Learning to Act Anywhere with Task-centric Latent Actions

Qingwen Bu[1,2], Yanting Yang[2], Jisong Cai[2], Shenyuan Gao[2], Guanghui Ren[3], Maoqing Yao[3], Ping Luo[1,2] and Hongyang Li[1,2]

[1] The University of Hong Kong [2] OpenDriveLab [3] AgiBot

Code: https://github.com/OpenDriveLab/UniVLA

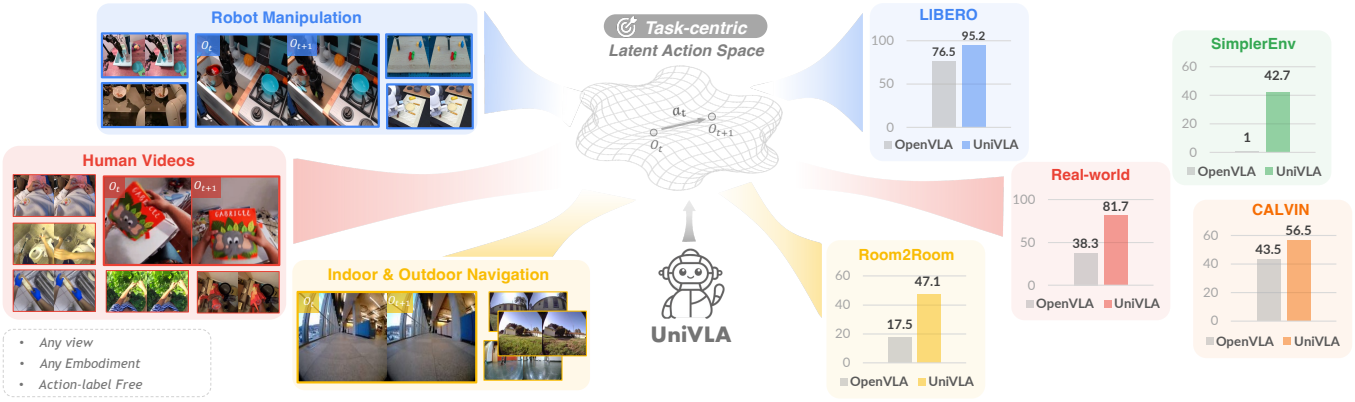

Fig. 1: We introduce **UniVLA**, a unified vision-language-action (VLA) framework that enables policy learning across different environments. By deriving task-centric latent actions in an unsupervised manner, UniVLA can leverage data from arbitrary embodiments and perspectives without action labels. After large-scale pretraining from videos, UniVLA develops a cross-embodiment generalist policy that can be readily deployed across various robots by learning an action decoding with minimal cost. Compared to OpenVLA [39], UniVLA exhibits unanimous improvement on multiple manipulation and navigation tasks.

***Abstract*—A generalist robot should perform effectively across various environments. However, most existing approaches heavily rely on scaling action-annotated data to enhance their capabilities. Consequently, they are often limited to single physical specification and struggle to learn transferable knowledge across different embodiments and environments. To confront these limitations, we propose UniVLA, a new framework for learning cross-embodiment vision-language-action (VLA) policies. Our key innovation is to derive task-centric action representations from videos with a latent action model. This enables us to exploit extensive data across a wide spectrum of embodiments and perspectives. To mitigate the effect of task-irrelevant dynamics, we incorporate language instructions and establish a latent action model within the DINO feature space. Learned from internet-scale videos, the generalist policy can be deployed to various robots through efficient latent action decoding. We obtain state-of-the-art results across multiple manipulation and navigation benchmarks, as well as real-robot deployments. UniVLA achieves superior performance over OpenVLA with less than 1/20 of pretraining compute and 1/10 of downstream data. Continuous performance improvements are observed as heterogeneous data, even including human videos, are incorporated into the training pipeline. The results underscore UniVLA's potential to facilitate scalable and efficient robot policy learning.**

I. Introduction

Empowered by the emergence of large-scale robotic datasets [78, 63, 38, 1], robot policies based on vision-language-action models (VLA) have made encouraging strides recently [10, 28, 39]. However, they typically rely on ground-truth action labels for supervision, which limits their scalability in utilizing internet-scale data from diverse environments. Furthermore, the heterogeneity of action and observation spaces across different embodiments (*e.g.*, Franka, WidowX, and even human hands) and tasks (*e.g.*, manipulation and navigation) poses a significant challenge to effective knowledge transfer. This raises a crucial question: could we learn a *unified action representation* that enables the generalist policy to plan effectively, unlocking the potential of internet-scale videos and facilitating knowledge transfer *across different embodiments and environments*?

To address these challenges, we propose UniVLA, a generalist policy learning framework that enables scalable and efficient planning across various embodiments and environments. Much like large language models (LLMs) learn cross-lingual shared knowledge [22, 18], we aim to construct a unified action space that facilitates knowledge transfer across video data, including various robot demonstrations and egocentric human videos. Our recipe for generalist policy consists of three key stages: **1) Task-centric Latent Action Learning**, where we extract task-relevant action representations from massive cross-embodiment videos in an unsupervised manner. This is achieved by discretizing latent actions from the inverse dynamics of paired frames using a VQ-VAE [76]. **2) Next-**

latent action prediction, where we train an auto-regressive vision-language model with discretized latent action tokens, endowing it with embodiment-agnostic planning capabilities. **3) Latents decoding**, where we decode latent plans into physical behaviors and specialize the pretrained generalist policy for deployment in unseen tasks efficiently.

While recent studies [87, 16] have investigated the viability of learning latent actions from web-scale videos, they suffer from a critical limitation: their naive reconstruction-based objectives often capture task-irrelevant dynamics, such as movements of non-ego agents or unpredictable camera shifts. These noisy representations hinder policy pretraining by introducing distractions unrelated to the task. To address this, we leverage pre-trained DINOv2 features [62] to extract patch-level representations from pixels, providing both spatial and object-centric priors that better capture task-relevant information. By using the readily available language instructions as conditions, we further disentangle movements into two complementary action representations, one of which explicitly represents task-centric actions.

UniVLA achieves state-of-the-art performance across multiple manipulation benchmarks and navigation tasks, outperforming OpenVLA [39] by a significant margin while requiring merely 1/20 of the pretraining cost (in GPU hours). This efficiency stems from its task-centric latent action space, which decouples task-relevant dynamics from extraneous visual changes. Our action representation not only reduces computational overhead but also enables efficient scaling - as dataset size grows, UniVLA's performance improves, effectively leveraging cross-embodiment, cross-view robot datasets and even unlabeled human videos to expand its pretraining corpus and extract transferable knowledge. Remarkably, when pretrained solely on the Bridge-V2 dataset [78], UniVLA surpasses OpenVLA and LAPA trained on the larger Open X-Embodiment [63] dataset, underscoring its ability to distill transferrable knowledge from *limited* data.

In addition, we employ a lightweight decoder with only 10.8M parameters to translate latent actions into executable trajectories, significantly reducing the need for extensive fine-tuning. This design leverages the compact and informative nature of the task-centric latent action space, enabling UniVLA to adapt efficiently to diverse tasks and embodiments with minimal downstream data. Our comprehensive evaluation, spanning manipulation, navigation, and real-world deployment, underscores the framework's efficiency, scalability, and generalizability, positioning it as a promising pathway toward next-generation generalist robotic policies.

In summary, our main **contributions** are three-folds:

- We propose UniVLA, a recipe towards generalist policy by planning in a unified, embodiment-agnostic action space, enabling scalable and efficient decision-making by learning from web-scale videos.
- We introduce a novel approach for extracting task-relevant latent actions from cross-embodiment videos, decoupling task-centric dynamics from irrelevant visual changes. Both qualitative and quantitative experiments highlight its merits and advantages over existing works.
- UniVLA achieves state-of-the-art performance on multiple benchmarks and real-robot tests, achieving an 18.5% increase in success rate over OpenVLA on the LIBERO [48] benchmark, 29.6% in navigation tasks [4], and a 36.7% improvement in real-world deployments.

II. Related Work

A. Vision-language-action Models

Building on the success of pretrained vision foundation models, large language models (LLMs), and vision-language models (VLMs), VLAs have been introduced to process multimodal inputs—visual observations and language instructions—and generate robotic actions for completing embodied tasks. RT-1 [11] and Octo [28] employ a transformer-based policy that integrates diverse data, including robot trajectories across various tasks, objects, environments, and embodiments. In contrast, some prior works [10, 39, 46] leverage pretrained VLMs to generate robotic actions by tapping into world knowledge from large-scale vision-language datasets. For instance, RT-2 [10] and OpenVLA [39] treat actions as tokens within the language model's vocabulary, while RoboFlamingo [46] introduces an additional policy head for action prediction. Building on these generalist policies, RoboDual [13] proposes a synergistic dual-system that combines the strengths of both generalist and specialist policy. Other works incorporate goal image [9] or video [24, 82, 14] prediction tasks to generate valid, executable plans conditioned on language instructions, with these visual cues subsequently guiding the policy in action generation. However, these methods heavily rely on interactive data with ground-truth action labels, which significantly restricts the scalability of VLAs. In contrast, our approach unlocks the potential of internet-scale, action-free videos by learning a unified latent action representation from visual changes, independent of action labels.

B. Cross-embodiment Learning

Training a general-purpose robot policy is challenging due to the diversity in camera perspectives, proprioceptive inputs, joint configurations, action spaces, and control frequencies across robotic systems. Early approach [86] focused on aligning action space manually between navigation and manipulation but were limited to wrist cameras in manipulation. Recent transformer-based approaches [28, 23] address these challenges by accommodating variable observations and actions, with CrossFormer [23] co-training across four distinct action spaces without imposing constraints on observation spaces or requiring explicit action-space alignment. Flow representations, capturing future trajectories of query points in images or point clouds, have been widely explored for cross-embodiment learning [81, 88, 26, 83]. ATM [81] learns flow generation from human demonstrations, while Im2Flow2Act [83] predicts object flows from human videos without in-domain data. Meanwhile, object-centric representations [32, 8] offer an alternative approach, with SPOT [32] predicting object trajectories in SE(3) to decouple embodiment actions from sensory

inputs. Existing approaches demand extensive, diverse datasets to cover all possible state-transition patterns and need explicit annotations, leading to inefficient data utilization. Our method sets itself apart by using a discrete codebook to encode latent actions in an unsupervised manner. Our approach effectively filters out visual noise and achieves efficient information compression via vector quantization, thereby enhancing training efficiency and lessening the reliance on data diversity.

C. Latent Action Learning

Several prior works focus on learning variational auto-encoders [64, 76] on raw action trajectories to structure new action spaces, emphasizing compact latent representations that facilitate behavior generation and task adaptation, as seen in VQ-BeT [44] and Quest [59]. These methods are also adopted in reinforcement learning to accelerate convergence [3]. Recent works [79, 74] explore vector quantization as action space adapters to better integrate actions into large language models. However, a key limitation of these approaches is their reliance on ground-truth action labels, which limits their scalability.

To leverage broader video data, Genie [12] extracts latent actions via a causal latent action model, conditioning on next-frame prediction. Similarly, LAPO [70] and DynaMo [20] learn latent actions directly from visual data, bypassing methods using explicit action labels on in-domain manipulation tasks. LAPA [87] and IGOR [16] introduce unsupervised pretraining methods to teach VLAs discrete latent actions, aiming to transfer knowledge from human videos. However, these approaches encode all visual changes from raw pixels, capturing task-irrelevant dynamics such as camera shakiness, movements of other agents, or new object appearances, which ultimately degrade policy performance. We propose a novel training framework to decouple task-centric dynamics from irrelevant visual changes, structuring a more effective latent action space to enable robust policy planning.

III. Methodology

We develop three steps to implement UniVLA: **1)** (Sec. III-A) Leveraging language-based goal specifications, we extract inverse dynamics from extensive video datasets in an unsupervised manner, yielding a discretized set of task-centric latent actions that generalize across diverse embodiments and domains; **2)** (Sec. III-B) Based on this, we train an auto-regressive transformer-based vision-language-action model, which takes visual observations and task instructions as inputs to predict latent action tokens in a unified latent space; **3)** (Sec. III-C) To facilitate efficient adaptation to various robotic control systems, we introduce specialized policy heads that decode latent actions into executable control signals.

A. Task-centric Latent Action Learning

The first step establishes the foundational groundwork of our framework by generating the pseudo action labels (*i.e.*, latent action tokens), which serve as the basis for training our generalist policy in subsequent stages.

Latent action quantization. Fig. 2 illustrates the two-stage training pipeline and overall architecture of our latent action model. We start with a pair of consecutive video frames, denoted as $\{o_t, o_{t+k}\}$, separated by a frame interval k. To ensure a uniform time interval of approximately 1 second across diverse datasets, the frame interval is calibrated according to the recording frequency specific to each dataset. To derive latent actions from videos, our latent action model is constructed around an Inverse Dynamics Model (IDM) based encoder $\mathcal{I}(a_t|o_t, o_{t+k})$ and a Forward Dynamics Model (FDM) based decoder $\mathcal{F}(o_{t+k}|o_t, a_t)$. The encoder infers latent action given consecutive observations, and the decoder is trained to predict future observations given specified latent actions. We implement the encoder as a spatial-temporal transformer [84] with casual temporal masks, following Villegas et al. [77]. A group of learnable action tokens $a_q \in \mathbb{R}^{N\times d}$, with predefined dimension d, are concatenated sequentially to the video features to extract the dynamics.

To further compress the information and align it with the learning objective [66] of an auto-regressive transformer-based policy, we apply latent quantization to the action tokens. Quantized action tokens $a_z \in \mathcal{R}^{N\times d}$ are optimized with VQ-VAE [76] objective, with a codebook of $|C|$ vocabulary size. The decoder, implemented as a spatial transformer, is optimized to predict future frames utilizing only the quantized action tokens. We do not feed decoder with historical frames to prevent the model from over-relying on contextual information or merely memorizing the dataset.

While recent works [12, 27, 87] employs raw pixels for prediction, we observe that pixel-space prediction forces models to attend to noisy, task-irrelevant details (e.g., textures, lighting) [30]. This issue is amplified in web-scale and crowd-sourced video datasets [29], where uncontrolled capture conditions introduce further variability. Inspired by joint-embedding predictive architectures (JEPA) [5, 6, 96], we propose using DINOv2 [62] spatial patch features as semantically rich representations. Their object-centric and spatially aware properties make them ideal not only as inputs but also as prediction targets for latent action models. Our self-supervised objective minimizes the embedding reconstruction error: $\|\hat{O}_{t+k} - O_{t+k}\|^2$. We use $\{O_t, O_{t+k}\}$ to represent the DINOv2 feature of paired video frames $\{o_t, o_{t+k}\}$. The compact latent action must thus encode the transformation between observations to minimize prediction error.

Latent action decoupling. As discussed earlier, the actions of the robots are often entangled with irrelevant environmental variations in web-scale videos. To mitigate the unfavorable effect of task-irrelevant dynamics, we incorporate readily available language instructions into the first training stage of latent action model (Fig. 2 Left). The language inputs are encoded using a pretrained T5 text encoder [67] and serve as conditioning signals in the context for both the encoder and decoder. This process can be formally described as:

$$\begin{cases} \text{Encode:} & \hat{a}_{\mathcal{I}} = \mathcal{I}([O_t; O_{t+k}; a_{\mathcal{I}}; \ell]),\ \tilde{a}_{\mathcal{I}} = \mathbf{VQ}(\hat{a}_{\mathcal{I}}), \\ \text{Decode:} & \hat{O}_{t+k} = \mathcal{F}([O_t; \tilde{a}_{\mathcal{I}}; \ell]), \end{cases}$$

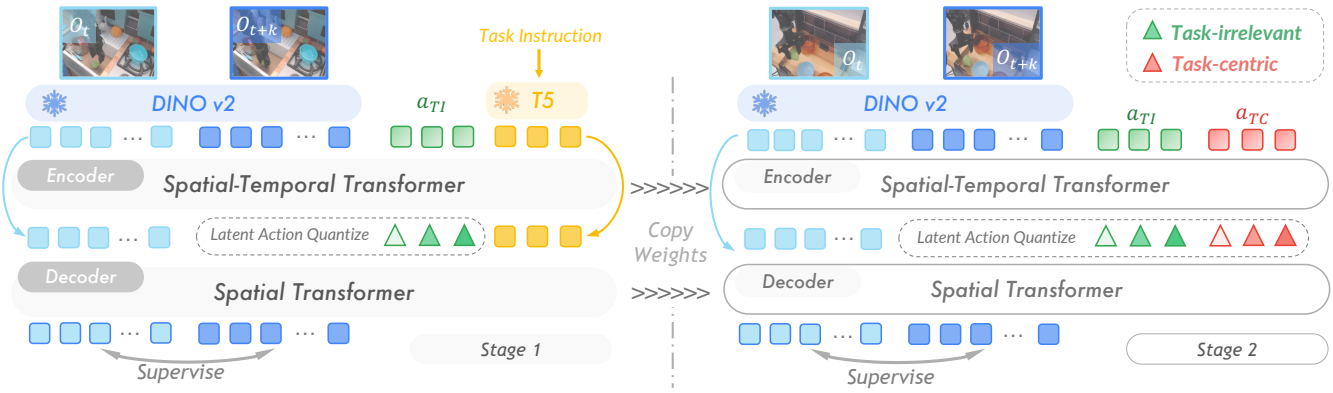

Fig. 2: **Task-centric latent action learning.** We propose a two-stage training framework aimed at disentangling task-centric visual dynamics and changes from extraneous factors. In **Stage 1**, task instruction embeddings, derived from a pre-trained T5 text encoder [67], are utilized as inputs to both the encoder and decoder. These embeddings provide task-relevant semantic information to enhance predictive accuracy. In **Stage 2**, a novel set of latent actions is introduced, specifically designed to replace the role of language and to capture task-centric dynamics from DINOv2-encoded features of video frames.

where $[;]$ denotes sequence-wise concatenation, $\mathbf{VQ}$ represents the codebook for vector quantized action representation, and ℓ is the instruction embedding from the T5 text encoder. Sending task instructions to the decoder provides high-level semantic guidance regarding the underlying actions. As a result, the quantized latent actions are optimized to encode only the environmental changes and visual details [89], omitting higher-level task-relevant information due to the constrained capacity of the codebook [2]. This stage establishes a set of latent actions that encapsulate *task-irrelevant* information, such as the emergence of new objects, movements of external agents, or camera-induced motion artifacts. These dynamics, while critical for grounding the model in the visual environment, are orthogonal to the specific objectives of the task.

Following this, we repurpose the task-irrelevant codebook and parameters of the latent action model trained in Stage 1 for the following stage (depicted in Fig. 2 Right), where the objective is to learn a new set of *task-centric* latent actions $\hat{a}_{TC}$ upon which the policy is trained. In this stage, the model extracts action information through:

$$\begin{cases} \text{Encode:} & \{\hat{a}_{TI},\ \hat{a}_{TC}\} = \mathcal{I}([O_t; O_{t+k}; a_{TI}; a_{TC}]), \\ & \tilde{a}_{TI} = \mathbf{VQ}(\hat{a}_{TI}),\ \tilde{a}_{TC} = \mathbf{VQ}_{TC}(\hat{a}_{TC}), \\ \text{Decode:} & \hat{O}_{t+k} = \mathcal{F}([O_t; \tilde{a}_{TI}; \tilde{a}_{TC}]), \end{cases}$$

where $\mathbf{VQ}_{TC}$ denotes the newly initialized codebook for learning task-centric dynamics. Building upon the acquired task-irrelevant representations, we freeze the corresponding codebook, enabling the model to focus on refining and specializing the new set of latent actions. This specialization facilitates the precise modeling of task-related dynamics, such as object manipulation or goal-directed motion trajectories. The explicit decoupling of latent action representations enhances our generalist policy's generalization capability across diverse environments and tasks. Compared to naive latent action learning approaches (*e.g.*, LAPA [87]), training exclusively on task-centric representations yields faster convergence while achieving robust performance, suggesting these latent actions are more informative for subsequent policy learning.

B. Pretraining of Generalist Policy

With the latent action model trained in the preceding step, we proceed to label any video frame o_t with latent actions a_z, given o_{t+k}. We then employ those labels to develop a generalist policy. To align with Kim et al. [39], our generalist policy is built upon the Prismatic-7B [37] vision-language model (VLM). The architecture integrates a fused visual encoder derived from SigLip [90] and DINOv2 [62], a projection layer to align visual embeddings with the language modality, and the LLaMA-2 large language model (LLM) [75]. Unlike prior LLM-based generalist policies (*i.e.,* RT-2 [10] and OpenVLA [39]) that directly plan in low-level action spaces by mapping infrequently used words in the LLaMA tokenizer vocabulary to uniformly distributed action bins within $[-1, 1]$, we extend the vocabulary with $|C|$ special tokens, specifically $\{\texttt{ACT\_1},\ \texttt{ACT\_2},\ \texttt{ACT\_3},\ldots,\ \texttt{ACT\_C}\}$. Latent actions are projected into this vocabulary based on their indices in the action codebook. This approach preserves the original model architecture and training objectives of the VLM, fully leveraging its pretrained knowledge for transfer to robotic control tasks. Specifically, our policy model π_ϕ receives observation o_t, task instructions l and prefixes of latent action tokens $a_{z,<i}$, and is optimized to minimize the sum of next-latent-action negative log-probabilities:

$$\mathcal{L} = \mathbb{E}_{o_t, l, a_{z,<i}} \left[-\sum_{i=1}^{N} \log\ \pi_\phi(\hat{a}_{z,i} = a_{z,i} \mid o_t, l, a_{z,<i}) \right],$$

where N represents the total length of action tokens. We set $N = 4$ for all our experiments. Moreover, empirical evidence indicates that a compressed action space (*e.g.*, reducing from 256^7 in OpenVLA [39] to 16^4 when $|C| = 16$) significantly accelerates model convergence. Our approach achieves competitive results with only 960 A100-hours of pretraining, a substantial reduction compared to the 21,500 A100-hours required for OpenVLA pretraining.

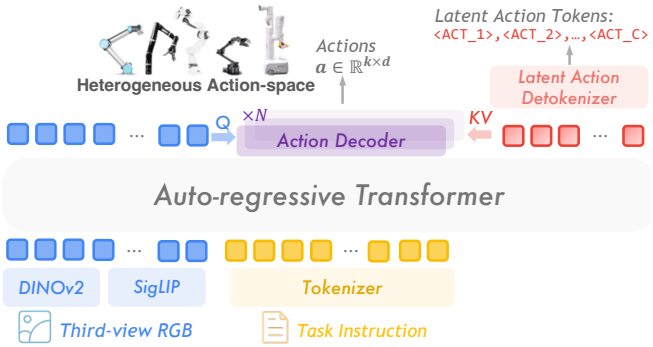

Fig. 3: **Architecture of the generalist policy.** Our policy architecture is founded on the Prismatic-7B Vision-Language Model (VLM) [37], which processes projected visual embeddings and tokenized task instructions as inputs to predict latent action tokens in an auto-regressive manner. To adapt to specific robotic systems, specialized action decoder heads are employed. These decoders leverage visual information to extract context-specific features from latent actions and subsequently translate them into executable control signals of robotic systems with heterogeneous action spaces.

By training our policy within a unified latent action space, the model capitalizes on transferable knowledge derived from cross-domain datasets. Unlike Yang et al. [86] which necessitates manual alignment of action spaces through visually similar egocentric motions, such as wrist camera movements in manipulation tasks and egocentric navigation, our method eliminates this requirement. Consequently, UniVLA expands the scope of utilizable datasets and enhances overall performance, demonstrating the efficacy of leveraging task-centric latent action representations for scalable policy learning.

C. Post-training for Deployment

Latent action decoding. During downstream adaptation, the pre-trained generalist policy maintains its embodiment-agnostic characteristics by predicting the next latent action during downstream adaptation. To bridge the gap between latent actions and executable behaviors, additional action decoders are employed (as depicted in Fig. 3). Specifically, the sequence of visual embeddings is first aggregated into a single token through multi-head attention pooling [43], which then functions as the query to extract information from the latent action embeddings. This process is formulated as:

$$\begin{cases} \text{Visual Embed.:} & E_v' = \mathcal{A}(Q = q_v, K = V = E_v), \\ \text{Action Embed.:} & E_a' = \mathcal{A}(Q = q_a + E_v', K = V = E_a), \end{cases}$$

where $\mathcal{A}$ represents multi-head attention, $\{E_v, E_a\}$ are visual and latent action embeddings from the last layer of VLM, and $\{q_v, q_a\}$ are randomly initialized queries to extract visual and action information respectively. The resultant action embedding E_a' is subsequently projected linearly into the desired action space of the target robotic system. Given that latent actions are designed to represent actions occurring within approximately a one-second interval (mentioned in Sec. III-A), they can be naturally decoded into action chunks [93]. The chunk size can be easily customized for specific embodiments to achieve smoother and more precise control.

In practice, we employ parameter-efficient fine-tuning using LoRA [33] to achieve efficient adaptation. With the addition of the action head comprising merely 12.6M parameters, the total number of trainable parameters is approximately 123M. The entire model is trained end-to-end, optimizing both the next-latent action prediction loss and the L1 loss between the ground-truth and predicted low-level actions.

Learn from history outputs. Historical observations have been demonstrated to play a critical role in enhancing sequential decision-making processes for robotic control [60, 42, 45]. However, directly providing large vision-language-action models with multiple historical observations introduces significant inference latency and results in redundant information within visual tokens [94, 45]. Drawing inspiration from the well-established Chain-of-Thought (CoT) reasoning paradigm [80] in large language models (LLMs), which generates intermediate reasoning steps to address complex tasks, we propose leveraging historical latent action outputs to facilitate decision-making in robotic control. Much like LLMs resolve questions step-by-step, we incorporate past actions into the input prompt at each timestep during rollouts. This establishes a feedback loop for the robot policy, enabling policy to learn from its own decisions and adapt to dynamic environments.

To operationalize this approach, we employ the latent action model to annotate actions extracted from historical frames. These annotated actions are then mapped into the LLaMA token vocabulary and appended to task instructions. During post-training, historical action inputs are integrated as inputs to endow the model with in-context learning capabilities. At inference time, one step of historical latent action (encoded as $N = 4$ tokens) is incorporated at each timestep, with the exception of the initial step. Empirical results demonstrate that this straightforward design improves model performance, particularly in long-horizon tasks (see Sec. IV-C).

IV. Evaluations

To demonstrate the performance of our proposed generalist policy, our evaluation framework assesses the capabilities of UniVLA across a diverse suite of benchmarks (including manipulation benchmarks: LIBERO [48], CALVIN [56], SimplerEnv [47], and a navigation benchmark: R2R [4]) and real-world scenarios. Additionally, we conduct latent action analysis to quantify the task-centric property, and perform ablation studies to explore critical design choices. With comprehensive evaluations, we mainly intend to investigate:

1) **Performance & Adaptability.** Can UniVLA successfully transfer the knowledge acquired during pretraining to novel embodiments and tasks and adapt efficiently? (See Sec. IV-A for manipulation performance and Sec. IV-A2 for adaptability to navigation.)

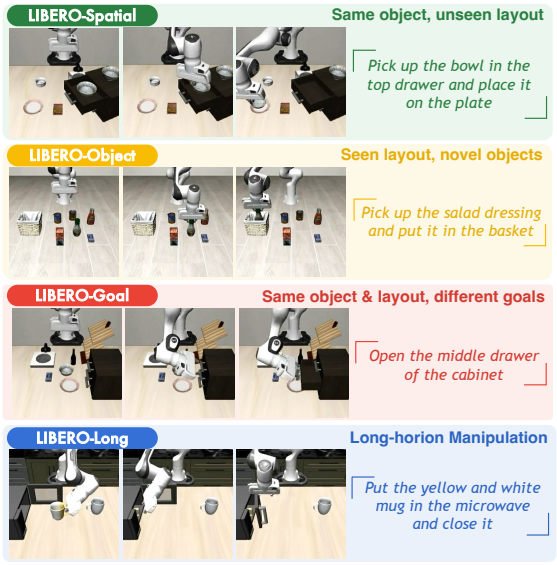

Fig. 4: Task setup on the LIBERO benchmark.

TABLE I: **Results on LIBERO benchmark across four evaluation suites.** Our proposed UniVLA exhibits superior performance across all benchmarked tasks compared to existing baseline methods, attributable to its enhanced knowledge transferability and generalization capabilities. Our model achieves state-of-the-art results despite being pre-trained exclusively on either the Bridge-V2 [78] dataset or action-free human video data (denoted as "Bridge" and "Human" respectively). †Methods use additional wrist-view camera inputs. *We reproduced results of LAPA using the Prismatic-7B VLM.

Method		Spatial	Object	Goal	Long	Average
LAPA* [87]		73.8	74.6	58.8	55.4	65.7
Diffusion Policy [17]		78.3	92.5	68.3	50.5	72.4
Octo [28]		78.9	85.7	84.6	51.1	75.1
MDT† [68]		78.5	87.5	73.5	64.8	76.1
OpenVLA [39]		84.7	88.4	79.2	53.7	76.5
MaIL† [35]		74.3	90.1	81.8	78.6	83.5
UniVLA (Ours)	Human	91.2	94.2	90.2	79.4	88.7
	Bridge	95.2	95.4	91.9	87.5	92.5
	Full	**96.5**	**96.8**	**95.6**	**92.0**	**95.2**

2) **Generalizability.** How does UniVLA generalize to unseen scenarios? (See Sec. IV-A3 for the analysis of its generalizability in novel settings.)
3) **Scalability.** Can UniVLA effectively utilize diverse data sources, even including human videos, and derive scalable benefits from the continuously expanding dataset? (See Sec. IV-C for data scalability analysis.)

A. Main Results

1) Manipulation Benchmark on LIBERO

Experiment setup. We pretrain our full latent action model on manipulation data, navigation data and human videos data, which are a subset of Open X-Embodiment (OpenX) dataset [63], GNM dataset [72], and human videos (Ego4D [29]) respectively. The pretraining details can be found in Appendix A1. The LIBERO benchmark [48] comprises four task suites specifically designed to facilitate research on lifelong learning in robotic manipulation. Our experiments exclusively focus on supervised fine-tuning within the target task suite, evaluating the performance of various policies trained through behavioral cloning on successful task demonstrations. As illustrated in Fig. 4, our experimental setup includes the following task suites, each consisting of 10 tasks with 50 human-teleoperated demonstrations per task:

1) **LIBERO-Spatial** requires the policy to infer spatial relationships to accurately place a bowl, evaluating the model's ability to reason about geometric configurations;
2) **LIBERO-Object** maintains identical scene layouts but introduces variations in object types, assessing the policy's capacity to generalize across object instances;
3) **LIBERO-Goal** retains consistent objects and layouts while assigning diverse task objectives, challenging the policy to exhibit goal-oriented behavior and adaptability;
4) **LIBERO-Long** focuses on long-horizon manipulation tasks involving multiple sub-goals, incorporating heterogeneous objects, layouts, and task sequences to evaluate the model's proficiency in complex, multi-step planning.

We adhere to the data processing pipeline introduced in OpenVLA [39] to exclude failure cases from the demonstration data used for training. UniVLA is trained on LIBERO-Long for 40k steps and other test suites for 30k steps, with a global batch size of 128. We only use third-person image and language instructions as inputs. Notably, *none* of the samples in LIBERO is included in the pretraining dataset of policy, and the training data for our latent action model, necessitating generalizability for both. In addition to presenting the results of our most performant model, which is pre-trained on the full dataset, we also provide results from models pre-trained exclusively on the Bridge-V2 [78] and human data, denoted as "Bridge" and "Human" in Tab. I, respectively. To minimize variance, all methods are evaluated over 500 trials per task suite (*i.e.,* 50 trials per task), with the reported performance reflecting the average success rate across three seeds.

Baselines. Our selected baseline models include the following five representative models, where OpenVLA and LAPA are more closely related to our method:

- **LAPA** [87] introduces an unsupervised framework for learning latent actions from unlabeled human videos.
- **Octo** [28] is a transformer-based policy trained on diverse robotic datasets, which employs a unified action representation to handle heterogeneous action spaces.
- **MDT** [68] leverages diffusion models to generate flexible action sequences conditioned by multimodal goals.
- **OpenVLA** [39] is a vision-language-action model that leverages large-scale pretraining on diverse datasets, including OpenX, to enable generalist robotic policies.
- **MaIL** [35] enhances imitation learning by incorporating selective state space models, which improve the efficiency and scalability of policy learning.

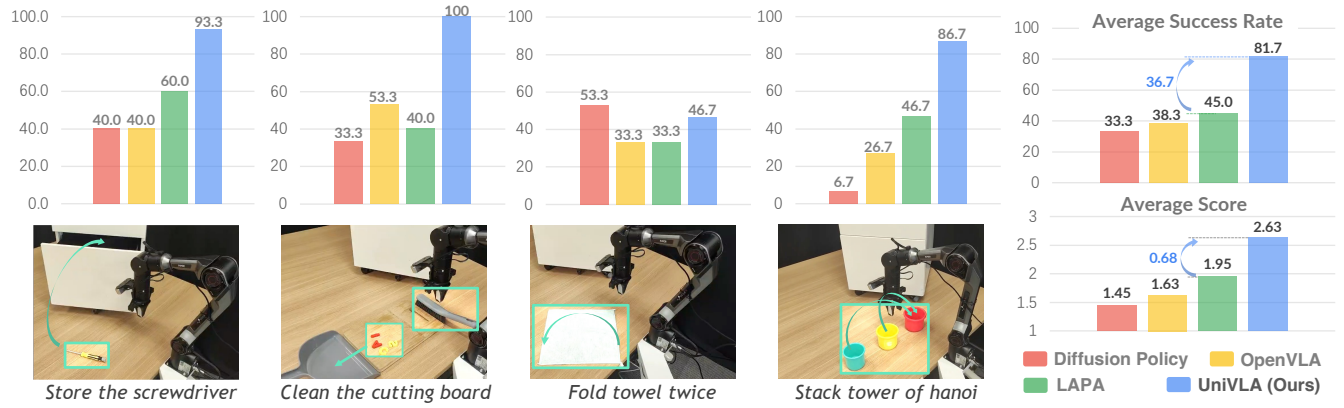

Fig. 5: **Real-world robot experiments.** We propose four different tasks: "Store the screwdriver", "Clean the cutting board", "Fold towel twice", and "Stack tower of hanoi", towards the evaluation of four axis of policy's capabilities. UniVLA outperforms previous state-of-the-art with an average elevation of 36.7% success rate and 0.68 average score across all tasks.

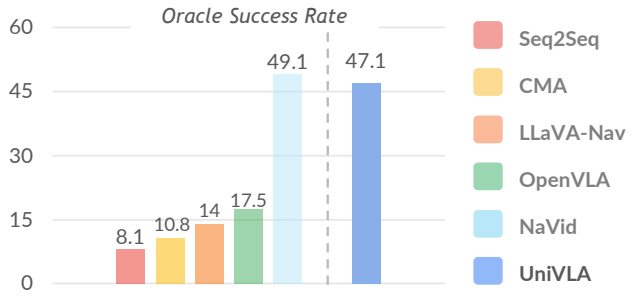

Fig. 6: **Oracle success rate on R2R in VLN-CE.** With only a single-frame RGB input, UniVLA demonstrates performance on par with NaVid, a navigation model that incorporates the entirety of historical observations, while markedly outperforming OpenVLA in success rate.

Results. The results presented in Tab. I demonstrate the exceptional performance of UniVLA across all four evaluation suites, significantly outperforming prior generalist policies such as OpenVLA, LAPA, and Octo. Notably, UniVLA achieves an average performance of 95.2% by pretraining on the full dataset, surpassing OpenVLA and LAPA by margins of 18.7% and 29.5% respectively. Despite being pretrained solely on the Bridge-V2 dataset, UniVLA attains 92.5% average performance, outperforming methods like MaIL (83.5%) and MDT (76.1%) that leverage additional wrist-view camera inputs. Pretraining our policy with human data outcompetes OpenVLA, which is trained with in-domain OpenX data, by a margin of 12.2%. In conclusion, UniVLA shows unparalleled knowledge transfer capability and establishes a new state-of-the-art on LIBERO benchmark. We provide additional results on CALVIN and SimplerEnv benchmark in Appendix B.

2) Navigation Benchmark on Room2Room

Experiment setup. In this experiment, we evaluate UniVLA on the VLN-CE benchmarks [41] to assess its performance on navigation tasks. These benchmarks offer a set of language-guided navigation tasks and continuous environments for executing low-level actions in reconstructed photorealistic indoor scenes. Specifically, we focus on the Room2Room (R2R) [4] task in VLN-CE, one of the most widely recognized benchmarks in vision-and-language navigation (VLN). All methods are trained on the 10,819 samples in the R2R training split and evaluated on the 1,839 samples in the R2R val-unseen split. We use the oracle success rate to evaluate navigation performance. An episode is considered successful if the agent arrives within 3 meters of the goal in the VLN-CE.

Baselines. To ensure a fair comparison with UniVLA, we evaluate RGB-only methods that operate without depth or odometry data, directly predicting low-level actions within the VLN-CE environments. Selected baselines are as follows:

- **Seq2Seq** [40] is a recurrent sequence-to-sequence policy that predicts actions from RGB observations.
- **CMA** [40] employs cross-modal attention to integrate instructions with RGB observations for action prediction.
- **LLaVA-Nav** is a modified version of LLaVA [49], co-finetuned with data proposed by NaVid [91], and encodes history using an observation-to-history technique.
- **OpenVLA** [39] is a vision-language-action model. We introduce several special tokens to tokenize navigation actions and finetune the model on the R2R training split.
- **NaVid** [91] is a video-based large vision-language model that encodes all historical RGB observations. It uses a pretrained vision encoder to encode visual observations and a pretrained LLM to predict actions.

Results. In Fig. 6, we report the oracle success rate for each method. UniVLA significantly outperforms Seq2Seq and CMA, increasing the oracle success rate from 8.10% to 47.1%. Given the high computational cost of prompting history in LLaVA-Nav, we refer to NaVid and present its results on a 100-episode subset of the VLN-CE R2R val-unseen split. UniVLA surpasses the oracle success rate of LLaVA-Nav by **33.1%** and OpenVLA by **29.6%**. Furthermore, UniVLA achieves an oracle success rate comparable to NaVid, which

TABLE II: **Generalizability evaluation.** UniVLA demonstrates superior performance across all evaluated tasks, showcasing its exceptional ability to generalize from high-level semantic comprehension to low-level visual robustness.

Method	Lightning Variation		Visual Distractor		Novel Object		Average ↑	
	Succ.	Score	Succ.	Score	Succ.	Score	Succ.	Score
Diffusion Policy [17]	20.0	0.60	26.7	0.80	26.7	0.67	24.4	0.69
OpenVLA [39]	13.3	0.93	20.0	0.73	26.7	1.27	20.0	0.98
LAPA [87]	26.7	1.60	6.7	0.6	53.3	1.87	28.9	1.36
UniVLA (Ours)	**66.7**	**2.33**	**53.3**	**2.40**	**86.7**	**2.73**	**68.9**	**2.49**

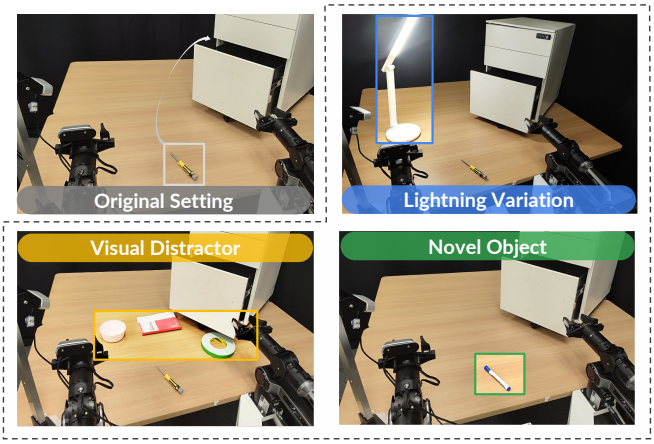

Fig. 7: **Setting on generalizability evaluations.** We evaluate the generalizability of policies in 3 different settings. **(a)** Lightning Variation: We dimmed the ambient light and applied strong lighting in a specified direction. **(b)** Visual Distractor: We added a bowl, notebook, and tape on the tabletop. **(c)** Novel Object: We replaced the object to be manipulated from a screwdriver to an unseen marker pen.

encodes all historical observations, while UniVLA conditions only on the current observation and historical latent action.

3) Real-world Robot Deployment

Experiment setup. All real-world experiments are conducted with a Piper arm from AgileX Robotics featuring a 7-DoF action space and a third-view Orbecc DABAI RGB-D camera, which we only utilize RGB images as input. To evaluate policies, we design a comprehensive set of tasks that span various dimensions of policy capabilities, including:

1) **Spatial Awareness:** Pick up the screwdriver to put it into the cabinet and close the door ("Store the screwdriver").
2) **Tool-usage and Nonprehensile Manipulation:** Pick up the broom and sweep the items on the cutting board into the dustpan ("Clean the cutting board").
3) **Deformable Objects Manipulation:** Fold the towel in half twice ("Fold towel twice").
4) **Semantic Understanding:** Stack the medium tower on top of the large one first, then stack the small one on top of the medium one.("Stack tower of hanoi")

For each task, we collect 20–80 trajectories, scaled according to task complexity, to finetune our model. To evaluate generalization comprehensively, we design experiments that span multiple axes of unseen scenarios, including lighting variations, visual distractors, and object generalization (see Fig. 7). Recognizing that success rate alone inadequately captures policy performance or distinguishes their capabilities, we introduce a step-wise scoring system. For each of the four tasks, we assign a maximum score of 3 points, reflecting the completion of distinct stages during task execution. Detailed scoring criteria, task setup and experiment results are provided in Appendix C.

Baselines. We choose Diffusion Policy [17], alongside generalist policies, OpenVLA [39] and LAPA [87] as our baselines. Diffusion Policy is trained in a single-task manner, whereas the generalist models are trained on all tasks simultaneously with instruction inputs. For a fair comparison, we reproduce LAPA with Prismatic-7B VLM [37] and action decoder heads, aligning its architecture with our method. This setup allows us to isolate and emphasize the contribution of our task-centric latent action space. Specific parameters and architectural details can be found in Appendix C.

Results. We plot task success rates in Fig. 5. The single-task Diffusion Policy (DP), optimized for trajectory fidelity and low-latency control, excels in tasks like towel folding, where success hinges on executing a fixed trajectory once the correct towel edge is selected. This specialization allows DP to achieve a higher success rate (53.3%) compared to UniVLA (46.7%) in this task. However, UniVLA achieves a higher step-wise score (2.47 vs. DP's 2.33, detailed in Appendix C), reflecting its ability to reliably complete intermediate stages (e.g., edge selection, partial folding) even when final execution falters—a critical advantage in dynamic real-world environments where partial progress is valuable.

This trade-off arises from UniVLA's generalist design: while DP's single-task training maximizes trajectory precision for specific workflows, it struggles in tasks requiring semantic reasoning (*e.g.*, stack tower of hanoi, where DP achieves only 6.7% success). In contrast, UniVLA demonstrates superior generalization and semantic understanding, achieving an unparalleled 86.7% success rate. This is further evidenced by a 93.3% success rate in scenarios requiring precise object manipulation and spatial reasoning (where the object is placed at varied poses and positions in "Store the screwdriver" task).

In addition, our method achieves a real-time, closed-loop inference frequency of 10Hz on an NVIDIA RTX 4090 GPU by planning in a compact latent action space, and allowing efficient action chunk prediction (we use a chunk size of 12

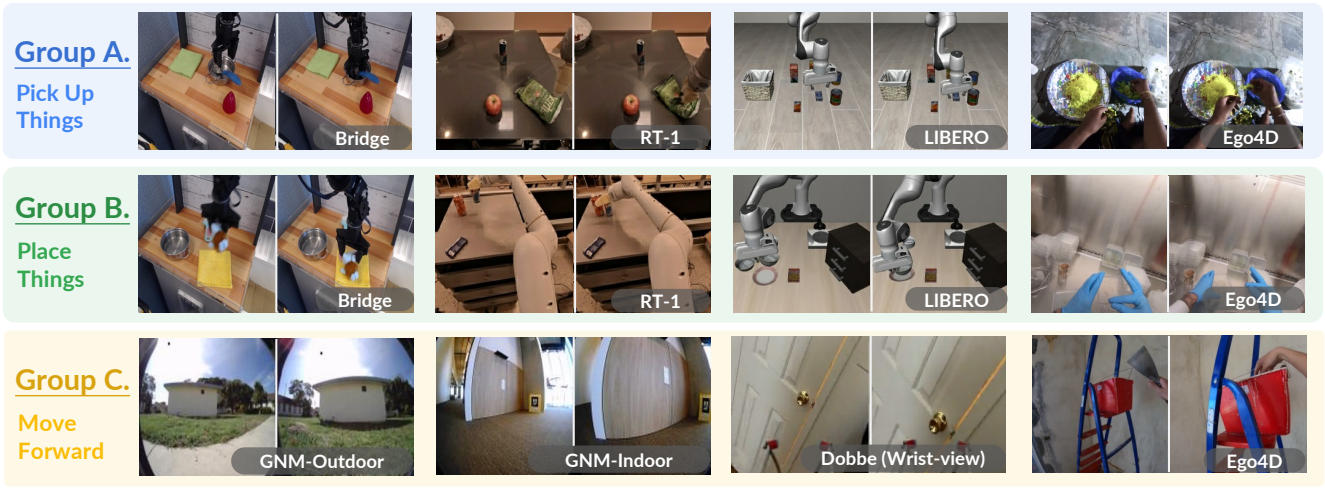

Fig. 8: **Latent action analysis.** We plot image pairs labeled with the same latent action from different sources of data and embodiments. Each group of latent actions exhibits semantic-consistent actions. More examples are in Appendix B.

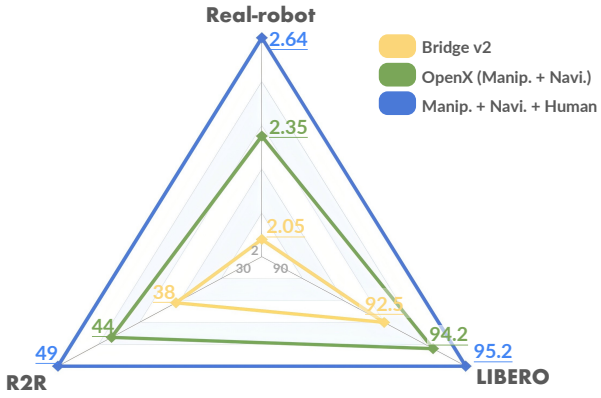

Fig. 9: **Data scalability.** UniVLA effectively expands its pretraining corpus by incorporating cross-embodiment data from OpenX and unlabeled human demonstrations, leading to continuously improved downstream performance.

in practice). OpenVLA, despite extensive training on large-scale robot datasets, suffers from execution stuttering due to inference latency (*e.g.*, 0.18s when predicting a single action step, 0.68s when predicting action chunks with size 4), resulting in poor real-world performance with only a 38.3% average success rate. In a nutshell, UniVLA outperforms LAPA, the second best policy, by **36.7%** in success rate and **0.68** in average score, demonstrating its real-world effectiveness and the advantages of our proposed task-centric latent action space.

Generalizability Analysis. We investigate the generalizability of policies from 3 different aspects, with the specific experiment setups shown in Fig. 7. The results in Tab. II highlight UniVLA's exceptional generalizability, significantly outperforming baseline methods in success rates and step-wise scores. It achieves a 66.7% success rate under varying lighting conditions, surpassing Diffusion Policy (20.0%), OpenVLA (13.3%), and LAPA (26.7%), demonstrating robustness to environmental change. In scenarios with visual distractors, policies that rely more on semantic information, such as LAPA and UniVLA, experience a relatively notable performance drop. In the novel object setting, we replaced the screwdriver with a marker and adjusted the language inputs for the generalist policy accordingly. This change had minimal impact on our policy as the success rate only drops by 6.6%. Overall, UniVLA achieves an average success rate of **68.9%** and an average score of **2.49**, significantly outperforming prior VLAs like LAPA (28.9%, 1.36) and OpenVLA (20.0%, 0.98). We also provide video demos in the supplementary material.

B. Discussion on Latent Action

Qualitative analysis. We investigate the cross-domain transferability of latent actions by visualizing image pairs from different data sources sharing the same latent action in Fig. 8. Each group of latent actions maps to semantically consistent behaviors across embodiments (*e.g.,* latent actions representing "`Pick up things`" in Group A). Notably, our latent action model, trained without any data from the LIBERO [48] dataset, generalizes effectively to label accurate actions in this unseen domain. Furthermore, LAM learns to align wrist-view observations in manipulation with ego-centric movements in navigation, as demonstrated in Group C, highlighting its ability to bridge diverse modalities and embodiments.

Quantitative analysis. To evaluate the effectiveness of our proposed dynamics decomposition approach for task-centric latent action learning, we assess the deployment performance of policies trained with labels derived from different latent actions. The results on LIBERO are shown in Tab. III. We pre-train policies using only human videos, which contain significant amounts of unpredictable motion, to amplify the advantages of our method. In comparison to the latent action construction approach introduced in Genie [12], which captures all visual changes, our method demonstrates clear superiority. Specifically, we achieve a 6.4% improvement in

TABLE III: **Performance on LIBERO using various latent actions.** We pretrain policies using different latent actions on Ego4D [29], which features human videos with diverse movements and task-irrelevant dynamics, to demonstrate our successful decoupling of task-centric dynamics. While task-irrelevant ones yield poor performance, task-centric latent action learning produces more meaningful action representations, ultimately achieving superior deployment success rates.

Latent Action	Spatial	Object	Goal	Long	Avg.
Genie [12]	89.8	92.8	77.2	69.6	82.3
Task-irrelevant	68.0	90.4	67.2	0.2	56.5
Task-centric	**91.2**	**94.2**	**90.2**	**79.4**	**88.7**

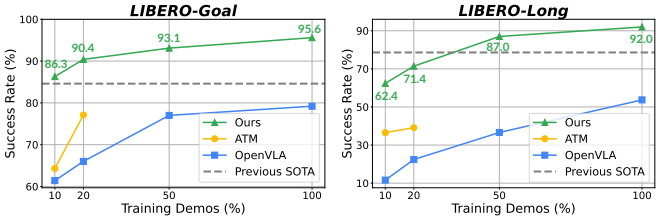

Fig. 10: **Data efficiency.** We present the success rate of UniVLA across varying dataset proportions (10%, 20%, 50%, and the full dataset). Our policy can be adapted to an unseen environment without requiring extensive expert demos for training, showing notable superiority over baselines.

average success rate, with substantial gains in LIBERO-Goal and LIBERO-Long (13% and 9.8% improvement, respectively). In contrast, latent actions that are task-irrelevant are poorly aligned with true actions, making it difficult for policies to infer them from observations and task instructions. This is reflected in both lower action token prediction accuracy during training and poorer inference performance. Notably, training with task-irrelevant latent actions results in near-zero success rates on the challenging LIBERO-Long benchmark.

C. More Ablations

Data scalability. We show how UniVLA evolves with the growing data scale and the incorporation of data from distinct domains in Fig. 9. Though UniVLA already sets a new state-of-the-art on LIBERO by pretraining only with Bridge-V2 [78]. Cross-embodiment data in OpenX [63] and Ego4D [29] further amplifies the average success rate by 2.0%. While the performance on the LIBERO benchmark appears to plateau, consistent performance improvements are observed across our challenging real-world test suites. In real-world evaluations, expanding the pretraining data to OpenX increases the average score by 0.3, compared to Bridge-only pretraining. Further incorporating human data, despite the absence of action labels and the substantial embodiment gap it introduces, yields an additional 0.28 increase. This trend of performance improvement is similarly observed in the R2R navigation benchmark, highlighting the scalability of our approach as it effectively leverages diverse data sources.

TABLE IV: **Ablations on decoder design.** "Auto-regressive" represents that we follow the approach of OpenVLA and LAPA, predicting actions sequentially over discretized action bins in an auto-regressive fashion. "w/o visual" indicates that visual embeddings are not utilized as query inputs for decoding latent actions, as depicted in Fig. 3. The proposed action decoder head, augmented by visual features, proves to be the most effective, yielding the highest results on all test suites.

Action Decoder	Spatial	Object	Goal	Long	Avg.
Auto-regressive	85.2	81.2	79.0	49.0	73.6
Ours w/o Visual	95.0	95.4	93.7	86.0	92.5
Ours	**96.5**	**96.8**	**95.6**	**92.0**	**95.2**

TABLE V: **Ablations on the use of history action.** Incorporating latent action outputs from previous steps as prompt inputs, despite its simplicity, enhances performance, particularly in long-horizon tasks, such as LIBERO-Long and R2R.

Prompt Input	LIBERO (Manip.) Goal	LIBERO (Manip.) Long	R2R (Navi.)
Instruction-only	95.0	88.1	30.6
w/ History Action	**95.6**	**92.0**	**47.1**

Data efficiency. The preceding section highlights UniVLA's scalability with respect to pretraining data, consistently enhancing its capabilities. We next explore its ability to adapt efficiently to unseen environments with minimal data, as detailed in Fig. 10. Specifically, we evaluate performance on the LIBERO-Goal and LIBERO-Long benchmarks using partial training data. UniVLA demonstrates superior data efficiency compared to prior generalist policy, such as OpenVLA [39], and explicit point prediction methods like ATM [81]. Notably, with only 10% of the demonstration data, UniVLA achieves a higher success rate on LIBERO-Goal (86.3% *vs.* 79.2%) than OpenVLA trained on the full dataset. Moreover, it sets a new state-of-the-art performance on both LIBERO-Goal and LIBERO-Long with only 10% and 50% of the training episodes, respectively. By planning within a unified latent action space, UniVLA maximizes pretraining knowledge, enabling highly efficient adaptation to new environments.

Latent action decoder. We compare our proposed action decoding scheme with the auto-regressive approach, which sequentially generates discretized actions as in OpenVLA and LAPA. As shown in Tab. IV, our method consistently achieves higher success rates across all test suites, with a striking 42.1% improvement in LIBERO-Long. Leveraging visual embeddings as queries enhances action decoding by reducing ambiguity in the multimodal distribution, yielding an additional 2.2% gain in average success rate.

Furthermore, as discussed in Sec. III-A, latent actions are designed to encapsulate dynamics over a one-second time horizon. Given this temporal structure, decoding latent actions as action chunks [93] is an intuitive choice, aligning the chunk size with the control frequency of the target embodiment. This

is achieved by simply expanding the output dimension of the final linear projection layer, while introducing negligible additional inference cost compared to the auto-regressive approach.

History latent actions. As detailed in Sec. III-C, we augment the instruction input with historical latent actions to enhance sequential decision-making. We evaluate the efficacy of this minimal architectural modification on manipulation and navigation tasks, with quantitative results in Tab. V. The approach proves particularly impactful in long-horizon scenarios: using only four input tokens (representing one latent action group) improves success rates by 16.5% (R2R) and 3.9% (LIBERO-Long). Extending the history horizon yields diminishing returns. Unlike methods requiring redundant multi-frame visual tokens for temporal context (*e.g.*, [91, 45]), our design provides compact historical guidance while enabling iterative policy refinement through self-referential outputs. This streamlined integration enhances contextual awareness without incurring unnecessary computational overhead.

V. Conclusion

In this work, we introduce UniVLA, a vision-language-action model that plans within a unified, task-centric latent action space, enabling efficient adaptation to novel robotic setups. Through extensive evaluations, we demonstrate that UniVLA establishes state-of-the-art performance across multiple manipulation and navigation benchmarks. The model also exhibits scalability with heterogeneous pretraining data to enhance its downstream performance, and remains highly adaptable even in data-limited scenarios. We aim for our work to pave the way for the next generation of generalist policies, capable of leveraging web-scale video data for training, regardless of embodiment gaps or the availability of action labels.

VI. Limitations and Future Work

Latent action design. While UniVLA advances generalist robotic policies, several limitations remain. The fixed granularity of the latent action and the predefined codebook size may not be optimal for all tasks or embodiments. Exploring adaptive mechanisms to dynamically adjust these based on environmental conditions could potentially improve performance. In addition, UniVLA is primarily evaluated on single-arm manipulation tasks. The action granularity represented by latent action tokens are relatively fixed within our framework. Extending the framework to dual-arm humanoid systems or dexterous hands could require more complex and finer-grained action space modeling. We leave this for future exploration.

Requirements on language annotation. Regarding language granularity, task-relevant latent actions are designed to encode ego-agent movements critical for task completion, while excluding non-ego dynamics (*e.g.*, steam rising from a kettle during "boiling water"). The majority of our dataset comprises fine-grained instructions that describe short-horizon actions rather than high-level goals. While more expressive language instructions could potentially reduce ambiguity in latent action learning, we want to emphasize that our approach enables scalable learning from instructions of varying granularity. Without any special handling of instruction, our method outperforms naive latent action learning approaches.

Integration with world model. The decoder of latent action model is essentially a world model, predicting future observations given latent actions. It can be conditioned on latent actions sampled by our policy on-the-fly and generate multiple corresponding visual plans. This opens the door to reference alignment [92] with reinforcement learning and test-time scaling through planning trees [25], where VLMs [54] or heuristic functions can be adopted as reward models.

In-context learning capability is critical for enhancing the performance ceiling of vision-language-action models. Given our finding that the proposed latent action model can extract transferable motion representations bridging human and robotic manipulations, we propose encoding human demonstration videos into a sequence of compact latent action embeddings, serving as in-context samples (conceptually, the latent action model functions as a video tokenizer). This approach enables zero-shot skill acquisition without additional fine-tuning. We will explore this direction in future work.

Acknowledgment

We thank Li Chen, Modi Shi, and Chengen Xie for their valuable feedback and fruitful discussions.

References

[1] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 1

[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017. 4

[3] Arthur Allshire, Roberto Martín-Martín, Charles Lin, Shawn Manuel, Silvio Savarese, and Animesh Garg. Laser: Learning a latent action space for efficient reinforcement learning. In *ICRA*, 2021. 3

[4] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 2, 5, 7

[5] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023. 3

[6] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. 3

[7] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. HYDRA: Hybrid robot actions for imitation learning. *arXiv preprint arXiv:2306.17237*, 2023. 16

[8] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *ICRA*, 2024. 2

[9] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. In *ICLR*, 2024. 2, 16

[10] 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. In *CoRL*, 2023. 1, 2, 4

[11] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *RSS*, 2023. 2, 16

[12] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *ICML*, 2024. 3, 9, 10

[13] Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001*, 2024. 2, 16

[14] Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. In *NeurIPS*, 2024. 2, 16

[15] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home. 16

[16] Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. IGOR: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024. 2, 3

[17] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023. 6, 8, 18

[18] A Conneau. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019. 1

[19] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022. 16

[20] Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. In *NeurIPS*, 2024. 3

[21] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. CLVR jaco play dataset, 2023. URL https://github.com/clvrai/clvr_jaco_play_dataset. 16

[22] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[23] Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling Cross-Embodied Learning: One policy for manipulation, navigation, locomotion and aviation. In *CoRL*, 2024. 2

[24] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *NeurIPS*, 2024. 2

[25] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, brian ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *ICLR*, 2024. 11

[26] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. FLIP: Flow-centric generative planning for general-purpose manipulation tasks. In *ICLR*, 2025. 2

[27] Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025. 3

[28] Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy. In *RSS*, 2024. 1, 2, 6

[29] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 3, 6, 10, 16

[30] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2019. 3

[31] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible real-world

benchmark for long-horizon complex manipulation. In *RSS*, 2023. 16

[32] Cheng-Chun Hsu, Bowen Wen, Jie Xu, Yashraj Narang, Xiaolong Wang, Yuke Zhu, Joydeep Biswas, and Stan Birchfield. SPOT: Se (3) pose trajectory diffusion for object-centric manipulation. *arXiv preprint arXiv:2411.00965*, 2024. 2

[33] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 5, 16, 17

[34] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *CoRL*, 2022. 16

[35] Xiaogang Jia, Qian Wang, Atalay Donat, Bowen Xing, Ge Li, Hongyi Zhou, Onur Celik, Denis Blessing, Rudolf Lioutikov, and Gerhard Neumann. MaIL: Improving imitation learning with selective state space models. In *CoRL*, 2024. 6

[36] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*, 2018. 16

[37] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. In *ICML*, 2024. 4, 5, 8

[38] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS*, 2024. 1, 16

[39] 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. In *CoRL*, 2024. 1, 2, 4, 6, 7, 8, 10, 16, 18

[40] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020. 7

[41] Jacob Krantz, Erik Wijmans, Arjun Majundar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *ECCV*, 2020. 7

[42] Hanna Kurniawati. Partially observable markov decision processes (pomdps) and robotics. *arXiv preprint arXiv:2107.07599*, 2021. 5

[43] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 5

[44] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. In *ICML*, 2024. 3

[45] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024. 5, 11, 17

[46] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *ICLR*, 2024. 2, 16

[47] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *CoRL*, 2024. 5, 16, 17

[48] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2024. 2, 5, 6, 9

[49] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2023. 7

[50] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *RSS*, 2023. 16

[51] Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *TRO*, 2023. 16

[52] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: a functional manipulation benchmark for generalizable robotic learning. *IJRR*, 2023. 16

[53] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *RA-L*, 2023. 16

[54] Yecheng Jason Ma, Joey Hejna, Ayzaan Wahid, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, et al. Vision language models are in-context value learners. In *ICLR*, 2025. 11

[55] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. RoboTurk: A crowdsourcing platform for robotic skill learning through imitation. In *CoRL*, 2018. 16

[56] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *RA-L*, 2022. 5, 16

[57] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *ICRA*, 2023. 16

[58] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. In *CoRL*, 2023. 16

[59] Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. Quest: Self-supervised skill abstractions for learning continuous control. In *NeurIPS*, 2024. 3

[60] Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie Pack Kaelbling. Learning finite-state controllers for partially observable environments. *arXiv preprint arXiv:1301.6721*, 2013. 5

[61] Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *CoRL*, 2022. 16

[62] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 2, 3, 4

[63] 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. In *ICRA*, 2024. 1, 2, 6, 10, 16

[64] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *NeurIPS*, 2016. 3

[65] Gabriel Quere, Annette Hagengruber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freek Stulp, and Joern Vogel. Shared Control Templates for Assistive Robotics. In *ICRA*, Paris, France, 2020. 16

[66] Alec Radford. Improving language understanding by generative pre-training. 2018. 3

[67] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 3, 4

[68] Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal Diffusion Transformer: Learning versatile behavior from multimodal goals. In *ICRA Workshops*, 2024. 6

[69] Saumya Saxena, Mohit Sharma, and Oliver Kroemer. Multi-resolution sensing for real-time control with Vision-Language Models. In *CoRL*, 2023. 16

[70] Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *ICLR*, 2024. 3

[71] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023. 16

[72] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A general navigation model to drive any robot. In *ICRA*, 2023. 6, 16

[73] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MUTEX: Learning unified policies from multimodal task specifications. In *CoRL*, 2023. 16

[74] Andrew Szot, Bogdan Mazoure, Harsh Agrawal, Devon Hjelm, Zsolt Kira, and Alexander Toshev. Grounding multimodal large language models in actions. In *NeurIPS*, 2024. 3

[75] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 4

[76] Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 1, 3

[77] 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 descriptions. In *ICLR*, 2023. 3

[78] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. BridgeData v2: A dataset for robot learning at scale. In *CoRL*, 2023. 1, 2, 6, 10, 16, 17

[79] Zihao Wang, Shaofei Cai, Zhancun Mu, Haowei Lin, Ceyao Zhang, Xuejie Liu, Qing Li, Anji Liu, Xiaojian Ma, and Yitao Liang. OmniJARVIS: Unified vision-language-action tokenization enables open-world instruction following agents. In *NeurIPS*, 2024. 3

[80] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. 5

[81] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *RSS*, 2023. 2, 10

[82] 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. In *ICLR*, 2024. 2, 16

[83] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *CoRL*, 2024. 2

[84] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020. 3

[85] Ge Yan, Kris Wu, and Xiaolong Wang. ucsd kitchens Dataset. August 2023. 16

[86] Jonathan Yang, Catherine Glossop, Arjun Bhorkar,

Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. In *RSS*, 2024. 2, 5

[87] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *ICLR*, 2025. 2, 3, 4, 6, 8, 18

[88] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. In *CoRL*, 2024. 2

[89] Kaiwen Zha, Lijun Yu, Alireza Fathi, David A Ross, Cordelia Schmid, Dina Katabi, and Xiuye Gu. Language-guided image tokenization for generation. *arXiv preprint arXiv:2412.05796*, 2024. 4

[90] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 4

[91] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based vlm plans the next step for vision-and-language navigation. In *RSS*, 2024. 7, 11

[92] Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. GRAPE: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024. 11

[93] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *RSS*, 2023. 5, 10

[94] 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. In *ICLR*, 2025. 5

[95] Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, et al. Train offline, test online: A real robot learning benchmark. In *ICRA*, 2023. 16

[96] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024. 3

[97] Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingxiao Huo, Wei Zhan, Masayoshi Tomizuka, and Mingyu Ding. Fanuc Manipulation: A dataset for learning-based manipulation with FANUC Mate 200iD Robot. 2023. 16

[98] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *RA-L*, 2022. 16

[99] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. VIOLA: Imitation learning for vision-based manipulation with object proposal priors. *arXiv preprint arXiv:2210.11339*, 2023. 16

APPENDIX

A. Implementation Details

1) Pretraining Details

For robotic manipulation data, we select a subset in Open X-Embodiment dataset [63] with single arm end-effector control. For navigation data, we use a sub-split of GNM [72] dataset containing both indoor and off-road scenes featuring a ego-view fisheye camera. While actions and proprioceptive states are available in the robot datasets, these are excluded during pretraining; only episode frames and text instructions are used. Additionally, we incorporate open-world human videos, specifically ego-centric videos that depict daily human activities from the Ego4D dataset [29]. Notably, with the exception of the SimplerEnv benchmark [47], which is designed to replicate the environmental setup of the Bridge-V2 dataset, *none* of the downstream evaluation environments have been seen by either our policy or the latent action model during pretraining. This necessitates strong generalization capabilities for both. The detailed composition of the datasets and mixture weights are listed in Tab. A-I.

Training Dataset Mixture	
Fractal [11]	13.9%
Kuka [36]	6.3%
Bridge [78]	6.8%
Taco Play [57]	3.5%
Jaco Play [21]	0.6%
Berkeley Cable Routing [51]	0.3%
Roboturk [55]	2.8%
Viola [99]	1.1%
Berkeley Autolab UR5 [15]	1.4%
Toto [95]	2.4%
Language Table [53]	5.2%
Stanford Hydra Dataset [7]	5.3%
Austin Buds Dataset [98]	0.3%
NYU Franka Play Dataset [19]	1.0%
Furniture Bench Dataset [31]	2.9%
UCSD Kitchen Dataset [85]	<0.1%
Austin Sailor Dataset [61]	2.6%
Austin Sirius Dataset [50]	2.0%
DLR EDAN Shared Control [65]	0.1%
IAMLab CMU Pickup Insert [69]	1.1%
UTAustin Mutex [73]	2.6%
Berkeley Fanuc Manipulation [97]	0.9%
CMU Stretch [58]	0.2%
BC-Z [34]	8.8%
FMB Dataset [52]	8.4%
DobbE [71]	1.7%
RECON [38]	8.9%
CoryHall [38]	2.3%
SACSoN [38]	3.5%
Ego4D [38]	3.0%

TABLE A-I: UniVLA training data mixture using datasets from the OXE [63], GNM [72] and Ego4D [29].

During training, we jointly optimize all components of our generalist policy, encompassing the visual encoders, the large language model (LLM) backbone, and the token prediction head. We utilize a batch size of 1,024 (with a per-device batch size of 32) and maintain a constant learning rate of $2e-5$. Empirical results indicate that 20,000 optimization steps are sufficient to achieve robust downstream performance, requiring approximately 30 hours of computation on a cluster equipped with 32 NVIDIA A100 GPUs. For pretraining on the "Human" and "Bridge" datasets (as presented in Table Tab. I), we employ a global batch size of 258 distributed across 8 GPUs, totaling approximately 200 A100 GPU-hours.

TABLE A-II: **Language-conditioned visuomotor control on CALVIN ABC→D.** We report success rates along with the average length of completed tasks (out of the whole 5 tasks) per evaluation sequence. UniVLA achieves competitive results while being the only method that relies on solely third-view RGB inputs. *Reproduced with action chunks prediction.

Method	Task completed in a row (%) ↑					Avg.
	1	2	3	4	5	Len.
RT-1 [11]	53.3	22.2	9.4	3.8	1.3	0.90
RoboFlamingo [46]	82.4	61.9	46.6	33.1	23.5	2.48
SuSIE [9]	87.0	69.0	49.0	38.0	26.0	2.69
GR-1 [82]	85.4	71.2	59.6	49.7	40.1	3.06
OpenVLA* [39]	91.3	77.8	62.0	52.1	43.5	3.27
CLOVER [14]	96.0	83.5	70.8	57.5	45.4	3.53
RoboDual [13]	94.4	82.7	72.1	62.4	54.4	3.66
UniVLA (Ours)	**95.5**	**85.8**	**75.4**	**66.9**	**56.5**	**3.80**

B. Additional Results

1) CALVIN

Experiment setup. CALVIN [56] encompasses 34 distinct tasks, characterized by unconstrained task instructions that span a spectrum of skills, ranging from basic pick-and-place operations to articulated object manipulation. The benchmark includes four distinct environments, each featuring a Franka Panda robotic arm for tabletop manipulation tasks. In our study, we adopt the challenging evaluation setting, wherein policies are trained using demonstrations from environments A, B, and C, followed by zero-shot evaluations in environment D. The evaluation protocol comprises a test set of 1,000 unique instruction chains, each consisting of five consecutive tasks, designed to rigorously assess the generalization capabilities of the policies.

For OpenVLA, we finetune the officially provided checkpoint with LoRA [33] for 200k steps and use an action chunk with size 8 to maximize performance. UniVLA is optimized for 100k steps with a batch size of 128. We use a learning rate of $1.5e-4$ for the first 80k steps, and $1.5e-5$ for the rest. Similar to LIBERO experiments, we only take as inputs third-view RGB images and language instructions.

Results. The results in Tab. A-II demonstrate UniVLA's state-of-the-art performance in language-conditioned visuomotor control. UniVLA achieves 56.5% success rate for completing all five tasks in sequence, surpassing the prior best method, CLOVER (45.4%) by 11.1%, and OpenVLA by 13%. The average number of consecutively completed tasks increases from OpenVLA's 3.27 to 3.80. Notably, UniVLA's performance gap

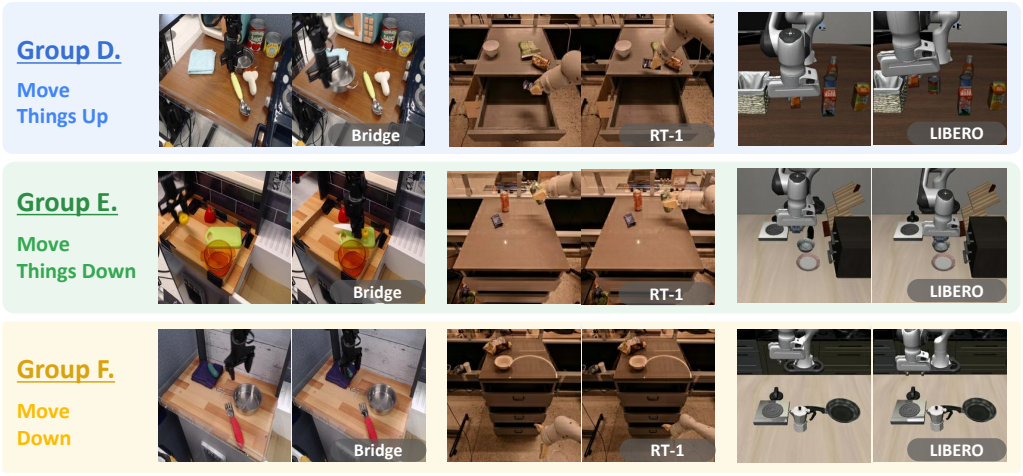

Fig. A-1: **Latent action analysis.** We show more image pairs labeled the same latent action from different source of data and embodiments. Each group of latent action presents semantic-consistent action

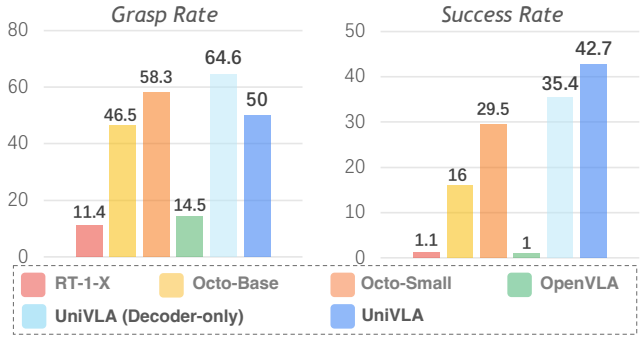

Fig. A-2: **Grasp and task success rates on SimplerEnv.** As BridgeData is incorporated in our pretraining dataset, we investigate only training the decoder for adaptation (Decoder-only). UniVLA outperforms all baselines in success rate.

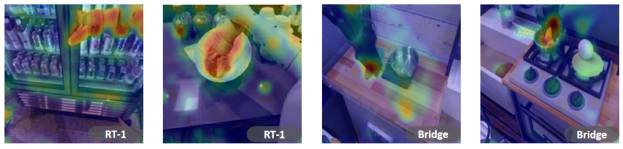

Fig. A-3: **Task-centric Latent action analysis.** We show the attention heatmap between task-centric latent actions and image patches, demonstrating concentrated focus on the robotic end-effector and target objects.

widens progressively with task length, reflecting its ability to tackle complex, long-horizon manipulation tasks.

2) SimplerEnv

Experiment setup. SimplerEnv [47] are developed to genuinely reflect the performance of real-world policies by mirroring physical dynamics and visual appearances. We focus on four tasks concerning the "WidowX + Bridge" setup: 1) "Put spoon on table cloth", 2) "Put carrot on plate", 3) "Stack green cude on the yellow cude", and 4) "Put eggplant in basket". The pose and position of objects to be grasped will be randomly initialized given different seeds. Given that Bridge-V2 [78] data is included in our pretraining dataset, we investigate training the action decoder head exclusively while keeping the remaining components of our model fixed, denoted as *decoder-only* in Fig. A-2. Additionally, we perform further fine-tuning using LoRA [33] on the complete Bridge-V2 dataset and evaluate the resulting policy. Our evaluation follows the pipeline proposed by Li et al. [45], wherein each task is assessed over 24 independent trials to ensure robust performance metrics.

Results. The bar plot in Fig. A-2 underscores UniVLA's superior performance in both grasp and task success rates on the SimplerEnv-Bridge benchmark, even under the constrained decoder-only adaptation setting. Specifically, decoder-only adaptation achieves a 35.4% success rate, demonstrating its ability to retain pretrained knowledge while minimizing adaptation costs. However, full fine-tuning results in a reduced grasp rate compared to decoder-only training, likely due to overfitting to seen scene layouts in training samples. Overall, UniVLA achieves a 42.7% task success rate, outperforming OpenVLA and Octo-Base by 41.7% and 26.7%, respectively.

3) More Visualization

Additional examples for latent action analysis. As discussed in Sec. IV-A2, We explore the cross-domain transferability of latent actions by presenting image pairs from diverse data sources that share the same latent action. Additional examples with distinct actions are provided in Fig. A-1.

Task-centric Latent action analysis. We visualize the attention maps between learned task-centric latent actions and image patches in Fig. A-3. The heatmaps reveal concentrated attention on task-critical regions: the robotic arm's end-effector (e.g., gripper) and interacted objects (e.g., egg), while ignoring irrelevant background. This demonstrates that the latent action inherently encodes task-centric spatial priors, focusing only on entities necessary for downstream learning.

TABLE A-III: **Scores of tasks.** Each sub-goal corresponds to one point.

Task Name	Total Score	Sub-goals
Store the screwdriver	3	Pick up the screwdriver. Place the screwdriver into the cabinet. Close the cabinet
Clean the cutting board	3	Pick up the broom. Sweep the items into dustpan. Sweep **all** items into dustpan.
Fold towel twice	3	Grasp the correct edge of the towel. Fold towel for the first time. Fold towel for the second time.
Stack tower of hanoi	3	Choose the right tower. Stack the medium tower on top of the large one. Stack the small tower on top of the medium one.

TABLE A-IV: **Experiment result.** We **bold** the best result and underline the second.

Method	Store screwdriver		Clean cutting board		Fold towel twice		Stack tower of hanoi		Average ↑	
	Succ.	Score	Succ.	Score	Succ.	Score	Succ.	Score	Succ.	Score
Diffusion Policy [17]	40.0	1.20	33.3	0.67	**53.3**	2.33	6.7	1.6	33.3	1.45
OpenVLA [39]	40.0	1.47	53.3	1.27	33.3	1.87	26.7	1.93	38.3	1.63
LAPA (OXE) [87]	60	2.0	40	1.47	33.3	2.2	46.7	2.13	45	1.95
UniVLA (Bridge)	66.7	2.13	73.3	1.87	33.3	2.07	46.7	2.13	55.0	2.05
UniVLA (OXE)	80.0	2.73	73.3	1.93	33.3	2.13	60.0	2.60	63.3	2.35
UniVLA (Full)	**93.3**	**2.87**	**100.0**	**2.33**	46.7	**2.47**	**86.7**	**2.87**	**81.7**	**2.63**

C. Real-world Robots

1) Task setup and evaluation

In task "Store the screwdriver", we randomly placed the screwdriver in three different positions for position generalization in training data, and tested it at four positions during evaluation. In task "Store the screwdriver", we randomly placed items on the cutting board, some of which in some cases could be swept into the dustpan in a single motion, while others required two sweeps during data collection. For task "Fold towel twice", we use a 20cm $\times$ 20cm towel and lay it flat on the table. During both training and testing, we randomly rotated the towel by different angles for evaluating generalization. In "Stack tower of hanoi", we randomly shuffle three cups and cover six different arrangements during both training and testing.

We evaluate policies using a combined metric of success rate and task-specific scoring. Tab. A-III shows the detailed scoring criteria. Full experimental results are presented in Tab. A-IV. Notably, certain policies achieve identical success rates but exhibit divergent scores, reflecting differences in their performance quality.

Single-task-trained methods such as Diffusion Policy perform well in tasks like Fold towel twice, which demand smooth, continuous, and highly structured action sequences. Although Diffusion Policy achieves a higher success rate in this task, its score remains lower than UniVLA due to limited generalization across varying rotation angles. In contrast, UniVLA 's generalizability enables robust performance across diverse positional configurations, resulting in a higher score even in cases of partial task completion.

TABLE A-V: **Architecture details of action decoder in real-world experiment.** Additional proprioceptive state projection module is only adopted in real-world experiments.

Architecture of Action Decoder		
Latent Action Attention Pooling	Heads	8
	Head Dim.	64
	Hidden Size	512
	MLP Ratio	4
Visual Embedding Attention Pooling	Heads	8
	Head Dim.	64
	Hidden Size	512
	MLP Ratio	4
Action Projection	Layers	1
	Hidden Size	512
Proprio. Projection	Layers	2
	Hidden Size	512
Parameters		12.6M

2) Architecture of the action decoder

In the design of the action decoder architecture for both LAPA and our method, we use 2 multi-head attention blocks to process the latent action and visual embeddings (also refer to Sec. III-C), with a MLP layer to process proprioceptive states. The resulting embeddings are then concatenated and mapped to the desired action dimensions through a projection layer. Detailed parameters are shown in Tab. A-V.