

PIXEL MOTION AS UNIVERSAL REPRESENTATION FOR ROBOT CONTROL

Kanchana Ranasinghe, Xiang Li, E-Ro Nguyen, Cristina Mata, Jongwoo Park,
Michael S Ryoo

Stony Brook University

kranasinghe@cs.stonybrook.edu

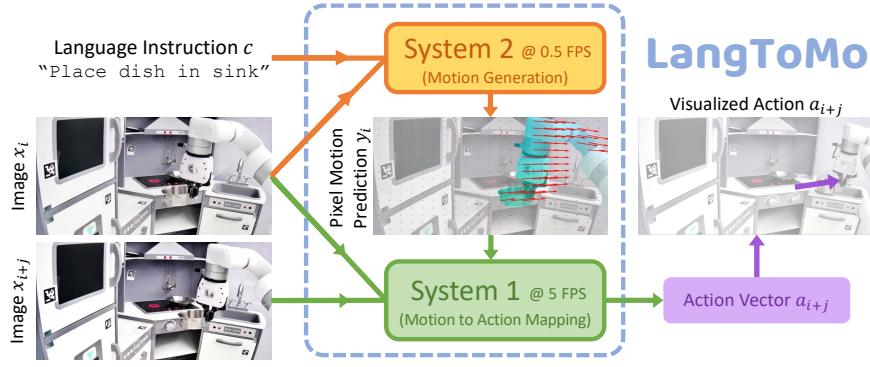


Figure 1: Dual-System VLA Framework, LangToMo, with pixel motion representations.

ABSTRACT

We present LangToMo, a vision-language-action framework structured as a dual-system architecture that uses pixel motion forecasts as intermediate representations. Our high-level *System 2*, an image diffusion model, generates text-conditioned pixel motion sequences from a single frame to guide robot control. Pixel motion—a universal, interpretable, and motion-centric representation—can be extracted from videos in a weakly-supervised manner, enabling diffusion model training on any video-caption data. Treating generated pixel motion as learned *universal representations*, our low level *System 1* module translates these into robot actions via motion-to-action mapping functions, which can be either hand-crafted or learned with minimal supervision. System 2 operates as a high-level policy applied at sparse temporal intervals, while System 1 acts as a low-level policy at dense temporal intervals. This hierarchical decoupling enables flexible, scalable, and generalizable robot control under both unsupervised and supervised settings, bridging the gap between language, motion, and action. Checkout kahnchana.github.io/LangToMo

1 INTRODUCTION

Translating open-ended natural language instructions into robot actions is a cornerstone of flexible robot control. We identify two key requirements to enable this: (i) universal representations that support operating diverse embodiments (Nair et al., 2022; Ren et al., 2025; Zheng et al., 2025), and (ii) benefiting from video-language data without action labels (Du et al., 2023b; Gu et al., 2023; Black et al., 2023; Ko et al., 2023; Cheang et al., 2025; Lee et al., 2025). We explore their intersection, proposing LangToMo, a vision–language–action framework structured as a *dual-system architecture*, inspired by dual-process theories of cognition (Kahneman, 2011) and recent hierarchical robotics frameworks (Belkhale et al., 2024; Black et al., 2024; Shi et al., 2025b; Nvidia et al., 2025; Intelligence et al., 2025). In our high level *System 2* module, we use pixel motion as the robot action representation. We use image diffusion to learn to predict pixel motion from a single image (observation) conditioned on a language described action. Subsequently, our embodiment-aware low level *System 1* deterministically projects these action representations into executable robot actions.

We adopt pixel motion—the apparent motion of pixels between frames—as our *universal motion representation*, because it is agnostic to embodiments, viewpoints, and tasks. By predicting pixel motion instead of full RGB images, LangToMo captures essential motion patterns more efficiently (i.e. with less training data, see Section 4.1) than text-to-video generation (Du et al., 2023b; Ko et al., 2023; Gu et al., 2023; Black et al., 2023). In contrast to operating with sparse point tracks (Yuan et al., 2024a; Wen et al., 2023; Xu et al., 2024; Bharadhwaj et al., 2024b), our dense pixel motion representation can capture both manipulator and object movements (see Figure 2). Our pixel motion features also retain the inherent 2D structure of the visual domain unlike prior work modeling pixel trajectories as 1D point tracks (Wen et al., 2023; Xu et al., 2024). Moreover, dense pixel motion can be freely computed from videos using off-the-shelf algorithms like RAFT (Teed & Deng, 2020), enabling scalable, weakly supervised training on large video-caption datasets, similar to prior work on predictive world models (Gu et al., 2023; Black et al., 2023; Zhang et al., 2025).

Optical flow, a measure of pixel motion (PM) between consecutive frames, has been leveraged to enhance motion-focused video generation (Liang et al., 2024; Koroglu et al., 2024), including in the robotics domain (Gao et al., 2025). PM calculated from current and future frames is used for robot control in Ko et al. (2023); Bharadhwaj et al. (2024a), further establishing the promise of this direction. In contrast, we directly generate PM from language and a single current frame (without access to future frames) using our System-2 module, offering greater data efficiency and performance (see Tables 2 to 4). Our predicted PM serves as an interpretable intermediate representation for downstream systems (e.g., our System-1), enabling even unsupervised control via hand-crafted mappings. Alternate motion signals in image-space are used in works like Sudhakar et al. (2024); Shridhar et al. (2024); Huang et al. (2024); Shi et al. (2025a), but they rely on explicit dense annotations limiting training scalability, unlike our System-2 formulation. Generating PM from a single image has also been explored (Walker et al., 2015; Gao et al., 2018; Aleotti et al., 2021), but has been limited to the visual domain with no language conditioning. In contrast, our System-2 module generates PM conditioned on both visual and textual cues with no access to future frames.

Sequences of PM generated by our System 2 are then be transformed into robot actions via *System 1*, a fast and deterministic controller. Specifically, System 1 consists of motion to action mappings that are *embodiment aware*. We explore two instantiations of System 1: (a) learning mappings directly from limited expert demonstrations, and (b) hand-crafting mappings by leveraging the interpretable nature of pixel motion (motivated by Ko et al. (2023)). Connecting System 1 and System 2 forms our overall language-conditioned robot control framework, LangToMo. This hierarchical formulation allows operating the expensive high-level System 2 at sparse temporal intervals while invoking the lightweight low-level System 1 at dense temporal intervals for efficient inference. This also allows independent training of each system, leading to better overall training efficiency.

In summary, our contributions are as follows:

- **Universal Action Representation:** 2D structured dense pixel motion as a learnable, interpretable, and manipulator-motion focused representation for robot control tasks.
- **Simple & Scalable Learning:** mapping natural language actions to motion representations (pixel motion sequences) with a history-aware conditional diffusion model trained on any video-caption data, without requiring pixel-level or action trajectory annotations.
- **Robotics Application:** conversion of learned action representations into action policies with minimal supervision, enabling operation under zero-shot and even unsupervised settings.

We evaluate LangToMo on both simulated and real-world environments, highlighting its effectiveness and generality across diverse robot control tasks.

2 RELATED WORK

Learning from Videos: Robot learning has a rich history of leveraging videos to extract sub-goal information, learn strong representations, or build dynamics models for planning (Lee & Ryoo, 2017; Finn & Levine, 2017; Sun et al., 2018; Kurutach et al., 2018; Pari et al., 2022; Nair et al., 2022; Shao et al., 2021; Chen et al., 2021; Bahl et al., 2022; Sharma et al., 2019; Du et al., 2023b; Sivakumar et al., 2022; Sudhakar et al., 2024; Ko et al., 2023; Hu et al., 2025; Ren et al., 2025). Several recent works learn representations connected to language modality from video-caption data (Du et al., 2023b; Sudhakar et al., 2024; Ko et al., 2023; Hu et al., 2025), but depend on additional action-trajectory

Table 1: **Unique Features of LangToMo.** *Dual System*: Decoupled architecture with system 1 & 2 modules trained separately with inference at distinct frequencies. *Dense Motion*: Uses no heuristic / training based point sub-sampling. *2D Structure*: Pixel motion as 2D grid instead of coordinate based 1D point sequence. Prior work A to G are Yuan et al. (2024a), Gao et al. (2025), Wen et al. (2023), Xu et al. (2024), Bharadhwaj et al. (2024b), Bharadhwaj et al. (2024a), Hu et al. (2025) respectively.

Feature	Ours	A	B	C	D	E	F	G
Past Aware	✓	✗	✗	✗	✗	✗	✗	✗
Dual System	✓	✗	✗	✓	✓	✗	✗	✓
Dense Motion	✓	✗	✓	✗	✗	✗	✗	✗
2D Structure	✓	✓	✓	✗	✗	✗	✗	✓
Text Condition	✓	✓	✓	✓	✓	✗	✓	✓

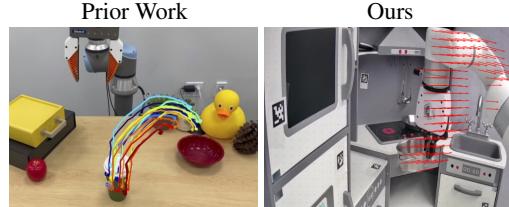


Figure 2: **Dense Motions:** Most prior work that use pixel trajectories focus on a subset of pixels often limited to objects of interest. The example from Xu et al. (2024) (left) focuses on the cup movement, but ignores important action information relevant to manipulator movement. In contrast, proposed LangToMo generates dense pixel motions that account for both object and manipulator movements (right).

annotations, pretrained segmentation models, or task-specific heuristics for robot control. We explore a similar direction, learning language-conditioned motion representations from video-caption data. In contrast to these works, our LangToMo learns representations that are *interpretable* and *motion-focused*, which we use for robot control with no additional supervision. Our focus on pixel motion also allows learning more generalizable representations with less data.

Pixel Motion to Actions: Robot navigation and control, especially in the context of aerial drones, has long benefited from optical flow representations (de Croon et al., 2021; Lee et al., 2020; Hu et al., 2024; Argus et al., 2020), inspired by animal perception system that use optical flow for stable control and movement (Götz, 1968; Arnold, 1974; Ros & Biewener, 2016; Baird et al., 2021). Video self-supervised learning has also extensively leveraged optical flow to learn motion representations (Han et al., 2020; Sharma et al., 2022). In robot control, trajectories of pixel subsets (Yuan et al., 2024a; Wen et al., 2023; Xu et al., 2024; Bharadhwaj et al., 2024b) have been used as intermediate representations, but often limit focus to specific image regions or objects, ignoring global information such as manipulator movement (e.g. see Figure 2). In contrast to prior work, our LangToMo models dense pixel motion (focusing on both object and manipulator movement) conditioned on textual action descriptions and visual current observations with no future frame dependency.

Diffusion-Based Motion Generation: Diffusion models have emerged as powerful generative frameworks capable of capturing complex data distributions through iterative denoising processes (Ho et al., 2020; 2022; Ramesh et al., 2022; Zhang et al., 2023; Singer et al., 2022; Villegas et al., 2022; Ge et al., 2022; Kumari et al., 2023; Zhang et al., 2022; Ren et al., 2022; Chen et al., 2023; Janner et al., 2022; Du et al., 2023a; Liu et al., 2023; Wang et al., 2023; Chi et al., 2023; Shridhar et al., 2024). While some works directly predict optical flow from image pairs (Saxena et al., 2023; Luo et al., 2024), these tackle well-defined inputs. In contrast, LangToMo generates pixel motion from a single image and language command, capturing the multi-modal nature of future motions. By also conditioning on past motion (extracted from current observations), our approach introduces temporal grounding, making it well-suited for robot control.

Language-Conditioned Robotic Manipulation: Several recent works use vision-language models for robot control (Brohan et al., 2023b;a; Padalkar et al., 2023; Reed et al., 2022; Wu et al., 2023; Octo Model Team et al., 2024; Driess et al., 2023; Kim et al., 2024; Yuan et al., 2024b; Niu et al., 2024; Zheng et al., 2024; Li et al., 2024; Zawalski et al., 2024; Hu et al., 2025; Sudhakar et al., 2024; Ko et al., 2023; Tian et al., 2024; Jeong et al., 2025) taking advantage of large-scale training with web-scale vision-language data. In contrast to prior work using sequential language models, we learn motion representations under weak supervision (only video-caption data) using zero action trajectory annotations. We also utilize an image diffusion model similar to Hu et al. (2025); Sudhakar et al. (2024); Ko et al. (2023) but differ by learning universal and interpretable motion representations directly, which even allows conversion to robot actions directly with no further training.

3 METHODOLOGY

We tackle the problem of robot control from natural language instructions by introducing a two-stage framework. Language and visual inputs are first encoded into pixel motion based representations,

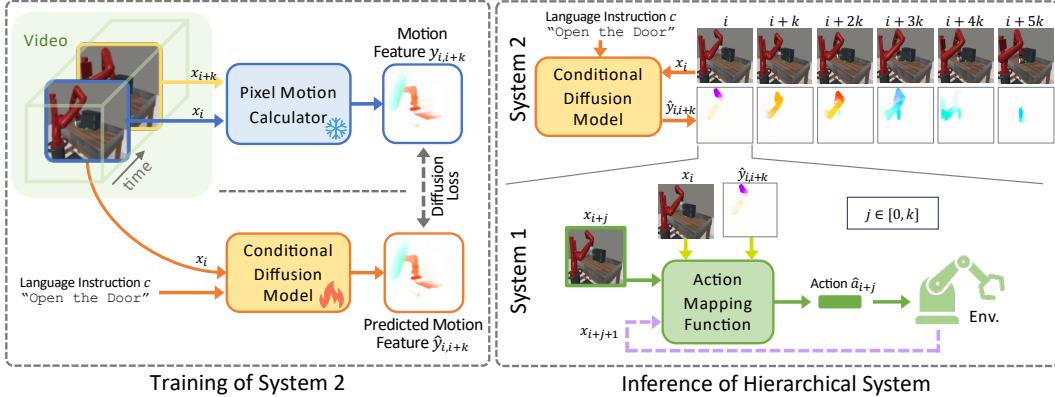


Figure 3: **Overview of LangToMo:** (Left) We learn to forecast pixel motion as universal motion features from video-caption pairs using scalable, self-supervised training of a diffusion model. (Right) Our *System 2* forecasts motion at sparse intervals (k), while *System 1* maps it to dense action vectors at j intervals ($j < k$).

which are then decoded into robot actions. This dual-system architecture comprises: *System 2*, a conditional image diffusion model that generates embodiment agnostic motion features at sparse temporal intervals acting as a high-level controller; and *System 1*, an embodiment aware low-level controller that maps these pixel motions to executable robot action vectors. An overview of our framework, LangToMo, is shown in Figure 3.

3.1 SYSTEM 2: PIXEL MOTION FORECAST

Optical flow estimation from frame pairs is a well-defined problem (exact solutions exist) that has been extensively studied (Liu et al., 2019; Teed & Deng, 2020; Xu et al., 2022; Luo et al., 2024). In contrast, estimating pixel motion (PM) from a single image and language instruction is inherently multi-modal: a caption-frame pair may correspond to multiple valid flows, each representing a different trajectory toward the goal. We use this challenging task as our training objective: learning a mapping from *language to motion*. Furthermore, we incorporate temporal context by conditioning on the motion of a previous state.

Consider a video clip $\mathbf{x} \in \mathbb{R}^{t \times h \times w \times c}$ with t, h, w, c for frames, height, width, and channels respectively. Also consider an embedding vector, \mathbf{c} representing the paired caption for that clip. Denoting the i -th frame of video as \mathbf{x}_i , we define pixel motion, $\mathbf{y}_{i,i+k}$, that corresponds to motion between frames $\mathbf{x}_i \rightarrow \mathbf{x}_{i+k}$ where k is a constant. Our language to motion mapping function, \mathcal{D} becomes,

$$\hat{\mathbf{y}}_{i,i+k} = \mathcal{D}(\mathbf{x}_i, \mathbf{y}_{i-k,i}, \mathbf{c} | \theta) \quad (1)$$

where $\hat{\mathbf{y}}_{i,i+k}$ is the predicted motion representation from the i -th state to $(i+k)$ -th state *without* knowing \mathbf{x}_{i+k} and θ are learnable parameters.

We reiterate the multi-modal output aspect of our mapping described in Equation (1) (i.e. one to many mapping due to multiple optimal $\hat{\mathbf{y}}_{i,i+k}$). Diffusion models have shown excellent abilities to model such distributions (Dhariwal & Nichol, 2021; Chi et al., 2023). Considering the 2D structure present in our images and pixel motion, for \mathcal{D} we elect to utilize a 2D conditional U-Net based diffusion model (Ramesh et al., 2022) operating at pixel level. Our goal is to learn a set of parameters, θ for this diffusion model based mapping as,

$$\arg \min_{\theta} \|\mathbf{y}_{i,i+k} - \mathcal{D}(\mathbf{x}_i, \mathbf{y}_{i-k,i}, \mathbf{c} | \theta)\|_2 \quad (2)$$

that allows our language to motion mapping to perform instruction based robot control. Next we dive into the learning process of our diffusion based implementation for this mapping function.

3.2 DIFFUSION BASED MOTION REPRESENTATION LEARNING

Background: Diffusion Models generate data by progressively denoising corrupted signals, optionally conditioned on a goal input. While inference follows this iterative refinement process, training is conducted more efficiently using parallel denoising steps: the model is trained to predict less

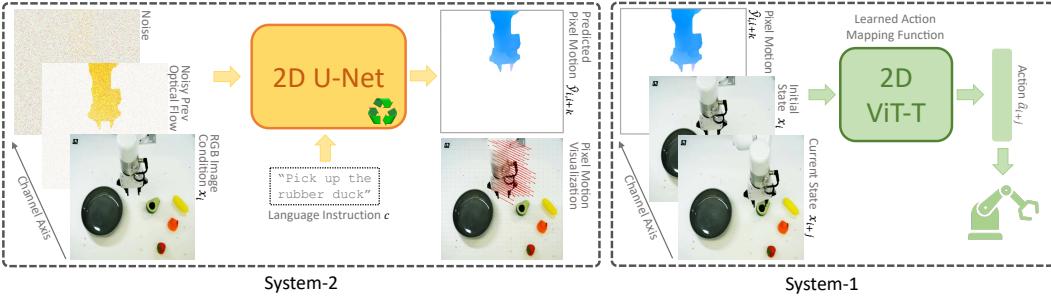


Figure 4: **LangToMo Architecture:** (Left) Diffusion model generates pixel motion conditioned on RGB image, prior motion, and caption. Visualized predictions are overlaid as arrows. (Right) ViT-T network maps predicted motion to robot actions in supervised setting, conditioned on initial/current states and target motion.

noisy versions of intermediate corrupted signals generated from clean data, a procedure analogous to teacher forcing (more details in Appendix E).

Architecture: The defacto architecture for diffusion based conditional image generation is the 2D conditional U-Net (Ronneberger et al., 2015), which maps between 2D RGB images with an embedding based conditioning through cross-attention in the model intermediate layers. Basing off this setup, we modify the input and output heads to process 7 and 2 channel tensors respectively (instead of default 3 channel RGB). Two of the input channels and the two output channels correspond to our pixel motion target (noise input and clean output). The remaining 5 input channels correspond to our 2D-structured conditions: previous pixel motion (2 channels) and current state image (3 channels). These conditional inputs are not subject to the standard noise corruption schedule during training or inference (details in Appendix E). The textual embedding is provided as the default embedding condition. Our channel modification to accommodate additional structured conditions allows a minimal design, retaining the general structure of the U-Net that is known to excel at 2D generative modeling. Such input channel concatenation based conditioning has been used in diffusion literature for different tasks (Saxena et al., 2023; Ho et al., 2022) and is inspiration for our design. We illustrate this architecture in Figure 4 (left).

Calculating Pixel Motion Ground-truth: We utilize the RAFT algorithm (Teed & Deng, 2020) to calculate our target pixel motion $\mathbf{y}_{i,i+k}$, using frames x_i and x_{i+k} . This is an efficient iterative algorithm that calculates a good estimate of optical flow, in other words, pixel motion. Each pixel motion, $\mathbf{y}_{i,i+k} \in \mathbb{R}^{h \times w \times 2}$, contains two channels for spatial directions, that are normalized to a $(0, 1)$ range. All motion is represented within this 2D space - extensions to a third depth dimension are left as a future direction. Our experiments indicate the sufficiency of such 2D spaces to encode motions relevant to robot actions. We note that given the presence of background motions in both natural and simulation images (e.g. shadows moving with objects), this target pixel motion contains noise that is not directly relevant to the underlying motion, underscoring the challenging nature of our self-supervision objective.

Previous Pixel Motion Representation: The other input signal to our mapping function is past pixel motion. Motivated by success of teacher forcing both language (Radford et al., 2019) and video (Song et al., 2025) generation, we use the target pixel motion of previous time steps during our System-2 training. We also note the importance of representing pixel motion relative to current state as our mapping function is conditioned on the current image (details in Appendix C). Similar findings are observed in image-pair based optical flow calculation literature (Ko et al., 2023).

Language Instruction Embedding: The primary input conditioning of our mapping function is the natural language based action description that is used to control the generated motions. Following prior robotics literature (Padalkar et al., 2023), we use a Universal Sentence Encoder model (Cer et al., 2018) to convert textual instructions to fixed size embedding vectors. This embedding model is trained to capture sentence level meanings. We use an off-the-shelf pretrained version, keeping all model parameters unchanged (more details in Appendix D).

Training: Our training uses the standard diffusion denoising objective (Ho et al., 2020) between predicted ($\hat{\mathbf{y}}_{i,i+k}$) and target ($\mathbf{y}_{i,i+k}$) pixel motion. The conditional 2D inputs, x_i and $\mathbf{y}_{i-k,i}$ are not subject to a noising schedule. The image condition, x_i , remains uncorrupted while the previous pixel motion, $\mathbf{y}_{i-1,i}$, is set to random noise or a partially corrupted version to align with inference settings.

We also introduce zero motion to ends of videos such that when textual instruction is complete, those visual states map to zero motion. More details in Appendix E.

Inference: We forecast pixel motion from i to $i + k$ timestamp using a 25-step DDIM schedule with only the current image observation \mathbf{x}_i . At the initial step, the model only takes the image \mathbf{x}_i (state observation), language instruction c , and zero vector as the previous pixel motion. For subsequent steps, previous motion is calculated from the past-current observation pair, enabling sequential pixel motion generation that drives the system toward fulfilling the language command.

3.3 SYSTEM 1: PIXEL MOTION TO ACTION MAPPING

Our System 2 produces pixel motion conditioned on a given state-instruction pair. We next detail how these pixel motion representations are mapped into action vectors that directly control the robot. Consider a mapping function, \mathcal{F} , operating at dense temporal intervals:

$$\hat{\mathbf{a}}_{i+j} = \mathcal{F}(\hat{\mathbf{y}}_{i,i+k}, \mathbf{x}_i, \mathbf{x}_{i+j}), \quad (3)$$

where $j \in [0, k]$, i is a multiple of k (for a hyperparameter k), and $\hat{\mathbf{a}}_{i+j}$ denotes the predicted action vector for the $(i + j)$ -th state. An overview of this formulation is shown in Figure 3 (right).

While *System 2* is trained as a general-purpose motion generator across diverse embodiments, viewpoints, and environments, action vectors \mathbf{a}_i are inherently embodiment-specific. Hence, we design *embodiment-aware* mapping functions to serve as *System 1 (Action Mapping)*, that are capable of converting pixel motion into executable robot actions.

Learned Mapping: We implement a neural network-based mapping function that can be trained using ground-truth action trajectories. Given the 2D spatial structure of the inputs to \mathcal{F} (i.e., $\hat{\mathbf{y}}_{i,i+j}$, \mathbf{x}_i , \mathbf{x}_{i+j}), we channel-concatenate them and feed the resulting tensor to a lightweight vision transformer to predict action vectors. This architecture is illustrated in Figure 4 (right). The network is trained on a limited amount of embodiment-specific demonstration data. Connecting this learned *System 1* with *System 2* following Equation (3), we obtain a complete pipeline for language-conditioned robot control. We refer to the resulting system, which uses a supervised learned mapping, as LTM-S.

Hand-Crafted Mapping: The interpretable nature of pixel motion also enables hand-crafted designs for \mathcal{F} . We refer to the resulting pipeline based on hand-crafted mappings as LTM-H. For simulated environments where ground-truth segmentations and depth maps are available, we follow the methodology in (Ko et al., 2023) to define action mappings, ensuring a fair evaluation of the utility of our pixel motion predictions compared to prior works. For real-world robot control, we construct viewpoint-specific hand-crafted mappings following (Li et al., 2024). Further details on both learned and hand-crafted mappings are provided in Appendix F.

We highlight how our System 1 operates at a frequency different to our System 2, allowing a balance between efficiency and dense control. Our System 1 is also designed to be lightweight, given how it performs an almost deterministic mapping.

4 EXPERIMENTAL RESULTS

We conduct experiments on 15 task styles spanning both simulated and real-world environments to highlight the strong performance of our proposed LangToMo framework. We also present multiple ablations to justify key design choices within our method.

Implementation Details: Our framework consists of *System 2 (Motion Generation)* containing a diffusion model, and *System 1 (Action Mapping)* containing either a learned or hand-crafted mapping function. We pretrain the diffusion model on a subset of the OpenX dataset (Padalkar et al., 2023), followed by optional fine-tuning on downstream task datasets. Pretraining is performed for 300,000 iterations with a learning rate of 1e-4, following a cosine learning rate schedule with 500 warmup steps, using 8 A100 GPUs (48GB) with a per-device batch size of 32 samples. Fine-tuning is performed for 100,000 iterations on 4 A5000 GPUs (24GB) with a batch size of 32 and a learning rate of 1e-5, again following a cosine schedule with 500 warmup steps. The learned action mapping (System 1) is trained separately using a vision transformer for 10,000 iterations on a single A5000 GPU with a batch size of 128 and a learning rate of 1e-4. During inference of our System 2 diffusion model, we use a DDIM scheduler with 25 steps to generate flow sequences, starting from noise.

Table 2: Zero-Shot Transfer on Real World Tasks: We directly deploy our pretrained model (with no fine-tuning) on real world tasks. We highlight our strong performance compared to baselines in this highly challenging setting. Evaluations follow [Li et al. \(2024\)](#).

Method	Video Only Training	T1	T2	T3	T4	Avg
RT-2 Style	✗	0	0	0	0	0
LLaRA	✗	40	20	10	20	22.5
AVDC	✓	0	0	0	0	0
GPT-4o	✓	20	30	10	15	18.8
LTM-H (ours)	✓	40	30	35	30	33.8

Table 3: Finetuned on Real World: LangToMo benefits from both robot (RD) and human (HD) demonstrations highlighting the embodiment agnostic nature of our Sys-2, in contrast to prior work.

Method	Data	T1	T2	T3	T4	Average
RT-2 Style	-	0	0	0	0	0
LLaRA	-	70	80	55	55	65.0
AVDC	RD	10	20	0	0	7.5
AVDC	RD+HD	0	0	0	0	0.0
LTM-H (ours)	HD	40	35	40	30	36.3
LTM-H (ours)	RD	80	70	65	60	68.8
LTM-H (ours)	RD+HD	80	75	65	65	71.3

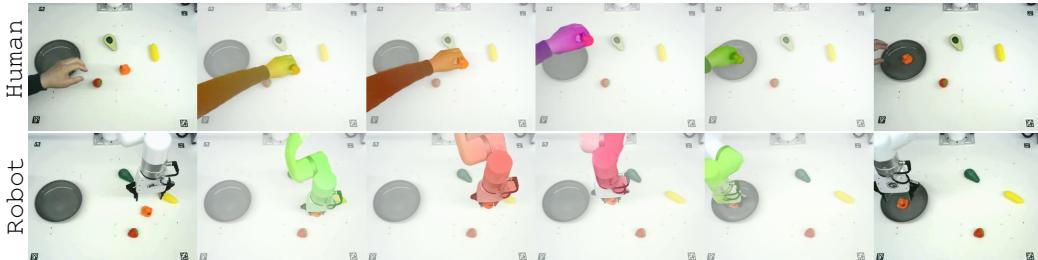


Figure 5: Human (HD) & Robot (RD) Demonstrations: We visualize frames from two sample demonstrations on our real world environment. Pixel motion overlaid on intermediate frames. These human (top) and robot (bottom) demonstrations can both be used to fine-tune our System-2, highlighting a unique aspect of LangToMo. Both examples use the same caption "Pick up the rubber duck and place on the bowl."

For each invocation of System 2, we run System 1 for 10 control steps (or until convergence in the hand-crafted setting). This hierarchical procedure is repeated until the episode terminates.

4.1 REAL-WORLD ENVIRONMENT

We first evaluate on four styles of real world tasks using the xArm Table Top environment constructed following [Li et al. \(2024\)](#). We select this environment and task styles for its ease of fair comparison to prior work, interpretable action dynamics (each state suggests a clear next motion), and demands for visual grounding, semantic understanding, and distractor robustness. The tasks involve object manipulations specified by language commands (details in Appendix G).

Training & Evaluation: We train *System 2* on the OpenX subset, followed by optional fine-tuning on demonstrations from the real-world environment. We collect 10 robot demonstrations (RD) and 50 human demonstrations (HD) per task style. We replicate AVDC ([Ko et al., 2023](#)) by training under identical conditions. All other baselines are implemented following settings from [Li et al. \(2024\)](#). For *System 1*, we construct a hand-crafted mapping function combining ideas from [Ko et al. \(2023\)](#); [Li et al. \(2024\)](#) (details in Appendix G). We follow evaluation settings identical to [Li et al. \(2024\)](#), evaluating each policy across 4 task styles with a fixed camera view and 20 randomized trials per task style. Each trial uses different initial positions of the objects present in the environment.

Zero-Shot Results: We present results for zero-shot evaluation in Table 2. Strong performance of this *pre-trained only* Sys-2 module highlight the significance of our large-scale pretraining.

Finetuning Results: We next fine-tune our Sys-2 module on the robot (RD) and human (HD) demonstrations, presenting these results in Table 3. Compared to AVDC ([Ko et al., 2023](#)) that makes RGB predictions, our Sys-2 module that predicts pixel motion benefits from human demonstrations (+2.5%). In contrast, AVDC predictions break down when trained on both RD and HD. We attribute this to the greater difference between human vs robot manipulators in RGB space compared to pixel motion space (see motion overlay in Figure 5, distribution analysis in Appendix J, and [Xu et al. \(2024\)](#)). We also highlight that fine-tuning here uses only video-caption pairs and no action ground-truth. This is what allows learning from both RD and HD data. Collection of such human demonstrations (HD) is much faster compared to teleoperated robot demonstrations (RD) ([Cheang et al., 2025](#)), underscoring the value of our LangToMo framework.

Table 4: **Results on MetaWorld Environment:** We report the mean success rate across tasks. Each entry of the table shows the average success rate aggregated from 3 camera poses with 25 seeds for each camera pose.

	<i>door-open</i>	<i>door-close</i>	<i>basketball</i>	<i>shelf-place</i>	<i>btn-press</i>	<i>btn-top</i>	<i>faucet-close</i>	<i>faucet-open</i>	<i>handle-press</i>	<i>hammer</i>	<i>assembly</i>	<i>Overall</i>
BC-Scratch	21.3	36.0	0.0	0.0	34.7	12.0	18.7	17.3	37.3	0.0	1.3	16.2
BC-R3M	1.3	58.7	0.0	0.0	36.0	4.0	18.7	22.7	28.0	0.0	0.0	15.4
Diffusion Policy	45.3	45.3	8.0	0.0	40.0	18.7	22.7	58.7	21.3	4.0	1.3	24.1
UniPi (With Replan)	0.0	36.0	0.0	0.0	6.7	0.0	4.0	9.3	13.3	4.0	0.0	6.1
Im2Flow2Act	0.0	0.0	0.0	4.0	6.3	0.0	7.3	4.7	0.0	0.0	0.0	2.0
ATM	75.3	90.7	24.0	16.3	77.3	76.7	50.0	62.7	92.3	4.3	2.0	52.0
AVDC (Flow)	0.0	0.0	0.0	0.0	1.3	40.0	42.7	0.0	66.7	0.0	0.0	13.7
AVDC (Default)	72.0	89.3	37.3	18.7	60.0	24.0	53.3	24.0	81.3	8.0	6.7	43.1
AVDC (PT)	72.0	88.7	37.3	18.7	58.7	24.3	53.3	24.0	81.3	8.0	6.7	42.9
LTM-H (Ours)	76.0	94.7	38.0	15.3	82.0	84.7	41.3	33.3	97.3	4.3	6.7	52.1
LTM-S (Ours)	77.3	95.0	39.0	20.3	82.7	84.3	52.3	68.3	98.0	10.3	7.7	57.7

4.2 METAWORLD SIMULATED ENVIRONMENT

Our second set of evaluations use 11 tasks from the MetaWorld (Yu et al., 2019) simulated environment containing a Sawyer robot arm, constructed following (Ko et al., 2023). We select this environment and tasks for direct comparison to (Ko et al., 2023), which is the closest prior work to our method, that similarly uses dense pixel motion for robot manipulation. These tasks also span key challenges in robot control such as complex 3D motions (e.g. button-press-top), contact-rich manipulation (e.g. basket-ball), and semantic understanding (e.g. door open vs close). Each task episode corresponds to successfully completing an action described in natural language.

Training: We pretrain *System 2* on the OpenX subset, followed by additional training on 165 MetaWorld videos (identical to the split used in Ko et al. (2023)). For the learned variant of *System 1*, we train on 20 expert demonstrations per task. We also implement a hand-crafted variant of System 1, following the design in Ko et al. (2023) to ensure fair comparison.

Baselines: All baselines follow settings in Ko et al. (2023)). The behaviour cloning (BC) baselines are trained on 15,216 labeled frame-action pairs (over 5x more data). BC-Scratch uses a randomly initialized ResNet-18 while BC-R3M uses pretrained weights from Nair et al. (2022). Diffusion Policy follows settings in Chi et al. (2023) and is trained on the same data. UniPi (Du et al., 2023b) uses the outputs of the AVDC model and its predictor is trained on the same data used for BC baselines. AVDC (Flow & Default) are trained identical to Ko et al. (2023) using same 165 Metaworld videos. AVDC (PT) is additionally pretrained on our OpenX subset making the training identical to LTM. Im2Flow2Act (Xu et al., 2024) and ATM (Wen et al., 2023) follow their default implementations and are also trained identically using the same training data as LangToMo.

Evaluation: Following evaluation settings identical to (Ko et al., 2023), we evaluate each policy across 11 tasks. For each task, videos are rendered from 3 distinct camera poses, with 25 randomized trials (different initial positions of the robot arm and objects) for each view.

Results: We present the success rates for the 11 tasks and the average across tasks in Table 4. Notably, several prior works (Du et al., 2023b; Ko et al., 2023) exhibit moderate success rates, underscoring the difficulty of the benchmark. Both our LTM-H and LTM-S variants achieve strong overall performance, highlighting the effectiveness of our framework. Our approach of directly predicting pixel motion compared to RGB in AVDC (Ko et al., 2023) achieves clear performance improvements (+9.0%). Moreover, AVDC fails to benefit from pretraining, which we attribute to the greater domain gap across embodiments in RGB space compared to pixel motion space. Another important point of comparison is the AVDC (flow) baseline, which also uses pixel motion prediction but differs in model architecture, flow representation, and training procedures. We attribute this improved performance of LangToMo over AVDC to our unique design choices. In comparison to ATM (Wen et al., 2023), our improved performance highlights the usefulness of our dense pixel motion features.

Table 5: Ablation Study: We report mean success rate % (overall) on MetaWorld benchmark with our LTM-S variant. (left) Results highlight importance of key components in our System-2 model. (right) Results justify several high-level design choices of our framework.

Img	Lang	Prev Flow	PT	Overall	Method	Overall
✓	✓	✓	✓	57.7	Ours (default)	57.7
✓	✓	✓	✗	53.1	No diffusion	16.2
✓	✓	✗	✗	50.2	CA instead of concat	15.8
✓	✗	✗	✗	39.7	Sys-1 & 2 same freq	48.7
✗	✓	✗	✗	5.6	Only learned Sys-1	3.2

4.3 ABLATION STUDIES

We conduct a series of ablative studies with LTM-S on the MetaWorld benchmark to evaluate the importance of key components within LangToMo. Results are summarized in Table 5.

System 2 Input Conditioning & Pretraining: Removing visual (“Img”), language (“Lang”), or history information (“Prev Flow”) from conditional inputs to the diffusion model significantly reduces performance, highlighting importance of each conditioning signal. On the other hand, removing diffusion model pretraining (“PT”) leads to a modest performance drop, indicating that while pretraining aids convergence and performance, the framework remains effective with limited finetuning alone.

Simpler Baselines: Replacing diffusion (“No diffusion”) with an autoencoder breaks System-2 learning process. We believe diffusion is more suited for learning the multi-modal output-space of language to motion distributions. Modifying conditioning strategy to cross-attention (“CA instead of concat”) also degrades performance. We attribute this to loss of spatial information when performing cross-attention with spatially-averaged visual embeddings. Skipping the iterative System-1 design (running System-1 at same frequency), and generating multiple actions per System-2 generated motion at once (“Sys-1 & 2 same freq”) also degrades success rates, validating our design choices. Additionally, bypassing intermediate motion representations (“Only learned Sys-1”) leads to poor results, underscoring the clear role played by our System-2 module. See Appendix I for a detailed discussion.

5 CONCLUSION

We presented LangToMo, a scalable vision-language-action framework that decouples motion generation and action execution through a dual-system architecture. By leveraging diffusion models to learn universal pixel motion representations from video-caption data, our *System 2* enables generalizable, interpretable motion planning without dense supervision. These motions are translated into robot actions by our embodiment-aware *System 1*, using either learned or hand-crafted mappings. Extensive experiments across simulated and real-world environments demonstrate strong performance of LangToMo, highlighting the promise of universal motion representations as a bridge between language, vision, and action for scalable robot learning.

LIMITATIONS

LangToMo is pretrained on large-scale video-caption data, but relies on hand-crafted or learned action mappings in System 1 which can be costly for each new embodiment. Learning robust, transferable mappings remains an open challenge. Also, our framework models motion using 2D pixel motions, which currently lacks depth cues. Extending to 3D motion representations is left as a future direction. In terms of speed, despite operating at sparse intervals, System 2 relies on diffusion models that remain computationally expensive at inference time, limiting use in resource-constrained deployments. This is another future direction we hope to explore further. Finally, we currently do not account for ego motion in training videos: we limit our training to fixed camera videos (no ego motion). A key next direction is extending our System-2 training to include videos with ego motion, which would allow scaling to any kind of video.

REFERENCES

- Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning optical flow from still images. In *CVPR*, pp. 15196–15206, 2021. [2](#)
- Max Argus, Lukás Hermann, Jon Long, and Thomas Brox. Flowcontrol: Optical flow based visual servoing. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7534–7541, 2020. URL <https://api.semanticscholar.org/CorpusID:220280145>. [3](#)
- Geoff Arnold. Rheotropism in fishes. *Biological Reviews*, 49, 1974. URL <https://api.semanticscholar.org/CorpusID:30755969>. [3](#)
- Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. In *Robotics: Science and Systems*, 2022. [2](#)
- Emily Baird, Norbert Boeddeker, and Mandyam V. Srinivasan. The effect of optic flow cues on honeybee flight control in wind. *Proceedings of the Royal Society B*, 288, 2021. URL <https://api.semanticscholar.org/CorpusID:231643236>. [3](#)
- Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *ArXiv*, abs/2403.01823, 2024. URL <https://api.semanticscholar.org/CorpusID:268249108>. [1](#)
- Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *ArXiv*, 2024a. [2](#), [3](#)
- Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *ECCV*, 2024b. [2](#), [3](#)
- Kevin Black, Mitsuhiro Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *ArXiv*, abs/2310.10639, 2023. URL <https://api.semanticscholar.org/CorpusID:264172455>. [1](#), [2](#)
- Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0 : A vision-language-action flow model for general robot control. *ArXiv*, abs/2410.24164, 2024. URL <https://api.semanticscholar.org/CorpusID:273811174>. [1](#)
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023a. [3](#)
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *Robotics science and systems (RSS)*, 2023b. [3](#)
- Daniel Matthew Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *ArXiv*, 2018. [5](#), [18](#)

-
- Chi-Lam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, Hao Niu, Wenxuan Ou, Wanli Peng, Zeyu Ren, Haixin Shi, Jiawen Tian, Hongtao Wu, Xin Xiao, Yuyang Xiao, Jiafeng Xu, and Yichu Yang. Gr-3 technical report. *ArXiv*, abs/2507.15493, 2025. [1](#), [7](#)
- Annie S Chen, Suraj Nair, and Chelsea Finn. Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos. In *Robotics: Science and Systems*, 2021. [2](#)
- Xin Chen, Yanchao Li, Zhen Li, Zhen Wang, Li Wang, and Chen Qian. Moddm: Text-to-motion synthesis using discrete diffusion model. *arXiv preprint arXiv:2308.06240*, 2023. [3](#)
- Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *ArXiv*, abs/2303.04137, 2023. [3](#), [4](#), [8](#)
- Guido C.H.E. de Croon, Christophe de Wagter, and Tobias Seidl. Enhancing optical-flow-based control by learning visual appearance cues for flying robots. *Nature Machine Intelligence*, 3:33 – 41, 2021. URL <https://api.semanticscholar.org/CorpusID:231655448>. [3](#)
- Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Neural Information Processing Systems*, 2021. [4](#)
- Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. [3](#)
- Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, Reuse, Recycle: Compositional Generation with Energy-Based Diffusion Models and MCMC. In *International Conference on Machine Learning*, 2023a. [3](#)
- Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning Universal Policies via Text-Guided Video Generation. *arXiv:2302.00111*, 2023b. [1](#), [2](#), [8](#)
- Chelsea Finn and Sergey Levine. Deep Visual Foresight for Planning Robot Motion. In *IEEE International Conference on Robotics and Automation*, 2017. [2](#)
- Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. In *ICLR*, 2025. [2](#), [3](#)
- Ruohan Gao, Bo Xiong, and Kristen Grauman. Im2flow: Motion hallucination from static images for action recognition. In *CVPR*, pp. 5937–5947, 2018. [2](#)
- Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer, 2022. [3](#)
- Karl Georg Götz. Flight control in drosophila by visual perception of motion. *Kybernetik*, 4:199–208, 1968. URL <https://api.semanticscholar.org/CorpusID:24070951>. [3](#)
- Xianfan Gu, Chuan Wen, Jiaming Song, and Yang Gao. Seer: Language instructed video prediction with latent diffusion models. *ArXiv*, abs/2303.14897, 2023. URL <https://api.semanticscholar.org/CorpusID:257766959>. [1](#), [2](#)
- Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *ArXiv*, abs/2010.09709, 2020. URL <https://api.semanticscholar.org/CorpusID:224703413>. [3](#)
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. [3](#), [5](#), [18](#)
- Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video Diffusion Models. In *Neural Information Processing Systems*, 2022. [3](#), [5](#)

Yu Hu, Yuang Zhang, Yunlong Song, Yang Deng, Feng Yu, Linzuo Zhang, Weiyao Lin, Danping Zou, and Wenxian Yu. Seeing through pixel motion: Learning obstacle avoidance from optical flow with one camera. *ArXiv*, abs/2411.04413, 2024. URL <https://api.semanticscholar.org/CorpusID:273877940>. 3

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *ICML*, 2025. 2, 3, 16, 17, 19

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Fei-Fei Li. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *ArXiv*, 2024. 2

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Rich Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. π 0.5: a vision-language-action model with open-world generalization, 2025. URL <https://api.semanticscholar.org/CorpusID:277993634>. 1

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning*, 2022. 3

Youngjoon Jeong, Junha Chun, Soonwoo Cha, and Taesup Kim. Object-centric world model for language-guided manipulation. *ArXiv*, abs/2503.06170, 2025. URL <https://api.semanticscholar.org/CorpusID:276903201>. 3

Daniel Kahneman. Thinking, fast and slow, 2011. 1

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 3

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Josh Tenenbaum. Learning to act from actionless videos through dense correspondences. *ArXiv*, abs/2310.08576, 2023. 1, 2, 3, 5, 6, 7, 8, 16, 17, 18, 19

Mathis Koroglu, Hugo Caselles-Dupr'e, Guillaume Jeanneret Sanmiguel, and Matthieu Cord. On-lyflow: Optical flow based motion conditioning for video diffusion models, 2024. 2

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion, 2023. 3

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning Plannable Representations with Causal InfoGAN. In *Neural Information Processing Systems*, 2018. 2

Jangwon Lee and Michael S Ryoo. Learning Robot Activities from First-Person Human Videos Using Convolutional Future Regression. In *CVPRW*, 2017. 2

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space. *ArXiv*, 2025. 1

Keuntaek Lee, Jason Gibson, and Evangelos A. Theodorou. Aggressive perception-aware navigation using deep optical flow dynamics and pixelmpc. *IEEE Robotics and Automation Letters*, 5:1207–1214, 2020. URL <https://api.semanticscholar.org/CorpusID:210064565>. 3

Xiang Li, Cristina Mata, Jong Sung Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, and Michael S. Ryoo. Llara: Supercharging robot learning data for vision-language policy. *ArXiv*, abs/2406.20095, 2024. 3, 6, 7, 18, 19

-
- Jingyun Liang, Yuchen Fan, Kai Zhang, Radu Timofte, Luc van Gool, and Rakesh Ranjan. Movideo: Motion-aware video generation with diffusion model. In *European Conference on Computer Vision*, 2024. URL <https://api.semanticscholar.org/CorpusID:273232410>. 2, 18
- Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selfflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4571–4580, 2019. 4
- Weiyu Liu, Yilun Du, Tucker Hermans, Sonia Chernova, and Chris Paxton. StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects. In *Robotics: Science and Systems*, 2023. 3
- Ao Luo, Xin Li, Fan Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. Flowdiffuser: Advancing optical flow estimation with diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19167–19176, 2024. 3, 4
- Oier Mees, Lukás Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7:7327–7334, 2021. 16
- Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *Conference on Robot Learning*, 2022. 1, 2, 8
- Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024. 3
- Nvidia, Johan Bjorck, et al. Gr0ot n1: An open foundation model for generalist humanoid robots. *ArXiv*, abs/2503.14734, 2025. 1
- Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics science and systems (RSS)*, Delft, Netherlands, 2024. 3
- Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 3, 5, 6, 18
- Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems*, 2022. 2
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. 5
- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *preprint*, 2022. [arxiv:2204.06125]. 3, 4
- Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Freitas de Nando. A generalist agent. In *Trans. on Machine Learning Research*, 2022. 3
- Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *ArXiv*, abs/2501.06994, 2025. URL <https://api.semanticscholar.org/CorpusID:275471722>. 1, 2
- Zhiyuan Ren, Zhihong Pan, Xin Zhou, and Le Kang. Diffusion motion: Generate text-guided 3d human motion by diffusion model. *arXiv preprint arXiv:2210.12315*, 2022. 3

-
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. [5](#)
- Ivo G. Ros and Andrew A. Biewener. Optic flow stabilizes flight in ruby-throated hummingbirds. *Journal of Experimental Biology*, 219:2443 – 2448, 2016. URL <https://api.semanticscholar.org/CorpusID:11106817>. [3](#)
- Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *ArXiv*, abs/2306.01923, 2023. [3](#), [5](#)
- Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2Robot: Learning Manipulation Concepts from Instructions and Human Demonstrations. *IJRR*, 2021. [2](#)
- Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. Third-person visual imitation learning via decoupled hierarchical controller. In *Neural Information Processing Systems*, 2019. [2](#)
- Yash Sharma, Yi Zhu, Chris Russell, and Thomas Brox. Pixel-level correspondence for self-supervised learning from video. *ArXiv*, abs/2207.03866, 2022. URL <https://api.semanticscholar.org/CorpusID:250407930>. [3](#)
- Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos, 2025a. [2](#)
- Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *ArXiv*, abs/2502.19417, 2025b. URL <https://api.semanticscholar.org/CorpusID:276618098>. [1](#)
- Mohit Shridhar, Yat Long Lo, and Stephen James. Generative image as action models. *ArXiv*, abs/2407.07875, 2024. [2](#), [3](#)
- Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. [3](#)
- Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on Youtube. In *Robotics: Science and Systems*, 2022. [2](#)
- Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion, 2025. [5](#)
- Sruthi Sudhakar, Ruoshi Liu, Basile Van Hoorick, Carl Vondrick, and Richard Zemel. Controlling the world by sleight of hand. *ArXiv*, abs/2408.07147, 2024. [2](#), [3](#)
- Shao-Hua Sun, Hyeonwoo Noh, Sriram Somasundaram, and Joseph Lim. Neural program synthesis from diverse demonstration videos. In *International Conference on Machine Learning*, 2018. [2](#)
- Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pp. 402–419. Springer, 2020. [2](#), [4](#), [5](#)
- 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*, abs/2412.15109, 2024. URL <https://api.semanticscholar.org/CorpusID:274859727>. [3](#)
- Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description, 2022. [3](#)
- Jacob Walker, Abhinav Kumar Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *ICCV*, pp. 2443–2451, 2015. [2](#)
- Hsiang-Chun Wang, Shang-Fu Chen, and Shao-Hua Sun. Diffusion Model-Augmented Behavioral Cloning. *arXiv:2302.13335*, 2023. [3](#)

-
- Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *ArXiv*, 2023. 2, 3, 8, 19
- Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. 3, 17
- Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning Optical Flow via Global Matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 4
- Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *Conference on Robot Learning*, 2024. 2, 3, 7, 8, 19
- Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Conference on Robot Learning*, 2019. 8
- Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024a. 2, 3
- Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024b. 3
- Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning*, 2024. URL <https://api.semanticscholar.org/CorpusID:271097636>. 3
- Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3
- Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 3
- Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinjiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. Dreamvla: A vision-language-action model dreamed with comprehensive world knowledge. *ArXiv*, abs/2507.04447, 2025. URL <https://api.semanticscholar.org/CorpusID:280147743>. 2, 17
- Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. *ArXiv*, abs/2501.10105, 2025. URL <https://api.semanticscholar.org/CorpusID:275606605>. 1
- Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024. 3

Appendix

A DATASET DETAILS

We use a subset of OpenX for pretraining of our System-2 module. We use `fractal`, `taco_play`, `language_table`, `stanford_hydra`, `ucsd_pick_place`, `cmu_pickup`, and `utaustin_mutex` datasets from the OpenX collection. Frame sampling is performed uniformly to maintain a fixed, common action count between frames across each dataset by normalizing for control frequency. We present details of each subset in Table 6.

Table 6: **Pretraining Dataset:** We use 7 sub-datasets from the OpenX collection for pretraining of our System-2 module. Note that training is performed jointly with 3 different embodiments operated at different control frequencies. Our pixel motion based representations allows training jointly with such data using a common training objective across data from all embodiments.

Dataset	Control Frequency	Episodes	Size (GB)	Robot
<code>fractal</code>	3	73,499	111.06	Google Robot
<code>taco_play</code>	15	3,242	47.77	Franka
<code>language_table</code>	10	442,226	399.22	xArm
<code>stanford_hydra</code>	10	550	72.48	Franka
<code>ucsd_pick_place</code>	3	1,355	3.53	xArm
<code>cmu_pickup</code>	20	520	50.29	Franka
<code>utaustin_mutex</code>	20	1,500	20.79	Franka

B ADDITIONAL EXPERIMENTAL RESULTS

We present more experiments on our real world environment as well as two additional simulation environments to further investigate behaviour of our LangToMo framework.

B.1 SEMANTIC AWARENESS

In LangToMo, our System-2 module plays the role of understanding semantics within the action goal (textual command) and converting it into meaningful pixel motion representations. We empirically validate this functionality by visualizing two examples that contain the same visual observation but different action goals. We illustrate this in Figure 6. Our LangToMo System-2 module shows much better language awareness in comparison to the AVDC baseline (Ko et al., 2023).



Figure 6: **Semantic Awareness Visualization:** We visualize outputs from our System-2 module (ours; left two figures) for two examples containing the same starting state (visual observation) but different action goals (textual command). LangToMo generates meaningful motions for scenarios needing semantic understanding. We also compare against the AVDC baseline (Ko et al., 2023) (trained on data identical to our LangToMo) that generates next frame RGB images instead of pixel motion. For both cases, AVDC generates the same input frame as its output, i.e. a static next image, seemingly disregarding the language command.

B.2 CALVIN EVALUATION

CALVIN (Mees et al., 2021) is another simulation benchmark used in several recent works such as Hu et al. (2025). We evaluate our model on this benchmark following settings in Hu et al. (2025) and summarize these results in Table 7. All prior work numbers are directly borrowed from Hu et al.

Table 7: **CALVIN Evaluation:** Zero-shot long-horizon evaluation on the Calvin ABC→D benchmark where agent is asked to complete five chained tasks sequentially based on instructions.

Method	Training Data	<i>ith</i> Task Success Rate ↑					Avg. Len ↑
		1	2	3	4	5	
RT-1	100% ABC	0.533	0.222	0.094	0.038	0.013	0.90
Diffusion Policy	100% ABC	0.402	0.123	0.026	0.008	0.00	0.56
Robo-Flamingo	100% ABC	0.824	0.619	0.466	0.331	0.235	2.47
Uni-Pi	100% ABC	0.560	0.160	0.080	0.080	0.040	0.92
MDT	100% ABC	0.631	0.429	0.247	0.151	0.091	1.55
Susie	100% ABC	0.870	0.690	0.490	0.380	0.260	2.69
GR-1	100% ABC	0.854	0.712	0.596	0.497	0.401	3.06
Vidman	100% ABC	0.915	0.764	0.682	0.592	0.467	3.42
RoboUniview	100% ABC	0.942	0.842	0.734	0.622	0.507	3.65
VPP	100% ABC	0.965	0.909	0.866	0.820	0.769	4.33
DreamVLA	100% ABC	0.982	0.946	0.895	0.834	0.781	4.44
LTM-S (ours)	100% ABC	0.971	0.824	0.728	0.672	0.606	3.81
GR-1	10% ABC	0.672	0.371	0.198	0.108	0.069	1.41
VPP	10% ABC	0.878	0.746	0.632	0.540	0.453	3.25
LTM-S (ours)	10% ABC	0.896	0.769	0.652	0.596	0.467	3.38

(2025) since we follow their exact settings for evaluation. We explore the two settings of training on the full ABC split and 10% of the ABC split. Evaluation is always performed on the unseen D split. Each task is a set of five sequential sub-tasks and we use the task success rates along with average length metrics for evaluation similar to Hu et al. (2025).

In the first case (100% ABC), we perform competitively outperforming several recent works. Concurrent works, VPP (Hu et al., 2025) and DreamVLA (Zhang et al., 2025) outperform us on this split. We note that both these models are pretrained on significantly more data than our LTM model. VPP also uses a larger sized model (1.5B) compared to our LTM (0.86B). In the second case, (10% ABC), we outperform VPP and GR-1 (Wu et al., 2023) highlighting the data efficiency aspect of our LangToMo framework.

B.3 iTHOR EVALUATION

We next explore the ability to extend our method to benchmarks that involve ego motion of the robot (e.g. simple navigation tasks). Following prior work AVDC (Ko et al., 2023), we evaluate on the iThor benchmark and present results in Table 8. Results indicate clear improvements of our proposed LangToMo over naive baselines and prior work AVDC (Ko et al., 2023). The behaviour cloning (BC) baselines are implemented following Ko et al. (2023). Both AVDC and LTM-H (ours) are trained on the same data under common training settings for fair comparison.

Table 8: **Results on iThor Benchmark:** We follow the iThor dataset based evalution setup used in AVDC (Ko et al., 2023) to demonstrate that our method generalizes to robot movement based control as well (i.e. where ego motion occurs). Results indicate that our method outperforms AVDC across categories and overall.

Method	Kitchen	Living Room	Bedroom	Bathroom	Overall
BC-Scratch	1.7	3.3	1.7	1.7	2.1
BC-R3M	0.0	0.0	1.7	0.0	0.4
AVDC	26.7	23.3	38.3	36.7	31.3
LTM-H (ours)	27.3	23.7	40.0	36.7	31.9

Our LangToMo was not explicitly designed for such ego-motion tasks, but nevertheless is capable of performing such tasks similar to AVDC. We take these results as a promising indication that LangToMo can be further extended to better handle such ego-motion tasks.

C RELATIVE PIXEL MOTION

A key design choice in our formulation is to represent pixel motion with respect to the current frame (\mathbf{x}_t), rather than the previous frame (\mathbf{x}_{t+1}) or some other frame. This aligns with the structure of our conditional diffusion model, which receives \mathbf{x}_t as a secondary conditioning input. Predicting the transformation from \mathbf{x}_t to the next frame allows the model to more directly focus on the visual cues present in the current state. In contrast, predicting motion from \mathbf{x}_{t-1} or some other different frame would require indirect reasoning over a non-visible state, introducing additional complexity. Hence our approach is to represent past pixel motion (e.g. \mathbf{x}_{t-1} to \mathbf{x}_t) as \mathbf{x}_t to \mathbf{x}_{t-1} instead. While this may seem counterintuitive, we note how prior literature on image-pair-based optical flow prediction for video tasks has also found that defining motion in terms of a reference image—particularly the current frame that is visible—can lead to more stable and accurate flow estimates (Liang et al., 2024). Moreover, our experiments representing previous motion in a different manner lead to subpar performance, standing as further evidence.

We also experiment trying to predict an additional future motion relative to a future frame. We compare this against predicting that same future motion relative to the current frames. In this setting, the latter performs well while the former variant fails to learn meaningful motion signals predictions.

D LANGUAGE EMBEDDING MODEL

For the language embedding model, we employ the Universal Sentence Encoder (USE), a pre-trained model from (Cer et al., 2018). USE generates fixed-length vector representations of text, capturing rich semantic meaning, making it suitable for various natural language processing (NLP) tasks. Its widespread use in research, including works like OpenX (Padalkar et al., 2023), highlights its effectiveness in transforming textual input into meaningful embeddings even for robotic tasks. In our framework, the USE serves as a key component, encoding language instructions into dense vectors that are later used to guide the generation of motion representations. The model’s ability to produce consistent and high-quality embeddings enables seamless integration between language and vision modalities, ensuring that our system can accurately interpret and respond to diverse language commands.

E DIFFUSION MODEL DETAILS

In our diffusion model training, input noising is applied by adding Gaussian noise to the target motion data (following standard settings from Ho et al. (2020)). The image condition input and the previous flow are not subject to this noising. The previous flow is corrupted with a 50% chance. During corruption, a random amount of Gaussian noise is added. To ensure diverse and meaningful training, filtering and augmentation operations are performed on the frames as described next. The indices corresponding to consecutive frames (i and $i + 1$) are selected such that they maintain fixed intervals based on the video frame rate. Frames with zero optical flow (i.e., no motion) between i and $i + 1$ are filtered out to avoid irrelevant data. Additionally, to handle the completion of textual instructions, we introduce zero motion at the ends of videos, ensuring that these states map to a lack of motion when the instruction concludes. The visual inputs (images and optical flow) are cropped and resized, with appropriate transformations applied to the flow data to maintain consistency.

F HAND-CRAFTED MAPPING FUNCTIONS

Synthetic Environments: We follow the formulation of Ko et al. (2023) using a segmentation map of robot controller and a depth map of environment. The generated pixel motions are converted into directions in 3D space to move the robot controller based on these dense maps. We direct the reader to Ko et al. (2023) for further details.

Real World Environments: Motivated by Li et al. (2024), we build our real world environment with a single plane assumption (e.g. table top manipulation) and map the predicted pixel motions for the robot controller center points onto the plane (using visual geometry). An initial camera calibration is performed for the environment to obtain necessary camera matrices. After extracting a start and end



Figure 7: Real World Tasks: We illustrate the four real-world tasks following LLaRA Li et al. (2024). Start and end states are shown in the first and last columns, with predicted pixel motion (color indicates motion direction) overlaid on intermediate states. LangToMo performs these challenging tasks successfully (see results in Table 3).

position for a manipulation task following this setting, our position to action vector conversion is identical to Li et al. (2024). Examples of these tasks are shown in Figure 7.

G REAL WORLD EXPERIMENTS

We perform four styles of real world experiments as illustrated in Figure 7. The language instructions for the four examples in this figure, where each belongs to one of the four task styles, are as follows:

1. Pick up the duck and place on the bowl.
2. Pick up the duck and place on the tray.
3. Pick up the avocado and place on the bowl.
4. Pick up the corn and place on the tray.

Each task style contains similar textual commands that require some object manipulation in the table top environment. We select these following Li et al. (2024) to ensure fair comparisons to prior work.

H BASELINE DETAILS

Our key baselines are from AVDC (Ko et al., 2023) and LLaRA (Li et al., 2024). For both methods, we use their official implementations to replicate their results and evaluate ours under identical settings. For LLaRA, all results are reported on their inBC variant for fair comparison against our method (i.e. similar inputs during inference / no external scene object information). We also use official implementations for VPP (Hu et al., 2025), Im2Flow2Act (Xu et al., 2024), and ATM (Wen et al., 2023) for evaluating those baselines. All these baselines are trained on the same data as our LangToMo model.

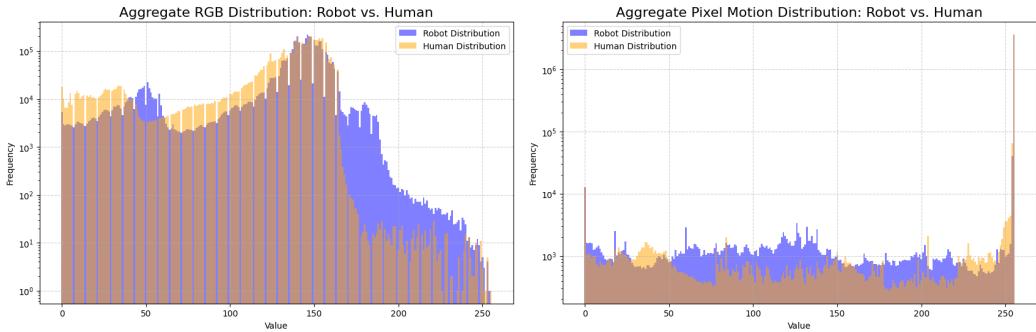


Figure 8: Histogram Comparisons for RGB and Pixel Motion Distributions of Robot vs Human Demonstrations: We illustrate two histograms that compare the aggregate pixel value distributions of 40 human and 40 robot demonstrations. For the RGB distributions (left), the high degree of separation and low overlap between the distributions of the two groups indicate significant differences in appearance, resulting in a high Symmetrized KL Divergence of 0.7881. In contrast, the pixel motion distributions (right) show substantial overlap between RGB and pixel motion, demonstrating the strong similarity in kinematic patterns between human and robot demonstrations, leading to a much lower Symmetrized KL Divergence of 0.0199. These results suggest that pixel motion is a more embodiment-agnostic metric for comparing demonstrations.

I DETAILED ABLATIONS

We discuss our ablations in Table 5 in detail in the following section.

System 2 Design Choices: We first ablate critical inputs to *System 2 (Motion Generation)*. Removing pretraining (“PT”) leads to a modest performance drop (from 53.6% to 53.1%), indicating that while pretraining aids convergence, the framework remains effective with limited finetuning alone. Removing the previous optical flow input (“Prev Flow”) results in a larger decline to 50.2%, validating the importance of temporal conditioning. Ablating the language embedding leads to a significant drop (to 39.7%), highlighting the necessity of semantic instruction guidance. Finally, removing the visual input (“Img”) results in near-random performance (5.6%), confirming that visual grounding is essential.

High-Level Framework Design: We next evaluate several higher-level architectural decisions. Removing the diffusion model (“No diffusion”) and training a direct regression using an autoencoder based setup (using same architecture as our diffusion model but without noise inputs and with a single time-step for training and inference) leads to a sharp performance drop (to 16.2%), underscoring the value of iterative, probabilistic modeling for motion generation. Replacing input concatenation with cross-attention (“CA instead of concat”) similarly degrades performance, suggesting that simple spatial concatenation is a more effective conditioning strategy for our setting. Using a multi-action decoder within *System 1* to run it at same frequency as our system 2 (“Sys-1 & 2 same freq”) results in slightly lower performance (48.7%), indicating that our default action mapping is more effective. Training only a learned *System 1* without leveraging pixel motions generated by Sys-2 (“Only learned Sys-1”) performs poorly (3.2%), demonstrating that our System-1 module simply learns to map the generated pixel motion to robot manipulator actions.

J PIXEL MOTION DISTRIBUTION ANALYSIS

We present an analysis of the divergence between human and robot demonstration data using both RGB pixel values and pixel motion. To quantify the difference, we first aggregated the pixel value distributions from 40 human and 40 robot demonstrations, creating two distinct distributions for each data type. We then computed the Symmetrized Kullback-Leibler (KL) divergence to measure the difference between the human and robot distributions. The results, with a high KL divergence for RGB (0.7881) and a very low one for pixel motion (0.0199), indicate that the distributions of RGB pixel values are significantly different between human and robot demonstrations, while the distributions of pixel motion are remarkably similar. As illustrated in Appendix J, this finding supports our hypothesis that pixel motion is a more embodiment-agnostic representation. The high divergence

in RGB is a function of embodiment-specific factors such as lighting, skin tone, robot color, and background, which vary greatly between human and robotic forms. Conversely, the low divergence in pixel motion demonstrates that, regardless of the physical embodiment, the fundamental kinematic patterns of the motion itself are consistent, making it a more universal measure for learning from demonstrations.