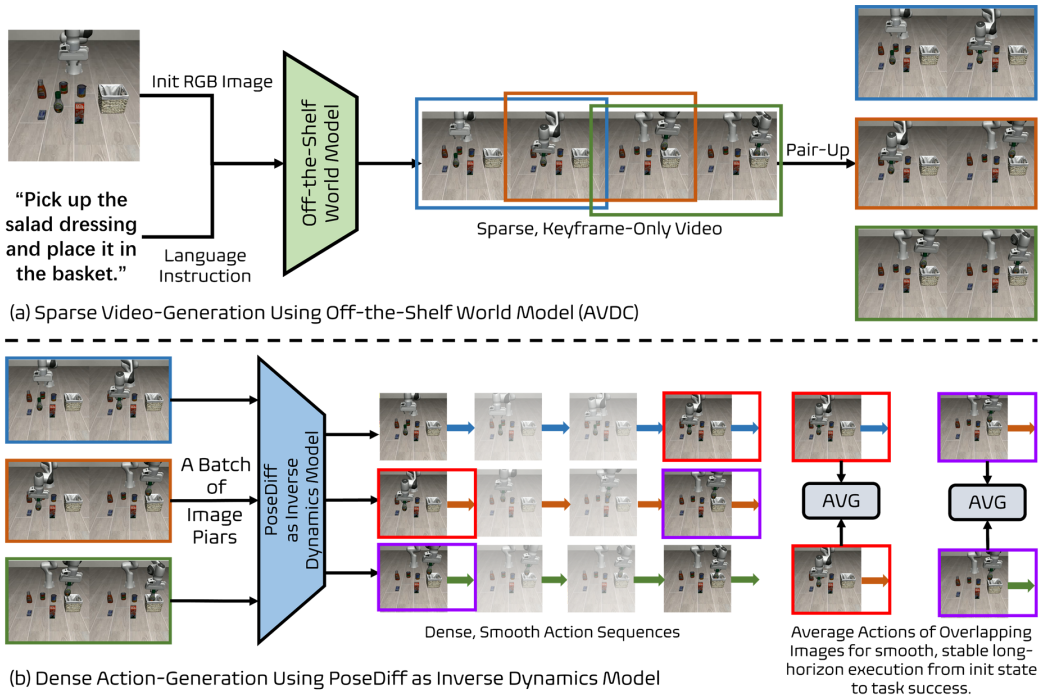

Figure 4: PoseDiff as an inverse dynamics model: (a) the world model generates sparse video frames from an initial image and language instruction; (b) PoseDiff fills in dense actions between frame pairs, averaging overlaps for smooth and consistent trajectories.

It is important to note that such generated videos are typically temporally sparse: frames correspond to high-level steps of the task, but cannot be directly translated into smooth and executable robot trajectories. Therefore, an inverse dynamics model is required to interpolate dense action sequences between consecutive frames. To address this, we adopt a sliding window strategy: the generated video frames are paired up using a window size of two frames and a stride of one. Each such frame pair is then batched and fed into a modified version of PoseDiff.

Specifically, PoseDiff is adapted in two ways: (i) instead of taking a single image as input, it now accepts an image pair; and (ii) instead of predicting a single robot pose, it outputs a dense sequence of N continuous actions. Here, N is a hyperparameter that ensures the resulting trajectory is smooth, temporally consistent, and physically executable.

Since the second frame of a given pair becomes the first frame of the subsequent pair, the last predicted action from one pair and the first predicted action from the next pair correspond to the same action step. However, because these predictions arise from different input pairs, they may differ slightly. To resolve this, we average the overlapping actions across pairs, thereby ensuring that the final action sequence across the entire episode is complete, stable, and coherent.

The pseudocode of this procedure is given below.

Algorithm 1 PoseDiff as Inverse Dynamics Model

Input: Generated video frames $\{I_1, I_2, \ldots, I_T\}$, hyperparameter N
Output: Executable action sequence $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$
Initialize empty action sequence $\mathcal{A} \leftarrow [\,]$
for $t = 1$ to $T - 1$ **do**
    Form frame pair (I_t, I_{t+1})
    Predict dense actions $\{a_t^1, a_t^2, \ldots, a_t^N\} \leftarrow$ PoseDiff(I_t, I_{t+1})
    if last action a_t^N overlaps with first action a_{t+1}^1 **then**
        Replace with their average
    end if
    Append actions $\{a_t^1, a_t^2, \ldots, a_t^N\}$ to $\mathcal{A}$
end for
return $\mathcal{A}$

5 Experiments

5.1 Robot Pose Estimation

5.1.1 Datasets and Evaluation Metrics

For the robot pose estimation task, following prior works in this domain, we adopt the DREAM-REAL dataset (Lee et al., 2020) as the evaluation benchmark. DREAM-REAL (Lee et al., 2020) consists of four subsets: AK, XK, RS, and ORB. Each subset contains approximately 5,000 images captured from different viewpoints of real-world scenes, where robots perform diverse actions. For each subset, we randomly split the data into 80% for training and 20% for testing.

We employ three widely used evaluation metrics:

(1) Average Distance (ADD). This metric computes the average Euclidean distance between the predicted 3D keypoints and the ground-truth 3D keypoints. Formally,

$$\mathrm{ADD} = \frac{1}{N} \sum_{i=1}^{N} \|p_i - g_i\|_2, \tag{6}$$

where p_i denotes the predicted 3D coordinates of the i-th keypoint, g_i denotes the corresponding ground-truth coordinate, and N is the total number of keypoints.

(2) Area Under the Curve (AUC). For each test sample j, we first compute its ADD value d_j according to Eq. 6. Given a set of thresholds $\{t_k\}$ in the range $[0, 100]$ mm, the accuracy at threshold t_k is defined as

$$\mathrm{Acc}(t_k) = \frac{1}{M} \sum_{j=1}^{M} \mathbb{I}(d_j < t_k), \tag{7}$$

where M is the number of test samples and $\mathbb{I}(\cdot)$ is the indicator function. The AUC value is then obtained by integrating the accuracy curve:

$$\mathrm{AUC} = \frac{1}{T}\int_0^T \mathrm{Acc}(t)\,\mathrm{d}t, \tag{8}$$

where $T = 100$ mm is the maximum threshold. This corresponds to the normalized area under the accuracy-threshold curve.

(3) Frames Per Second (FPS). We further measure inference efficiency in terms of FPS, i.e., the number of images the model can process per second, including the entire pipeline from input image to estimated robot state. Specifically, if the model processes F images in total time T seconds, FPS is given by

$$\mathrm{FPS} = \frac{F}{T}. \tag{9}$$

5.1.2 Quantitative Results

We compare PoseDiff with two major categories of existing robot pose estimation methods on the task of estimating 3D robot keypoints: (1) methods that require access to ground-truth joint angle information, and (2) methods that do not rely on any known joint angles. The first kind of methods include DREAM-F (Lee et al., 2020), DREAM-Q (Lee et al., 2020), DREAM-H (Lee et al., 2020), RoboPose* (Labbé et al., 2021), CtRNet (Lu et al., 2023), SGTAPose (Tian et al., 2023), and HoRoPose* (Ban et al., 2024), where DREAM-F, DREAM-H, and DREAM-Q denote decoder outputs at full (F), half (H), and quarter (Q) resolution, respectively. The methods that do not rely on ground truth joint angles include RoboPose (Labbé et al., 2021), HoRoPose (Ban et al., 2024), and RoboPEPP (Goswami et al., 2025).

Table 1: Comparison of ADD results (measured in millimeters) on four datasets. Methods are divided into two categories: those requiring ground-truth joint angle information (top) and those without joint angle supervision (bottom). **PoseDiff** achieves state-of-the-art performance across all datasets.

Method	Known Joint Angles	AK	XK	RS	ORB
DREAM-F	Yes	11413.1	491911.4	2077.4	95319.1
DREAM-Q	Yes	78089.3	54178.2	27.2	64247.6
DREAM-H	Yes	56.5	7381.6	23.6	25685.3
RoboPose*	Yes	24.2	14.0	23.1	19.4
CtRNet	Yes	13.0	32.0	10.0	21.0
SGTAPose	Yes	35.7	157.7	12.8	35.0
HoRoPose*	Yes	12.9	19.9	11.0	15.9
RoboPose	No	34.3	22.3	26.0	30.1
HoRoPose	No	18.9	24.0	24.8	24.8
RoboPEPP	No	29.0	22.0	23.0	27.0
Posediff(Ours)	**No**	**3.6**	**5.2**	**3.4**	**3.5**

Table 1 reports the comparison results in terms of the ADD metric, measured in millimeters. A smaller ADD indicates that the estimated 3D robot keypoints are closer to the ground truth, corresponding to higher accuracy. Our method consistently achieves millimeter-level accuracy across all four datasets, outperforming even those approaches that rely on joint angle information. Specifically, on the AK, XK, RS, and ORB datasets, PoseDiff achieves ADD values of 3.6, 5.2, 3.4, and 3.5, respectively, all of which establish new state-of-the-art performance. Compared with the best existing methods without joint angle supervision, PoseDiff reduces the error by 15.3, 16.8, 19.6, and 21.3 millimeters, representing a substantial improvement in precision. Moreover, even the strongest methods leveraging ground-truth joint angles achieve no better than 10.0 mm in ADD, whereas PoseDiff consistently remains below 5.5 mm across all datasets.

The primary source of this advantage lies in the conditional diffusion model introduced in PoseDiff. This model provides powerful capacity for high-dimensional probabilistic modeling, enabling it to

effectively learn the complex mapping from image space to robot pose space. In contrast, existing approaches often rely on multi-stage pipelines or explicitly designed structures, which limit their representational power, hinder generalization, and lead to fluctuating accuracy.

The visualization results can be found in Appendix A.

Table 2: Comparison of AUC (%) across four benchmark datasets. Methods are grouped by whether ground-truth joint angles are available. PoseDiff, without requiring joint-angle annotations, consistently outperforms all prior approaches and even surpasses methods that rely on joint-angle supervision.

Method	Known Joint Angles	AK	XK	RS	ORB
DREAM-F	Yes	68.9	24.4	76.1	61.9
DREAM-Q	Yes	52.4	37.5	78.0	57.1
DREAM-H	Yes	60.5	64.0	78.8	69.1
RoboPose*	Yes	76.5	86.0	76.9	80.5
CtRNet	Yes	89.9	79.5	90.8	85.3
SGTAPose	Yes	67.8	2.1	87.6	72.3
HoRoPose*	Yes	90.2	81.2	91.9	87.6
RoboPose	No	70.4	77.6	74.3	70.4
HoRoPose	No	82.2	76.0	75.2	75.2
RoboPEPP	No	75.3	78.5	80.5	77.5
Posediff(Ours)	**No**	**96.4**	**94.8**	**96.6**	**96.5**

The comparison of PoseDiff with existing approaches in terms of AUC is reported in Table 2. A higher AUC indicates more accurate pose prediction. Across all four datasets, our method consistently achieves AUC values exceeding 94.0, significantly outperforming prior baselines. Specifically, PoseDiff attains AUC scores of 96.4, 94.8, 96.6, and 96.5 on the AK, XK, RS, and ORB datasets, respectively. Relative to the strongest prior method that does not rely on joint-angle annotations, PoseDiff improves performance by 14.2, 16.3, 16.1, and 19.0 points. Remarkably, PoseDiff even surpasses methods that leverage ground-truth joint angles, highlighting the robustness and effectiveness of our unified diffusion-based framework.

Table 3: Comparison of inference speed (FPS) on the AK dataset. **PoseDiff** achieves the highest real-time performance among all evaluated methods.

	DREAM-F	RoboPose	HoRoPose	RoboPEPP	**Posediff**
FPS on AK	15.1	1.8	22.6	43.5	**73.5**

In addition to evaluating accuracy and prediction performance, we also compared the inference speed of PoseDiff against existing approaches on the AK dataset. Unlike prior methods that decompose the task into multiple sub-modules or rely on multi-stage predictions, PoseDiff directly maps input images to robot poses in an end-to-end manner. This design yields a substantial advantage in efficiency: PoseDiff achieves an inference speed of 73.5 FPS, which is 30 FPS faster than the best-performing baseline. Such performance comfortably satisfies the requirements for real-time deployment. The detailed results are reported in Table 3.

5.2 Inverse Dynamics Model

5.2.1 Dataset and Evaluation Metrics

For the inverse dynamics modeling task, we adopt a relatively aggressive experimental setup. Specifically, we select five distinct tasks from the Libero-Object dataset. For each task, we slightly modify the AVDC world model (Ko et al., 2023) to increase the density of generated video frames. Subsequently, videos are generated from 50 different initial states per task.

PoseDiff and the baseline methods are first trained on the original dataset. The generated videos are then fed into the models in the form of image pairs to obtain the complete action sequence for

each episode. This sequence is executed in the simulator to evaluate the success rate. It is important to emphasize that the success rate is measured conditioned on the world model generating videos that successfully and plausibly reflect task execution. Videos that fail to depict the task correctly, or contain ambiguous or physically inconsistent dynamics, are excluded from this evaluation, as our focus is on assessing the inverse dynamics model rather than the world model.

Formally, let $\{\mathbf{V}_i\}_{i=1}^N$ denote the set of generated videos, and $\{\mathbf{a}_i\}_{i=1}^N$ the corresponding predicted action sequences. Let $\mathcal{S} = \{i \mid \text{VideoValid}(\mathbf{V}_i)\}$ be the subset of indices corresponding to videos that plausibly reflect successful task execution. Each $\mathbf{a}_i$ with $i \in \mathcal{S}$ is executed in the simulator to produce a trajectory τ_i. The conditional success rate of the inverse dynamics model is then defined as:

$$\text{Success Rate} = P\big(\text{TaskSuccess}(\tau_i) \mid \text{VideoValid}(\mathbf{V}_i)\big) = \frac{\sum_{i\in\mathcal{S}} \mathbb{I}\big[\text{TaskSuccess}(\tau_i)\big]}{|\mathcal{S}|}, \tag{10}$$

where $\mathbb{I}[\cdot]$ is the indicator function, and $\text{TaskSuccess}(\tau_i)$ evaluates whether the trajectory τ_i achieves the task successfully in the simulator.

5.2.2 Quantitative Results

Table 4 reports quantitative comparisons on the Libero-Object dataset. We evaluate five representative tasks, each involving manipulation of a different object, denoted as *Soup, Cheese, Salad, Ketchup*, and *Tomato*. As baselines, we consider two inverse dynamics models: (i) the inverse dynamics module from Seer (Tian et al., 2024a), and (ii) a modified variant of RoboPEPP (Goswami et al., 2025), adapted by reformatting its inputs and outputs for inverse dynamics prediction. We then compare these against our proposed PoseDiff.

Table 4: Success rates of different inverse dynamics models on five Libero-Object tasks. PoseDiff significantly outperforms Seer and RoboPEPP, achieving robust performance across all object manipulation tasks.

Method	Soup	Cheese	Salad	Ketchup	Tomato
Seer	0%	0%	0%	0%	0%
RoboPEPP	0%	0%	0%	0%	0%
PoseDiff	**60%**	**58%**	**68%**	**62%**	**62%**

As expected, since all methods are executed entirely offline without access to real-time environment feedback, even small per-step prediction errors accumulate over time and ultimately lead to task failures. Consequently, both Seer and the adapted RoboPEPP achieve a 0% success rate across all tasks. In contrast, PoseDiff demonstrates substantially higher robustness, achieving 60%, 58%, 68%, 62%, and 62% success rates on the five tasks, respectively. These results highlight that PoseDiff, when employed as an inverse dynamics model, attains strong accuracy not only in robot pose estimation but also in downstream action execution. This finding supports our central claim: a model with sufficiently precise robot pose estimation can be effectively repurposed as an inverse dynamics model for world models, enabling purely offline inference while significantly improving task efficiency.

The visualization results can be found in Appendix A.

6 Conclusion

We introduced **PoseDiff**, a unified diffusion model that bridges robot pose estimation and video-to-action control by exploiting their intrinsic connection. Viewing both as visual-to-state generation, PoseDiff achieves SOTA pose accuracy and real-time efficiency on DREAM datasets, and substantially improves task success rates for inverse dynamics on Libero-Object datasets. The framework streamlines complexity while offering a scalable basis for linking visual perception to robotic control in embodied AI.

REFERENCES

Shikun Ban, Juling Fan, Xiaoxuan Ma, Wentao Zhu, Yu Qiao, and Yizhou Wang. Real-time holistic robot pose estimation with unknown states. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

Simon Bultmann, Raphael Memmesheimer, and Sven Behnke. External camera-based mobile robot pose estimation for collaborative perception with smart edge sensors. *arXiv preprint arXiv:2303.03797*, 2023.

Akshay L Chandra, Iman Nematollahi, Chenguang Huang, Tim Welschehold, Wolfram Burgard, and Abhinav Valada. Diwa: Diffusion policy adaptation with world models. *arXiv preprint arXiv:2508.03645*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Xiaowei Chi, Chun-Kai Fan, Hengyuan Zhang, Xingqun Qi, Rongyu Zhang, Anthony Chen, Chi-Min Chan, Wei Xue, Qifeng Liu, Shanghang Zhang, et al. Empowering world models with reflection for embodied video prediction. In *Forty-second International Conference on Machine Learning*.

Xiaowei Chi, Chun-Kai Fan, Hengyuan Zhang, Xingqun Qi, Rongyu Zhang, Anthony Chen, Chi-min Chan, Wei Xue, Qifeng Liu, Shanghang Zhang, et al. Eva: An embodied world model for future video anticipation. *arXiv preprint arXiv:2410.15461*, 2024.

Sammy Christen, Wei Yang, Claudia Pérez-D'Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning human-to-robot handovers from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9654–9664, 2023.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pp. 162–169, 2004.

Mark Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 590–596. IEEE, 2005.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. *arXiv preprint arXiv:2412.08261*, 2024.

Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

Raktim Gautam Goswami, Prashanth Krishnamurthy, Yann LeCun, and Farshad Khorrami. Robopepp: Vision-based robot pose and joint angle estimation through embedding predictive pre-training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6930–6939, 2025.

Xianfan Gu, Chuan Wen, Weirui Ye, Jiaming Song, and Yang Gao. Seer: Language instructed video prediction with latent diffusion models. *arXiv preprint arXiv:2303.14897*, 2023.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Xujun Han, Shaochen Wang, Xiucai Huang, and Zhen Kan. Posefusion: Multi-scale keypoint correspondence for monocular camera-to-robot pose estimation in robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 795–801. IEEE, 2024.

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

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023.

Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Single-view robot pose and joint angle estimation via render & compare. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1654–1663, 2021.

Timothy E Lee, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Oliver Kroemer, Dieter Fox, and Stan Birchfield. Camera-to-robot pose estimation from a single image. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9426–9432. IEEE, 2020.

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.

Shushuai Li, Christophe De Wagter, and Guido CHE De Croon. Self-supervised monocular multi-robot relative localization with efficient deep neural networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 9689–9695. IEEE, 2022.

Haofei Lu, Yifei Shen, Dongsheng Li, Junliang Xing, and Dongqi Han. Habitizing diffusion planning for efficient and effective decision making. *arXiv preprint arXiv:2502.06401*, 2025.

Jingpei Lu, Florian Richter, and Michael C Yip. Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21296–21306, 2023.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. *Advances in neural information processing systems*, 32, 2019.

Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*, pp. 3400–3407. IEEE, 2011.

Andreas Papadimitriou, Sina Sharif Mansouri, and George Nikolakopoulos. Range-aided egocentric collaborative pose estimation for multiple robots. *Expert Systems with Applications*, 202: 117052, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Arianna Rana, Fabio Vulpi, Rocco Galati, Annalisa Milella, and Antonio Petitti. A pose estimation algorithm for agricultural mobile robots using an rgb-d camera. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1–5. IEEE, 2022.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. Avid: Adapting video diffusion models to world models. *arXiv preprint arXiv:2410.12822*, 2024.

Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–31, 2019.

Alessandro Simoni, Stefano Pini, Guido Borghi, and Roberto Vezzani. Semi-perspective decoupled heatmaps for 3d robot pose estimation from depth maps. *IEEE Robotics and Automation Letters*, 7(4):11569–11576, 2022.

Tasbolat Taunyazov, Weicong Sng, Hian Hian See, Brian Lim, Jethro Kuan, Abdul Fatir Ansari, Benjamin CK Tee, and Harold Soh. Event-driven visual-tactile sensing and learning for robots. *arXiv preprint arXiv:2009.07083*, 2020.

Yang Tian, Jiyao Zhang, Zekai Yin, and Hao Dong. Robot structure prior guided temporal attention for camera-to-robot pose estimation from image sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8917–8926, 2023.

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024a.

Yang Tian, Jiyao Zhang, Guowei Huang, Bin Wang, Ping Wang, Jiangmiao Pang, and Hao Dong. Robokeygen: robot pose and joint angles estimation via diffusion-based 3d keypoint generation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5375–5381. IEEE, 2024b.

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

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. *arXiv preprint arXiv:2407.01812*, 2024.

Zhiren Xun, Jian Huang, Zhehan Li, Chao Xu, Fei Gao, and Yanjun Cao. Crepes: Cooperative relative pose estimation towards real-world multi-robot systems. *CoRR*, 2023.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023a.

Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Reactive human-to-robot handovers of arbitrary objects. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3118–3124. IEEE, 2021.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023b.

Xiaopin Zhong, Wenxuan Zhu, Weixiang Liu, Jianye Yi, Chengxiang Liu, and Zongze Wu. G-sam: A robust one-shot keypoint detection framework for pnp based robot pose estimation. *Journal of Intelligent & Robotic Systems*, 109(2):28, 2023.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human motion generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2430–2449, 2023.

A APPENDIX

This appendix presents visualization results that compare PoseDiff, evaluated both as a robot pose estimation model and as an inverse dynamics model, against baseline methods.

A.1 ROBOT POSE ESTIMATION

Figure 5 provides a qualitative comparison between PoseDiff and RoboPEPP (Goswami et al., 2025) on the DREAM-XK dataset. As shown, PoseDiff accurately reconstructs the robot states from the RGB images, whereas RoboPEPP exhibits noticeably larger errors.

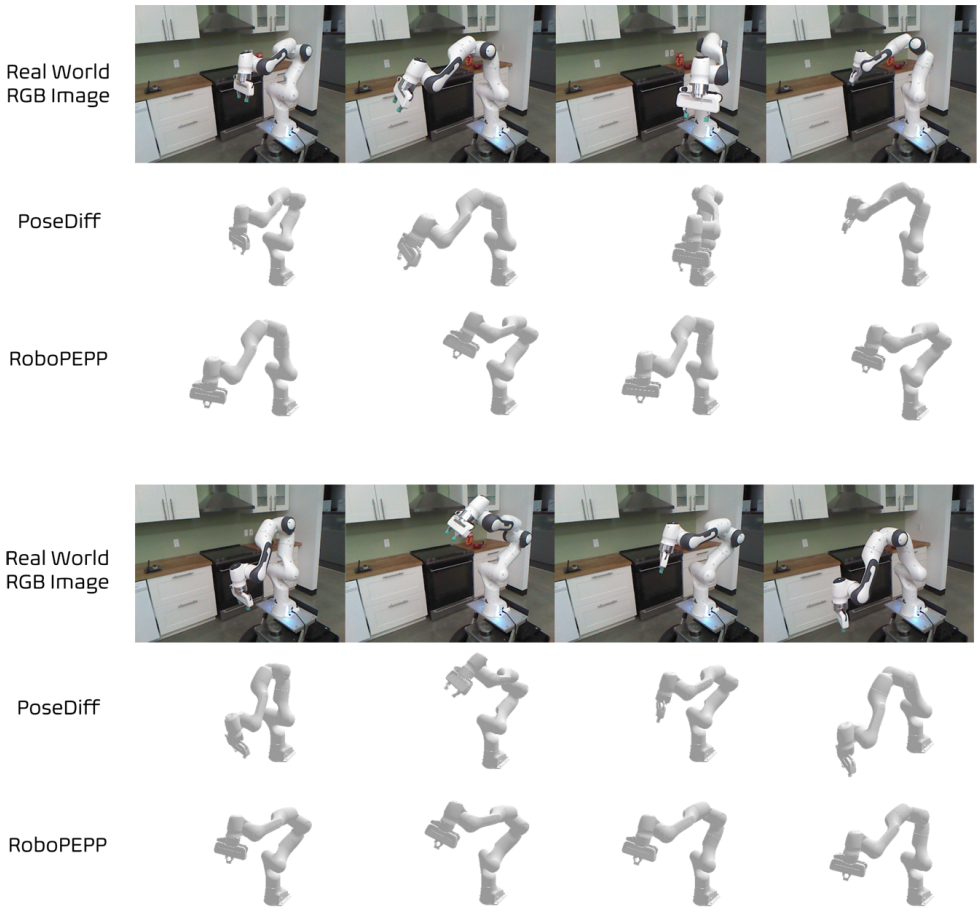

Figure 5: Qualitative comparison between PoseDiff and RoboPEPP on the DREAM-XK dataset. The top row shows input RGB observations, the middle rows depict the predicted robot poses by PoseDiff, and the bottom row illustrates that by RoboPEPP. PoseDiff produces reconstructions that closely match the ground-truth robot states, while RoboPEPP exhibits noticeably larger deviations.

A.2 INVERSE DYNAMICS MODEL

Figures 6 and 7 illustrate the effectiveness of PoseDiff as an inverse dynamics model for world models. As a concrete example, we use the Libero-Object task "Pick up the alphabet soup and place it in the basket". The first row shows video frames generated by the AVDC world model. In the second row, we execute actions predicted by a modified RoboPEPP (with adapted input-output) when used as an inverse dynamics model. The third row corresponds to the inverse dynamics model in Seer, and the fourth row presents the results of PoseDiff.

From the comparison, we observe that when provided with sparse offline video frames, both RoboPEPP and Seer fail to generate action sequences that faithfully capture the robot's motion as depicted in the world model, often producing significant deviations. In contrast, PoseDiff is able to accurately reconstruct the actions shown in the video, closely aligning execution with the world model's behavior. Moreover, the action sequences generated by PoseDiff successfully complete the task.

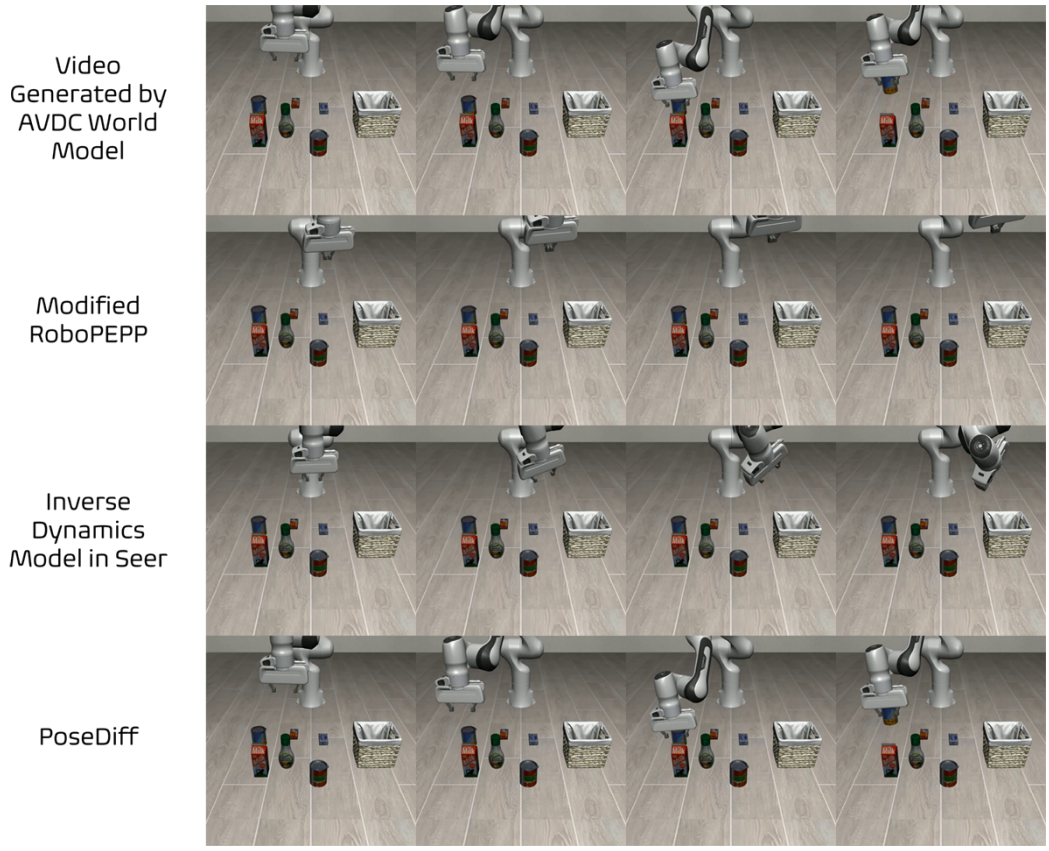

Figure 6: Results on the Libero-Object task "Pick up the alphabet soup and place it in the basket." Rows show AVDC world model Video(1st), RoboPEPP (2nd), Seer (3rd), and PoseDiff (4th). Only PoseDiff reconstructs the action sequence accurately and completes the task.

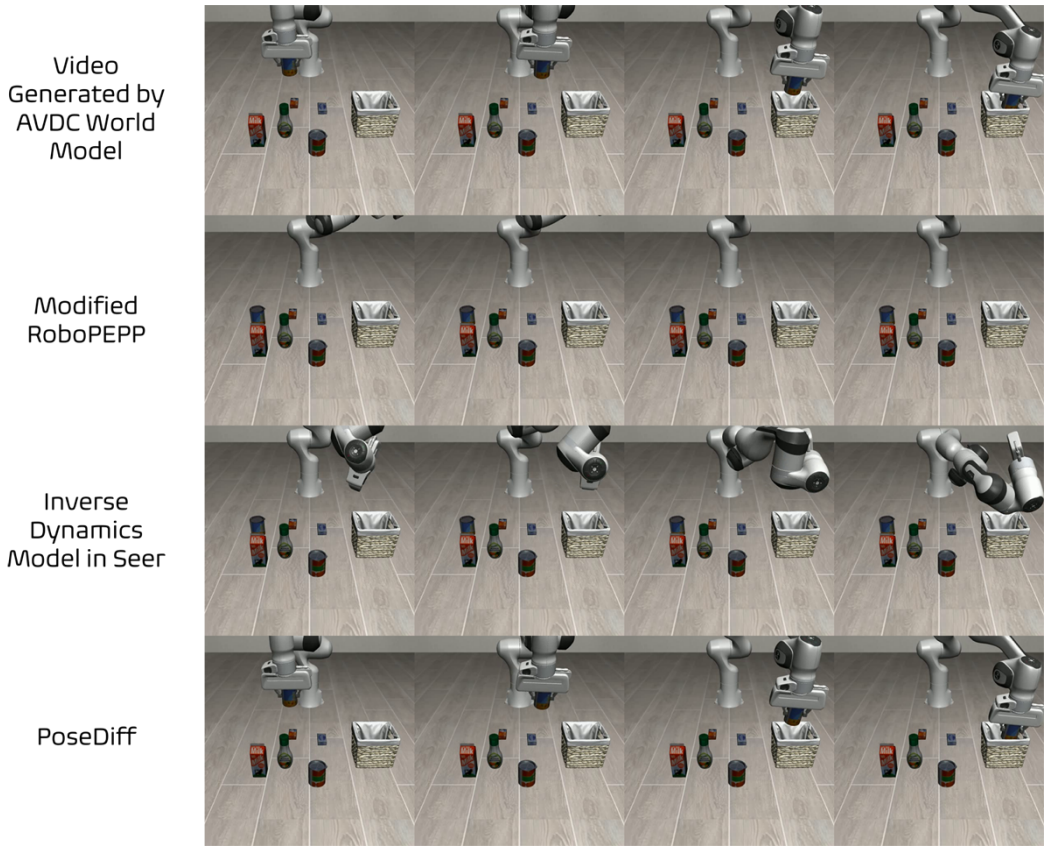

Figure 7: Results on the Libero-Object task "Pick up the alphabet soup and place it in the basket." Rows show AVDC world model Video(1st), RoboPEPP (2nd), Seer (3rd), and PoseDiff (4th). Only PoseDiff reconstructs the action sequence accurately and completes the task.

B THE USE OF LARGE LANGUAGE MODELS (LLMS)

During the preparation of this manuscript, large language models (LLMs) were used solely for correcting grammar, checking spelling, and improving the fluency of sentences. They did not contribute to idea generation, data analysis, experimental design, or drawing conclusions. All intellectual input—including the development of research questions, methodology, results, and interpretations—remains entirely the work of the authors. The function of LLMs was strictly confined to linguistic polishing to enhance clarity and readability.