# Pixel Motion as Universal Representation for Robot Control

**Kanchana Ranasinghe, Xiang Li, Cristina Mata, Jongwoo Park, Michael S Ryoo**

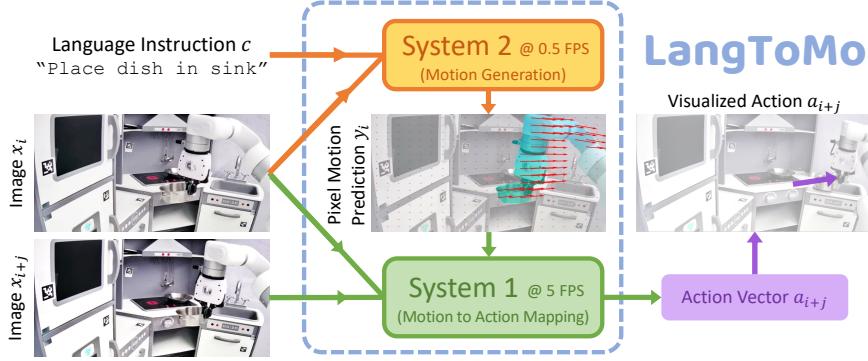Stony Brook University

`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 self-supervised manner, enabling diffusion model training on web-scale 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` for visualizations.

**Keywords:** Vision-Language-Action model, Self-Supervised, Diffusion

## 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 [1, 2, 3], and (ii) benefiting from large-scale video-language data without action labels [4, 5, 6, 7]. We explore their intersection, proposing LangToMo, a vision–language–action framework structured as a *dual-system architecture*, inspired by dual-process theories of cognition [8] and recent hierarchical robotics frameworks [9, 10, 11, 12, 13]. 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 (initial observation) conditioned on a

language described action. Subsequently, our embodiment-specific 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 than text-to-video generation [4, 7, 5, 6]. Pixel motion can be freely computed from videos using self-supervised methods like RAFT [14], enabling scalable, weakly supervised training on large video-caption datasets, similar to prior work on predictive world models [5, 6].

Optical flows, essentially a set of pixel motion (PM) between two consecutive frames, has been leveraged to enhance motion-focused video generation [15, 16]. Ko et al. [7] calculates flow from frame pairs to perform robot control, establishing the promise of this direction for robotics. In contrast, we directly generate PM from language and a single frame using our System-2 module, offering greater efficiency and performance. 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 [17, 18, 19, 20], but they rely on explicit dense annotations limiting training scalability, unlike our System-2 formulation.

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 task-specific action mappings tailored to different embodiments and viewpoints. 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 [7]). 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 control.

In summary, our contributions are as follows:

- **Universal Action Representation:** pixel motion as a learnable, interpretable, and motion-focused representation for robot control tasks.
- **Simple & Scalable Learning:** mapping natural language actions to motion representations (pixel motion sequences) with a conditional diffusion model trained on web-scale 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 [21, 22, 23, 24, 25, 1, 26, 27, 28, 29, 4, 30, 17, 7, 31, 2]. Several recent works learn representations connected to language modality from video-caption data [4, 17, 7, 31], but depend on additional action-trajectory 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 faster learning of more generalizable representations.

**Pixel Motion to Actions:** Robot navigation and control, especially in the context of aerial drones, has long benefited from optical flow representations [32, 33, 34, 35], inspired by animal perception system that use optical flow for stable control and movement [36, 37, 38, 39]. Video self-supervised learning has also extensively leveraged optical flow to learn motion representations [40, 41]. In
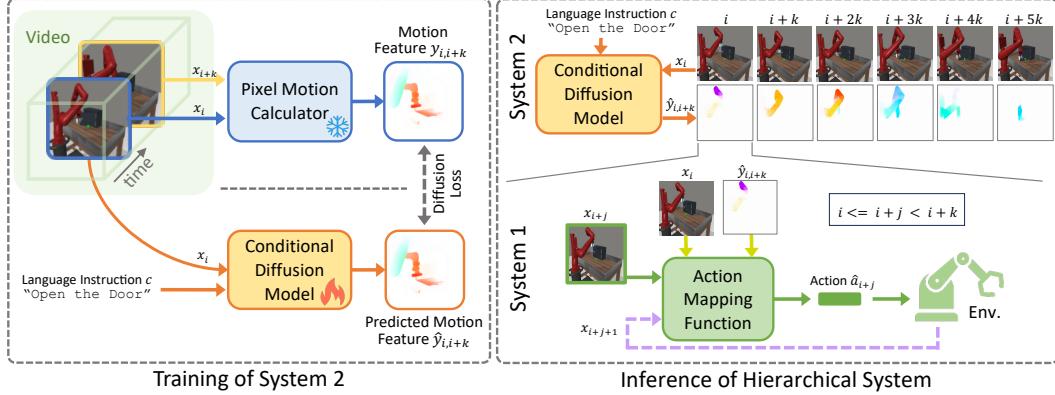
Figure 2: **Overview of LangToMo:** (Left) We learn to forecast pixel motion as universal motion features from video-caption pairs using scalable, self-supervised training of a diffusion model. (Right) Our *System 2* forecasts motion at sparse intervals ($k$), while *System 1* maps it to dense action vectors at $j$ intervals ($j < k$).

contrast to prior works, our LangToMo is the first to model optical flow from a single image (pixel motion) conditioned on textual action descriptions, allowing language conditioned robot control.

**Diffusion-Based Motion Generation:** Diffusion models have emerged as powerful generative frameworks capable of capturing complex data distributions through iterative denoising processes [42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 18]. While some works directly predict optical flow from image pairs [58, 59], these tackle well-defined inputs. In contrast, LangToMo generates pixel motion from a single image and language command, capturing the multimodal nature of future motions. By also conditioning on past motion, 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 [60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 31, 17, 7, 73, 74] 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 [31, 17, 7] 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, which are then decoded into robot actions. This dual-system architecture comprises: *System 2*, a conditional image diffusion model that generates motion at sparse temporal intervals as a high-level controller; and *System 1*, a task-specific low-level controller that maps these pixel motions to executable action vectors. An overview of our framework, LangToMo, is shown in Figure 2.

### 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 [75, 76, 14, 59]. 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 self-supervision 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 $x \in \mathbb{R}^{t \times h \times w \times c}$ with $t, h, w, c$ for frames, height, width, and channels respectively. Also consider an embedding vector, $c$ representing the paired caption for that clip. Denoting
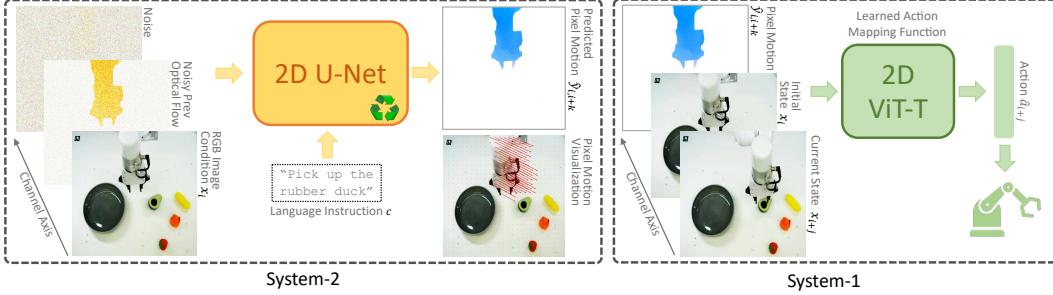
Figure 3: **LangToMo Architecture:** (Left) Diffusion model generates pixel motion conditioned on RGB image, prior motion, and caption. Visualized predictions are overlaid as arrows. (Right) ViT-T network maps predicted motion to robot actions in supervised setting, conditioned on initial/current states and target motion.

the $i$-th frame of video as $x_i$, we define pixel motion, $y_{i,i+k}$, that corresponds to motion between frames $x_i \rightarrow x_{i+k}$ where $k$ is a constant. Our language to motion mapping function, $\mathcal{D}$ becomes,

$$\hat{y}_{i,i+k} = \mathcal{D}\left(x_i, y_{i-k,i}, c \mid \theta\right) \tag{1}$$

where $\hat{y}_{i,i+k}$ is the predicted motion representation from the $i$-th state to $(i+k)$-th state *without* knowing $x_{i+k}$. $\theta$ 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 $\hat{y}_{i,i+k}$). Diffusion models have shown excellent abilities to model such distributions [77, 57]. 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 [44] operating at pixel level. Our goal is to learn a set of parameters, $\theta$ for this diffusion model based mapping as,

$$\arg\min_{\theta} ||y_{i,i+k} - \mathcal{D}\left(x_i, y_{i-k,i}, c \mid \theta\right)||_2 \tag{2}$$

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

**Architecture:** The defacto architecture for diffusion based conditional image generation is the 2D conditional U-Net [78], 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 D). 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 [58, 43] and is inspiration for our design. We illustrate this architecture in Figure 3 (left).

**Calculating Pixel Motion Ground-truth:** We utilize the RAFT algorithm [14] to calculate our target pixel motion $y_{i,i+k}$, 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, $y_{i,i+k} \in$

4

$\mathbb{R}^{h \times w \times 2}$, 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 frames pixel motion. Motivated by success of teacher forcing in generative modeling of both language [79] and videos [80], 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 B). Similar findings are observed in image-pair based optical flow calculation literature [58].

**Language Instruction Embeddding:** 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 [62], we use a Universal Sentence Encoder model [81] 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 C).

**Training:** Our training uses the standard diffusion denoising objective [42] between predicted $(\hat{y}_{i,i+k})$ and target $(y_{i,i+k})$ pixel motion. The conditional 2D inputs, $x_i$ and $y_{i-k,i}$ are not subject to a noising schedule. The image condition, $x_i$, remain uncorrupted while the previous pixel motion, $y_{i-1,i}$, 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 D.

**Inference:** We forecast pixel motion from $i$ to $i + k$ timestamp using a 25-step DDIM schedule with only the current image observation $x_i$. At the initial step, the model only takes the image $x_i$ (state observation), language instruction $c$, and random noise as the previous pixel motion. For subsequent steps, the previously predicted motion is reused, 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:

$$\hat{a}_{i+j} = \mathcal{F} \left( \hat{y}_{i,i+k}, x_i, x_{i+j} \right), \tag{3}$$

where $j \in [0, k]$, $i$ is a multiple of $k$ (for a hyperparameter $k$), and $\hat{a}_{i+j}$ denotes the predicted action vector for the $(i + j)$-th state. An overview of this formulation is shown in Figure 2 (right).

While *System 2* is trained as a general-purpose motion generator across diverse embodiments, viewpoints, and environments, action vectors $a_i$ are inherently embodiment-specific. Hence, we design *task-specific* mapping functions to serve as *System 1 (Action Mapping)*, 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., $\hat{y}_{i,i+j}$, $x_i$, $x_{i+j}$), we channel-concatenate them and feed the resulting tensor to a lightweight vision transformer to predict action vectors. This architecture is illustrated in Figure 3 (right). The network is trained on a limited amount of task-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 [7] 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 [71]. Further details on both learned and hand-crafted mappings are provided in Appendix E.

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 tasks 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 [62], 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. 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 MetaWorld Simulated Environment

MetaWorld [82] is a simulated benchmark containing several robot manipulation tasks with accompanying natural language instructions. Each task episode corresponds to successfully completing an action described in language. The environment utilizes a Sawyer robot arm.

**Training:** We train *System 2* (diffusion model) first on the OpenX subset, followed by additional training on 165 MetaWorld videos (identical to the split used in [7]). 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 [7] to ensure fair comparison.

**Evaluation:** Following evaluation settings identical to [7], 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. We replicate multiple baselines from [4, 7] under common settings for comparison.

**Results:** We present the success rates for the 11 tasks and the average across tasks in Table 1. Both our LTM-H and LTM-S variants achieve strong overall performance, highlighting the effectiveness of our framework. Notably, several strong approaches [4, 7] exhibit moderate success rates, underscoring the difficulty of the benchmark. An important point of comparison is the AVDC (flow) baseline from [7], which also uses pixel motion prediction but differs in model architecture, flow representation, and training procedures. The improved performance of LangToMo over AVDC demonstrates the impact of our design choices.

| | door-open | door-close | basketball | shelf-place | btn-press | btn-top | faucet-close | faucet-open | handle-press | hammer | assembly | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| 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 |
| 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 |
| LTM-H (Ours) | 76.0 | 94.7 | 38.0 | 15.2 | **82.0** | **84.7** | 41.3 | 33.3 | 97.3 | 4.2 | **6.9** | 52.1 |
| LTM-S (Ours) | **77.3** | **95.0** | **39.0** | **18.7** | **82.0** | 84.3 | **46.7** | **35.3** | **98.0** | 6.7 | **6.9** | **53.6** |

Table 1: **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.

| Method | Video Only Training | T1 | T2 | T3 | T4 | Avg |
|---|---|---|---|---|---|---|
| RT-2 Style [61] | ✗ | 0 | 0 | 0 | 0 | 0 |
| LLaRA [71] | ✗ | 70 | 80 | 55 | 55 | 65.0 |
| AVDC [7] | ✓ | 10 | 20 | 0 | 0 | 15.0 |
| LTM-H (ours) | ✓ | **80** | **70** | **65** | **60** | **68.8** |

Table 2: **Real World Task Performance:** We follow the setup in LLaRA [71] to evaluate model performance on real world tasks under fine-tuned settings.

| Method | Video Only Training | T1 | T2 | T3 | T4 | Avg |
|---|---|---|---|---|---|---|
| RT-2 Style [61] | ✗ | 0 | 0 | 0 | 0 | 0 |
| LLaRA [71] | ✗ | 40 | 20 | 10 | 20 | 22.5 |
| AVDC [7] | ✓ | 0 | 0 | 0 | 0 | 0 |
| GPT-4o [83] | ✓ | 20 | 30 | 10 | 15 | 18.8 |
| LTM-H (ours) | ✓ | 40 | 30 | 35 | 30 | 33.8 |

Table 3: **Zero-Shot Transfer on Real World Tasks:** Evaluations follow settings in [71].

## 4.2 Real-World Environment

We next evaluate on 4 challenging tasks in an xArm Table Top environment, constructed following the real-world setup in [71]. Examples of these tasks are shown in Figure 4. The tasks involve tabletop manipulations specified by language commands (details in Appendix F).

**Training:** We train *System 2* (diffusion model) on the OpenX subset, followed by optional fine-tuning on 10 videos per task collected in the same real-world environment. We replicate the AVDC baseline [7] by training under identical conditions. All other baselines are implemented following the settings used in [71]. For *System 1*, we construct a hand-crafted mapping function based on [7, 71] (details in Appendix F).

**Evaluation:** We follow evaluation settings identical to [71], evaluating each policy across 4 tasks with a fixed camera view and 20 randomized trials per task. Each trial uses different initial positions of the objects present in the environment.

**Results:** We present results in Tables 2 and 3 to highlight the strong performance of LangToMo (baseline details in Appendix G). The difficulty of these tasks is apparent by the moderate results from recent methods like LLaRA [71]. Notably, despite relying on heuristic-based hand-crafted mappings in *System 1*, LangToMo outperforms several state-of-the-art baselines such as RT-2 [61] and LLaRA [71], all without requiring action trajectory labels during training. Our framework learns directly from videos paired with natural language captions, showing the promise of this direction.

## 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 4.

**System 2 Input Conditioning & Pretraining:** Removing visual ("Img"), language ("Lang"), or previous flow ("Prev Flow") conditional inputs to the diffusion model significantly reduces performance, highlighting importance of each conditioning signal. On the other hand, removing diffusion

Figure 4: **Real World Tasks:** We illustrate the four real-world tasks following LLaRA [71]. 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 2).

Table 4: **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 |
|-----|------|-----------|-----|---------|
| ✓ | ✓ | ✓ | ✓ | 53.6 |
| ✓ | ✓ | ✓ | ✗ | 53.1 |
| ✓ | ✓ | ✗ | ✗ | 50.2 |
| ✓ | ✗ | ✗ | ✗ | 39.7 |
| ✗ | ✓ | ✗ | ✗ | 15.4 |

| Method | Overall |
|--------|---------|
| Ours (default) | 53.6 |
| No diffusion | 16.2 |
| CA instead of concat | 15.8 |
| Sys-1 & 2 same freq | 48.7 |
| Only learned Sys-1 | 15.8 |

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. Modifying conditioning strategy to cross-attention ("CA instead of concat") also degrades performance. 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 necessity of our two-stage architecture. See Appendix H 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 *System 1*, using either learned or hand-crafted mappings tailored to specific embodiments. 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 downstream task. 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.

**Contributions**

KR led the project formulating the initial idea, coding the implementation, and performing most experiments. XL proposed several design choices of the approach, built the initial setup for real world experiments, and discussed all aspects of the project. CM contributed to ideas on experiment design, performed several real world experiments, and discussed most aspects of the project. JP helped setup human and robot demonstrations in real world, supported data collection and evaluations, and discussed most aspects of the project. MR organized the project, set the research direction, and discussed all aspects of the project idea, scope, and implementation.
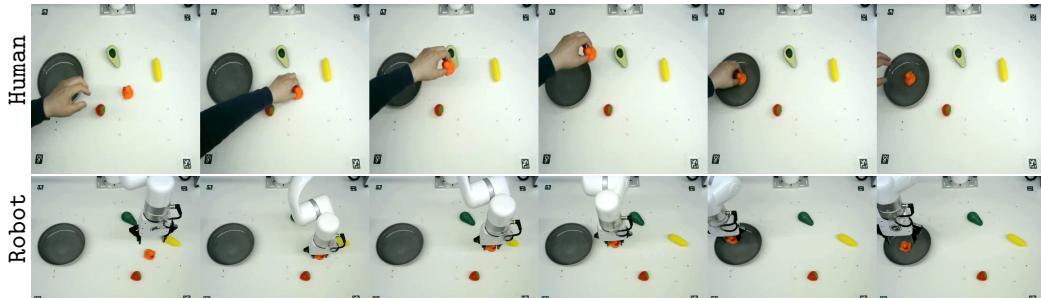
# Appendix

## A  Additional Experimental Results

We first present additional results on our real world environment, focused on highlighting the usefulness of human demonstrations for our method. A key benefit of our pixel motion based control (similar to prior work AVDC [7]) is the ability to learn from human demonstrations directly (with no requirement for keypoint based remapping or other dense annotations). We investigate this aspect of our proposed LangToMo first, presenting results in Table 5. Results indicate clear usefulness of incorporating human demonstrations in addition to robot demonstrations, as well as the ability to learn from human demonstrations directly. We illustrate some examples of human and robot demonstrations used for training in Figure 5.

| Method | Data | T1 | T2 | T3 | T4 | Average |
|---|---|---|---|---|---|---|
| AVDC | RD | 10 | 20 | 0 | 0 | 15.0 |
| LTM-H (ours) | RD | 80 | 70 | 65 | 60 | 68.8 |
| LTM-H (ours) | HD | 40 | 35 | 40 | 30 | 36.3 |
| LTM-H (ours) | RD+HD | 80 | 75 | 65 | 65 | 71.3 |

Table 5: **Extended Results on Real World Environment:** We evaluate the impact of using human demonstrations (HD) in addition to robot demonstrations (RD) as training data for our System 2 diffusion model. The standard setting following prior work is training on RD. AVDC trained on RD is provided as a baseline. Results indicate that training our method on HD alone performs reasonably, while using HD along with RD fortraining boosts performance further. RD here refers to human controlled robot demonstrations while HD refers to human controlled human demonstrations (e.g. human using their own human hand to move an object) See Figure 5 for examples.



Figure 5: **Human and Robotic Demonstrations:** We visualize frames from videos of two sample demonstrations on our real world environment. These human (top) and robot (bottom) demonstrations can both be used to fine-tune our System-2 diffusion model, highlighting a unique aspect of our hierarchical LangToMo approach. Both examples use the common caption of `"Pick up the rubber duck and place on the bowl."`

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 [7], we evaluate on the iThor benchmark and present results in Table 6. Results indite clear improvements of our proposed LangToMo over naive baselines and prior work AVDC [7].

## B  Relative Pixel Motion

A key design choice in our formulation is to represent pixel motion with respect to the current frame ($x_t$), rather than the previous frame ($x_{t+1}$) or some other frame. This aligns with the structure of our conditional diffusion model, which receives $x_t$ as a secondary conditioning input. Predicting the transformation from $x_t$ 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 $x_{t-1}$ or some other different frame

| 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 |

Table 6: **Results on iThor Benchmark:** We follow the iThor dataset based evalution setup used in AVDC paper 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.

would require indirect reasoning over a non-visible state, introducing additional complexity. Hence our approach is to represent past pixel motion (e.g. $x_{t-1}$ to $x_t$) as $x_t$ to $x_{t-1}$ 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 [15]. 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.

## C  Language Embedding Model

For the language embedding model, we employ the Universal Sentence Encoder (USE), a pre-trained model from [81]. 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 [62], 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.

## D  Diffusion Model Details

In our diffusion model training, input noising is applied by adding Gaussian noise to the target motion data (following standard settings [42]). The image condition input and the previous flow are not subject to this noising. The previous flow is corruption 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.

## E  Hand-Crafted Mapping Functions

**Synthetic Environments:** We follow the formulation of [7] 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. [7] for further details.

**Real World Environments:** Following Li et al. [71], 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 position for a manipulation task following this setting, our position to action vector conversion is identical to [71].

## F   Real World Experiments

We perform four types of real world experiments as illustrated in Figure 4. The language instructions for the four tasks 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.`

We select these following Li et al. [71] to ensure fair comparisons to prior works.

## G   Baseline Details

Our key baselines are from AVDC [7] and LLaRA [71]. 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).

## H   Detailed Ablations

We discuss our ablations in Table 4 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 (15.4%), 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 regressor 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 pre-generated optical flows ("Only learned Sys-1") performs poorly (15.8%), demonstrating that direct action generation without intermediate motion representation is insufficient for generalization.

# References

[1] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *Conference on Robot Learning*, 2022.

[2] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. 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.

[3] J. Zheng, J. Li, D. Liu, Y. Zheng, Z. Wang, Z. Ou, Y. Liu, J. Liu, Y.-Q. Zhang, and X. Zhan. Universal actions for enhanced embodied foundation models. *ArXiv*, abs/2501.10105, 2025. URL https://api.semanticscholar.org/CorpusID:275606605.

[4] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning Universal Policies via Text-Guided Video Generation. *arXiv:2302.00111*, 2023.

[5] X. Gu, C. Wen, J. Song, and Y. Gao. Seer: Language instructed video prediction with latent diffusion models. *ArXiv*, abs/2303.14897, 2023. URL https://api.semanticscholar.org/CorpusID:257766959.

[6] K. Black, M. Nakamoto, P. Atreya, H. R. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *ArXiv*, abs/2310.10639, 2023. URL https://api.semanticscholar.org/CorpusID:264172455.

[7] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. Tenenbaum. Learning to act from actionless videos through dense correspondences. *ArXiv*, abs/2310.08576, 2023.

[8] D. Kahneman. Thinking, fast and slow, 2011.

[9] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. Rt-h: Action hierarchies using language. *ArXiv*, abs/2403.01823, 2024. URL https://api.semanticscholar.org/CorpusID:268249108.

[10] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. $\pi 0$: A vision-language-action flow model for general robot control. *ArXiv*, abs/2410.24164, 2024. URL https://api.semanticscholar.org/CorpusID:273811174.

[11] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai, A. Li-Bell, D. Driess, L. Groom, S. Levine, and C. Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *ArXiv*, abs/2502.19417, 2025. URL https://api.semanticscholar.org/CorpusID:276618098.

[12] Nvidia, J. Bjorck, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *ArXiv*, abs/2503.14734, 2025.

[13] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. R. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi 0.5$: a vision-language-action model with open-world generalization, 2025. URL https://api.semanticscholar.org/CorpusID:277993634.

[14] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020.

[15] J. Liang, Y. Fan, K. Zhang, R. Timofte, L. van Gool, and R. Ranjan. Movideo: Motion-aware video generation with diffusion model. In *European Conference on Computer Vision*, 2024. URL https://api.semanticscholar.org/CorpusID:273232410.

[16] M. Koroglu, H. Caselles-Dupr'e, G. J. Sanmiguel, and M. Cord. Onlyflow: Optical flow based motion conditioning for video diffusion models, 2024.

[17] S. Sudhakar, R. Liu, B. V. Hoorick, C. Vondrick, and R. Zemel. Controlling the world by sleight of hand. *ArXiv*, abs/2408.07147, 2024.

[18] M. Shridhar, Y. L. Lo, and S. James. Generative image as action models. *ArXiv*, abs/2407.07875, 2024.

[19] W. Huang, C. Wang, Y. Li, R. Zhang, and F.-F. Li. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *ArXiv*, 2024.

[20] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos, 2025.

[21] J. Lee and M. S. Ryoo. Learning Robot Activities from First-Person Human Videos Using Convolutional Future Regression. In *CVPRW*, 2017.

[22] C. Finn and S. Levine. Deep Visual Foresight for Planning Robot Motion. In *IEEE International Conference on Robotics and Automation*, 2017.

[23] S.-H. Sun, H. Noh, S. Somasundaram, and J. Lim. Neural program synthesis from diverse demonstration videos. In *International Conference on Machine Learning*, 2018.

[24] T. Kurutach, A. Tamar, G. Yang, S. J. Russell, and P. Abbeel. Learning Plannable Representations with Causal InfoGAN. In *Neural Information Processing Systems*, 2018.

[25] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto. The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Robotics: Science and Systems*, 2022.

[26] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2Robot: Learning Manipulation Concepts from Instructions and Human Demonstrations. *IJRR*, 2021.

[27] A. S. Chen, S. Nair, and C. Finn. Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos. In *Robotics: Science and Systems*, 2021.

[28] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. In *Robotics: Science and Systems*, 2022.

[29] P. Sharma, D. Pathak, and A. Gupta. Third-person visual imitation learning via decoupled hierarchical controller. In *Neural Information Processing Systems*, 2019.

[30] A. Sivakumar, K. Shaw, and D. Pathak. Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on Youtube. In *Robotics: Science and Systems*, 2022.

[31] Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *ArXiv*, abs/2412.14803, 2024.

[32] G. C. de Croon, C. de Wagter, and T. 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.

[33] K. Lee, J. Gibson, and E. 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.

[34] Y. Hu, Y. Zhang, Y. Song, Y. Deng, F. Yu, L. Zhang, W. Lin, D. Zou, and W. 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.

[35] M. Argus, L. Hermann, J. Long, and T. Brox. Flowcontrol: Optical flow based visual servoing. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7534–7541, 2020. URL https://api.semanticscholar.org/CorpusID:220280145.

[36] K. G. Götz. Flight control in drosophila by visual perception of motion. *Kybernetik*, 4:199–208, 1968. URL https://api.semanticscholar.org/CorpusID:24070951.

[37] G. Arnold. Rheotropism in fishes. *Biological Reviews*, 49, 1974. URL https://api.semanticscholar.org/CorpusID:30755969.

[38] E. Baird, N. Boeddeker, and M. 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.

[39] I. G. Ros and A. 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.

[40] T. Han, W. Xie, and A. Zisserman. Self-supervised co-training for video representation learning. *ArXiv*, abs/2010.09709, 2020. URL https://api.semanticscholar.org/CorpusID:224703413.

[41] Y. Sharma, Y. Zhu, C. Russell, and T. Brox. Pixel-level correspondence for self-supervised learning from video. *ArXiv*, abs/2207.03866, 2022. URL https://api.semanticscholar.org/CorpusID:250407930.

[42] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[43] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video Diffusion Models. In *Neural Information Processing Systems*, 2022.

[44] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *preprint*, 2022. [arxiv:2204.06125].

[45] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[46] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman. Make-a-video: Text-to-video generation without text-video data, 2022.

[47] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual description, 2022.

[48] S. Ge, T. Hayes, H. Yang, X. Yin, G. Pang, D. Jacobs, J.-B. Huang, and D. Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer, 2022.

[49] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu. Multi-concept customization of text-to-image diffusion, 2023.

[50] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

[51] Z. Ren, Z. Pan, X. Zhou, and L. Kang. Diffusion motion: Generate text-guided 3d human motion by diffusion model. *arXiv preprint arXiv:2210.12315*, 2022.

[52] X. Chen, Y. Li, Z. Li, Z. Wang, L. Wang, and C. Qian. Moddm: Text-to-motion synthesis using discrete diffusion model. *arXiv preprint arXiv:2308.06240*, 2023.

[53] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning*, 2022.

[54] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. Grathwohl. Reduce, Reuse, Recycle: Compositional Generation with Energy-Based Diffusion Models and MCMC. In *International Conference on Machine Learning*, 2023.

[55] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton. StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects. In *Robotics: Science and Systems*, 2023.

[56] H.-C. Wang, S.-F. Chen, and S.-H. Sun. Diffusion Model-Augmented Behavioral Cloning. *arXiv:2302.13335*, 2023.

[57] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *ArXiv*, abs/2303.04137, 2023.

[58] S. Saxena, C. Herrmann, J. Hur, A. Kar, M. Norouzi, D. Sun, and D. J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *ArXiv*, abs/2306.01923, 2023.

[59] A. Luo, X. Li, F. Yang, J. Liu, H. Fan, and S. Liu. Flowdiffuser: Advancing optical flow estimation with diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19167–19176, 2024.

[60] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *Robotics science and systems (RSS)*, 2023.

[61] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[62] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[63] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and F. de Nando. A generalist agent. In *Trans. on Machine Learning Research*, 2022.

[64] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

[65] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Robotics science and systems (RSS)*, Delft, Netherlands, 2024.

[66] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[67] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[68] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

[69] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024.

[70] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

[71] X. Li, C. Mata, J. S. Park, K. Kahatapitiya, Y. S. Jang, J. Shang, K. Ranasinghe, R. Burgert, M. Cai, Y. J. Lee, and M. S. Ryoo. Llara: Supercharging robot learning data for vision-language policy. *ArXiv*, abs/2406.20095, 2024.

[72] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning*, 2024. URL https://api.semanticscholar.org/CorpusID:271097636.

[73] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *ArXiv*, abs/2412.15109, 2024. URL https://api.semanticscholar.org/CorpusID:274859727.

[74] Y. Jeong, J. Chun, S. Cha, and T. Kim. Object-centric world model for language-guided manipulation. *ArXiv*, abs/2503.06170, 2025. URL https://api.semanticscholar.org/CorpusID:276903201.

[75] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao. GMFlow: Learning Optical Flow via Global Matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[76] P. Liu, M. Lyu, I. King, and J. Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019.

[77] P. Dhariwal and A. Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Neural Information Processing Systems*, 2021.

[78] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[79] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019.

[80] K. Song, B. Chen, M. Simchowitz, Y. Du, R. Tedrake, and V. Sitzmann. History-guided video diffusion, 2025.

[81] D. M. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *ArXiv*, 2018.

[82] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Conference on Robot Learning*, 2019.

[83] OpenAI. Gpt-4 technical report, 2023.