HIERARCHICAL LATENT ACTION MODEL

Hanjung Kim[1,2], **Lerrel Pinto**[2], **Seon Joo Kim**[1]
[1] Yonsei University, [2] New York University

ABSTRACT

Latent Action Models (LAMs) enable learning from actionless data for applications ranging from robotic control to interactive world models. However, existing LAMs typically focus on short-horizon frame transitions and capture low-level motion while overlooking longer-term temporal structure. In contrast, actionless videos often contain temporally extended and high-level skills. We present HiLAM, a hierarchical latent action model that discovers latent skills by modeling long-term temporal information. To capture these dependencies across long horizons, we utilize a pretrained LAM as a low-level extractor. This architecture aggregates latent action sequences, which contain the underlying dynamic patterns of the video, into high-level latent skills. Our experiments demonstrate that HiLAM improves over the baseline and exhibits robust dynamic skill discovery.

1 INTRODUCTION

Recent progress in robot learning has increasingly relied on incorporating large-scale data for training. However, obtaining action-labeled data is prohibitively expensive and makes it difficult to ensure dataset diversity. To remedy this, Latent Action Models (Schmidt & Jiang, 2024; Bruce et al., 2024; Ye et al., 2025; Kim et al., 2025) have emerged as a prominent approach by extracting latent actions directly from observation-only data. Generally, by utilizing inverse and forward dynamics models, LAMs infer the latent action between two frames. These latent actions are then used for pretraining Vision-Language-Action models (VLAs) with actionless data (Ye et al., 2025; Bu et al., 2025), transferring actions across different embodiments (Kim et al., 2025), or enabling interaction within world models (Bruce et al., 2024; Gao et al., 2025).

Latent actions offer a promising way to leverage dynamic information from diverse video sources. Despite their flexibility, existing latent action models are largely limited to short-term motion. As a result, they can capture low-level dynamics from observation-only data but often miss higher-level structure, such as temporally extended skills. This exposes a key gap where actionless videos contain not only primitive motions but also high-level skills that remain underutilized.

This raises a natural question: how can we extract such skills from unlabeled video? Prior work typically either assumes a fixed set of skill vectors (Zhu et al., 2022; Liang et al., 2024) or encodes fixed-length sequences of low-level actions into skill representations (Pertsch et al., 2021a). In contrast, real-world skills vary in duration, and large-scale data introduces an increasingly diverse set of behaviors. Even for the same task, demonstrations can vary substantially in execution speed and, consequently, in skill duration. When skills are forced into a fixed-length window, two trajectories that express the same underlying behavior may be mapped to very different skill representations. Another line of work uses language to define skills (Shi et al., 2025). However, it typically segments behavior from task descriptions, such as by splitting an instruction into sub-instructions, rather than from motion cues. Therefore, language is a complementary signal for skill discovery and not a replacement for modeling dynamics.

To this end, we propose HiLAM, a hierarchical latent action model that encodes latent skills from sequences of latent actions, regardless of their length or the need for pre-defined skill sets. 1 demonstrates the overall architecture of HiLAM. To enable a dynamic hierarchical design, we adopt the H-Net (Hwang et al., 2025) architecture, which introduces a novel dynamic chunking mechanism that automatically segments boundaries. Following the H-Net framework, we formulate HiLAM using a next-token prediction approach during pretraining, utilizing latent actions extracted from an inverse dynamics model (IDM). Additionally, predicted latent actions are used to reconstruct future

frames, consistent with prior works (Ye et al., 2025; Kim et al., 2025), to maintain their dynamic motion properties. Due to H-Net's dynamic chunking mechanism, latent actions are naturally grouped into similar representations of varying lengths without the need for action labels. These representations are then processed through an encoder to serve as latent skills. Once these skills are obtained, we train a skill policy to predict the latent skill based on the current observation. Simultaneously, we train a skill-conditioned policy to predict low-level actions based on the observation and the predicted latent skills.

Our experiments show that HiLAM is able to detect and encode skill representations while remaining free from constraints on length or pre-defined skill sets. Furthermore, the predicted next latent actions maintain interpretability, which is demonstrated by predicting future frames corresponding to the given latent actions. In terms of training computation, since HiLAM reuses pretrained LAMs for extracting latent actions, it is capable of encoding long-horizon trajectories efficiently.

2 Related Work

2.1 Latent action learning

Latent action learning is a prominent approach for inferring actions from observation-only data. By analyzing frame transitions, Latent Action Models (LAMs) extract the latent action between frames using an Inverse Dynamics Model (IDM). LAPO (Schmidt & Jiang, 2024) focuses on inferring discrete latent actions from gaming environments, while Genie (Bruce et al., 2024) proposes an interactive world model for games through a discrete latent action space. Since prior works were largely limited to simulated environments, LAPA (Ye et al., 2025) introduces latent action learning to robotics, leveraging diverse actionless data by utilizing latent actions as pseudo-labels for training VLAs.

Standard LAMs often utilize a Forward Dynamics Model (FDM) to predict future images, where the reconstruction objective encodes dynamic information into the latent action. However, this process can accidentally incorporate task-irrelevant information into latent representation. To address this, UniVLA (Bu et al., 2025) proposes a task-centric learning approach, while UniSkill (Kim et al., 2025) adopts an image-editing pipeline and LAOM (Nikulin et al., 2025) utilizes supervised learning to reduce such dependencies. Furthermore, to represent more diverse action spaces, UniSkill (Kim et al., 2025) and CLAM (Liang et al., 2025) employ continuous latent actions rather than discrete ones. Latent actions also serve as a substitute for explicit action labels in world models, which is necessary for interaction. Recent works such as AdaWorld (Gao et al., 2025) and Latent Action World Model (Garrido et al., 2026) utilize latent action learning specifically for world model training.

However, these LAMs are generally limited to short-period motion patterns and lack the capacity to represent high-level skills. To address this, HiLAM introduces a method for extracting latent skills from actionless data.

2.2 Hierarchical Robot Learning

Hierarchical robot learning uses skill representations as an intermediate abstraction for action prediction, enabling policies to better handle long-horizon and complex tasks. Unlike low-level actions, skills are difficult to annotate manually within demonstrations. Consequently, prior work often adopts unsupervised training paradigms to learn these abstractions from action sequences.

SPiRL (Pertsch et al., 2021a) and SkiLD (Pertsch et al., 2021b) employ autoencoder architectures to learn skill representations from fixed-length action sequences, using these learned priors to accelerate reinforcement learning in downstream tasks. SAILOR (Nasiriany et al., 2022) also uses an autoencoder framework for skill extraction, further incorporating a temporal-distance objective to ensure the learned representations are temporally aware. Rather than using latent embeddings, BUDS (Zhu et al., 2022) discovers skills by clustering unsegmented demonstrations into temporal segments based on a set of predefined skill primitives. SkillDiffuser (Liang et al., 2024) adopts VQ-VAE (van den Oord et al., 2017) to encode high-level instructions into a discrete set of learnable skills and uses them for future-frame generation. Meanwhile, Hi Robot (Shi et al., 2025) uses a high-level VLM to map user prompts into low-level language commands.

Prior work typically assumes a fixed horizon for encoding skill representations, a fixed number of skill primitives, or a different modality. In contrast, HiLAM introduces a dynamic chunking mechanism that abstracts sequences of low-level motions in a data-driven and length-adaptive way.

3 METHOD

In this paper, we propose HiLAM, a hierarchical latent action model. The framework consists of two phases. In the first phase, HiLAM is trained to predict the next latent action utilizing a dynamic chunking mechanism. Due to its architectural properties, HiLAM automatically detects skill boundaries within untrimmed input latent action sequences. In the second phase, we utilize these latent actions and the encoded latent skills to train high-level and low-level policies. Since each latent skill contains shared information within a chunked segment, the high-level policy is trained to predict latent skills, while the low-level policy is trained to predict latent actions. Finally, we fine-tune the low-level policy with ground-truth actions to map the latent action space to the true action space.

3.1 PRELIMINARIES

3.1.1 PROBLEM FORMULATION

We formalize the objective as encoding high-level latent skills, z^h, from observation-only videos $\mathcal{V}$ to facilitate hierarchical policy learning. Our approach decomposes long-horizon trajectories into a hierarchy of latent representations, mapping visual observations to executable actions. Given an observation-only video $\mathcal{V}$ of length T, we first extract a sequence of low-level latent actions $\{z_1^l, ..., z_{T-k}^l\}$. Following prior work (Ye et al., 2025; Kim et al., 2025), we use an inverse dynamics model to infer the motion between two frames, I_t and I_{t+k}, separated by a fixed temporal interval k (Figure 1c). To capture temporal abstractions, this sequence of low-level latent actions is further compressed into a sequence of high-level latent skills $\{z_1^h, ..., z_S^h\}$, where $S < T - k$. We allow both T and S to be variable, accommodating trajectories and skills of varying temporal durations. For downstream control, we employ a hierarchical policy framework. At each decision step, a high-level policy observes the current state o_t and a task instruction l to predict a target latent skill, $z_t^h \sim \pi^h(z_t^h|o_t, l)$. Subsequently, conditioned on this high-level skill and the current observation, a low-level policy generates the primitive action a_t for robot execution via $a_t \sim \pi^l(a_t|o_t, z_t^h)$.

3.1.2 DYNAMIC CHUNKING MECHANISM

To abstract long sequences of latent actions into temporally extended skills, HiLAM adopts the H-Net (Hwang et al., 2025) architecture, which learns a data-driven segmentation of the input via dynamic chunking. At a given stage s, let the input sequence be $\mathbf{z}^s = (z_1^s, \dots, z_{L^s}^s)$. An encoder $\mathcal{E}^s$ maps each token to a feature vector $h_t^s \in \mathbb{R}^d$. H-Net then predicts boundary indicators $b_t^s \in \{0, 1\}$ that decide whether token t starts a new chunk. We interpret $b_t^s = 1$ as a *segment-start* boundary (i.e., token t is the first token of a new chunk). Following Hwang et al. (2025), we compute normalized query/key features $\hat{\mathbf{q}}_t^s$ and $\hat{\mathbf{k}}_t^s$ from h_t^s and define

$$p_t^s = \begin{cases} 1, & t = 1, \\ \frac{1}{2}\big(1 - (\hat{\mathbf{q}}_{t-1}^s)^\top \hat{\mathbf{k}}_t^s\big), & t > 1, \end{cases} \qquad b_t^s = \mathbb{1}_{\{p_t^s \geq 0.5\}}. \tag{1}$$

Intuitively, p_t^s is large when consecutive tokens are dissimilar, encouraging a boundary at t. Given the boundary pattern, we perform chunking (downsampling) by selecting only the boundary features. Let $\mathcal{I}^s = \{t \mid b_t^s = 1\}$ denote the selected boundary indices (ordered increasingly), and let $L^{s+1} = |\mathcal{I}^s|$. The chunked sequence is obtained as

$$z_i^{s+1} = h_{t_i}^s, \qquad \text{where } t_i \in \mathcal{I}^s. \tag{2}$$

That is, the stage-$(s+1)$ tokens are selected encoder features at boundary indices and serve as chunk-level summaries. A main network $\mathcal{M}^s$ then processes the shorter sequence $\mathbf{z}^{s+1} = (z_1^{s+1}, \dots, z_{L^{s+1}}^{s+1})$, and a decoder $\mathcal{D}^s$ expands the processed sequence back to length L^s conditioned on the same boundary pattern. Overall, the encode–chunk–main–dechunk stage is summarized by

$$\begin{aligned} \mathbf{h}^s &= \mathcal{E}^s(\mathbf{z}^s), & \mathbf{z}^{s+1} &= \mathrm{Chunk}(\mathbf{h}^s; \mathbf{b}^s), \\ \hat{\mathbf{z}}^{s+1} &= \mathcal{M}^s(\mathbf{z}^{s+1}), & \hat{\mathbf{z}}^s &= \mathrm{DeChunk}(\mathcal{D}^s(\hat{\mathbf{z}}^{s+1}); \mathbf{b}^s). \end{aligned} \tag{3}$$

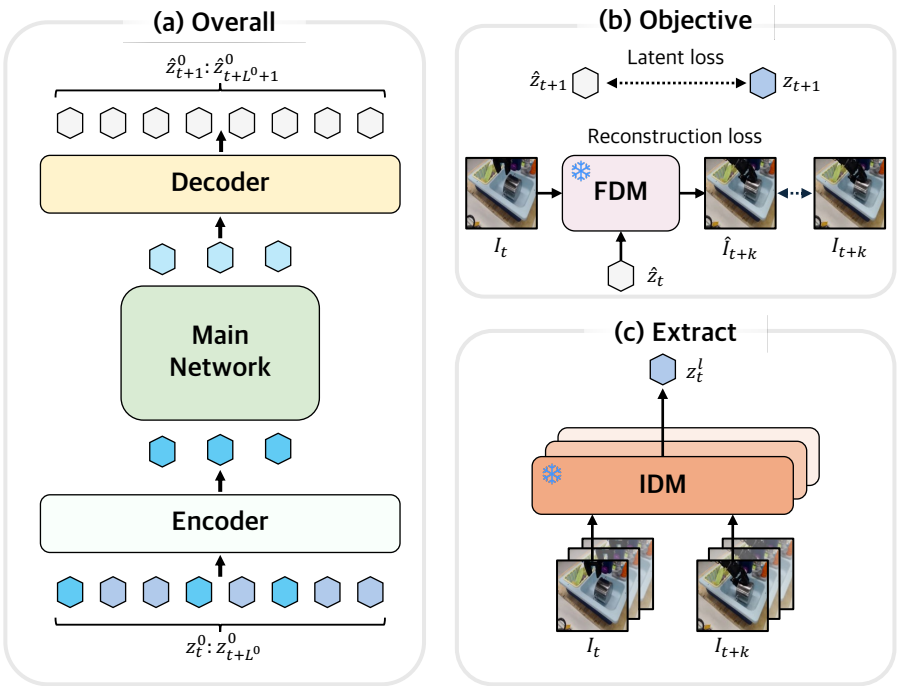


Figure 1: Overview of HiLAM. (a) Overall latent skill learning pipeline. (b) Training objectives used for latent skill learning. (c) Extracting latent actions using a pretrained inverse dynamics model (IDM).

Stacking multiple stages yields a hierarchical representation in which higher levels operate on progressively shorter, chunk-level sequences.

3.2 Latent Skill Learning

We learn a hierarchy of latent skill representations from observation-only videos using a hierarchical sequence model. In sequence modeling, inputs typically consist of language tokens, DNA bases, or action trajectories. However, because ground-truth actions are unavailable in observation-only data, we use a pretrained Inverse Dynamics Model (IDM) (Ye et al., 2025; Kim et al., 2025) to extract latent actions. We then apply the dynamic chunking mechanism (Hwang et al., 2025) to segment the latent action sequence into meaningful temporal chunks, encoding each chunk as a latent skill (Figure 1).

Let $\mathbf{z}^l$ denote the resulting latent action sequence, and we use it as the initial input to HiLAM, i.e., $\mathbf{z}^0 = \mathbf{z}^l$. After each encoder $\mathcal{E}^s$, the model predicts boundary indicators $\mathbf{b}^s$ and selects representative tokens to form the chunked sequence $\mathbf{z}^{s+1}$. As described in Equation (1), boundaries are determined from feature dissimilarities, encouraging segmentation at points of large temporal change. The selected tokens summarize each segment, effectively compressing a variable-length sequence of latent actions into a shorter sequence of segment-level representations. We treat these higher-level tokens $\mathbf{z}^s$ (for $s > 0$) as latent skill representations, denoted $\mathbf{z}^h$.

Finally, after hierarchical processing through the encoder–main–decoder stack, HiLAM predicts the next latent actions at the lowest level. Given an input sequence $\mathbf{z}^0 = (z_1^0, \ldots, z_{L^0}^0)$, the model outputs $\hat{\mathbf{z}}^0 = (\hat{z}_2^0, \ldots, \hat{z}_{L^0+1}^0)$ via next-token prediction.

Training objective. We optimize a weighted combination of (i) next-latent prediction, (ii) a visual supervision term that can be instantiated in different ways, and (iii) the H-Net chunking regularizer:

$$\mathcal{L} = \mathcal{L}_{\text{latent}} + \lambda_{\text{rec}}\,\mathcal{L}_{\text{rec}} + \lambda_{\text{ratio}}\,\mathcal{L}_{\text{ratio}}. \tag{4}$$

In this formulation, $\mathcal{L}_{\text{latent}}$ is an element-wise ℓ_1 loss (Bardes et al., 2024; Assran et al., 2025) between predicted and target latent actions, modeling the next-token prediction task. The term

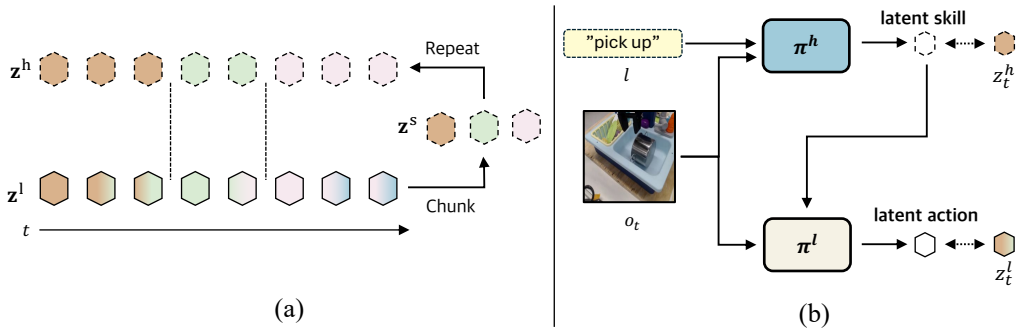


Figure 2: Latent skill extraction and policy learning. (a) Latent actions $\mathbf{z}^l$ are hierarchically encoded into stage-wise representations $\mathbf{z}^s$ and then expanded back to a per-timestep latent skill sequence $\mathbf{z}^h$. (b) Overall pipeline of the hierarchical skill policy.

$\mathcal{L}_{\text{rec}}$ provides additional supervision to ensure that the predicted latents maintain their action-like properties. To achieve this, we employ a pretrained Forward Dynamics Model (FDM) to predict future frames conditioned on the predicted latent actions. For instance, if $\hat{z}^0_{t+1}$ is predicted from z^0_t, the FDM is expected to predict the future frame $\hat{I}_{t+k+1}$ using the current frame I_{t+1} and the predicted latent action $\hat{z}^0_{t+1}$. To preserve the dynamic motion properties of the latent action, we define $\mathcal{L}_{\text{rec}}$ as the reconstruction error between the predicted frame $\hat{I}_{t+k+1}$ and the ground-truth frame I_{t+k+1}. Finally, $\mathcal{L}_{\text{ratio}}$ denotes the H-Net ratio regularizer (Hwang et al., 2025) that prevents degenerate boundary patterns and controls the average chunk length.

Latent Skill Extraction. After training HiLAM, we extract stage-wise latent skills and align them to the original video length T for downstream hierarchical policy learning. Given an observation-only video $\mathcal{V} = \{I_1, \ldots, I_T\}$, HiLAM produces at each stage s a downsampled sequence of latent representations $\mathbf{z}^s = \{z^s_1, \ldots, z^s_{L^s}\}$ with $L^s \leq T$ (Equation (3)). Because $\mathbf{z}^s$ is defined at the stage resolution, we expand it back to the original temporal resolution using the unfolded boundary.

Let $\bar{\mathbf{b}}^s \in \{0,1\}^T$ denote the *unfolded boundary* of stage s, i.e., the boundary pattern expressed at the original temporal resolution. We assign each timestep t to its segment ID via a cumulative sum and expand the stage-s sequence by indexing:

$$k^s_t = \sum_{\tau=1}^{t} \bar{b}^s_\tau, \qquad \bar{z}^s_t = z^s_{k^s_t}, \qquad t = 1, \ldots, T. \tag{5}$$

Here, k^s_t is constant within a segment and increases by 1 whenever $\bar{b}^s_t = 1$, so each timestep inherits the latent representation of its corresponding segment. As shown in Figure 2, the resulting sequence $\bar{\mathbf{z}}^s = \{\bar{z}^s_1, \ldots, \bar{z}^s_T\}$ serves as the per-timestep latent skill sequence $\mathbf{z}^h$ used for downstream policy learning.

3.3 Hierarchical Policy Learning

To leverage the learned latent skills for control, we train a hierarchical policy consisting of a high-level policy π^h and a low-level policy π^l (Figure 2b). We consider two training phases: pretraining on large-scale, actionless videos and fine-tuning on a target domain with action labels.

Pretraining. During pretraining, we supervise both policies using the latent skill/action sequences extracted by HiLAM. The high-level policy predicts a latent skill from the current observation and task description, $\hat{z}^h_t \sim \pi^h(z^h_t \mid o_t, l)$. Conditioned on the observation and the predicted skill, the low-level policy predicts the corresponding latent action, $\hat{z}^l_t \sim \pi^l(z^l_t \mid o_t, \hat{z}^h_t)$. We train π^h and π^l to match the extracted targets z^h_t and z^l_t, respectively. Because these targets are obtained from observation-only videos, pretraining can use diverse video sources (e.g., robot or human videos).

Fine-tuning. After pretraining, we freeze the high-level policy π^h and fine-tune the low-level policy on target-domain data with ground-truth actions. Given the predicted skill $\hat{z}_t^h$, the low-level policy outputs a real action for execution, $\hat{a}_t \sim \pi^l(a_t \mid o_t, \hat{z}_t^h)$.

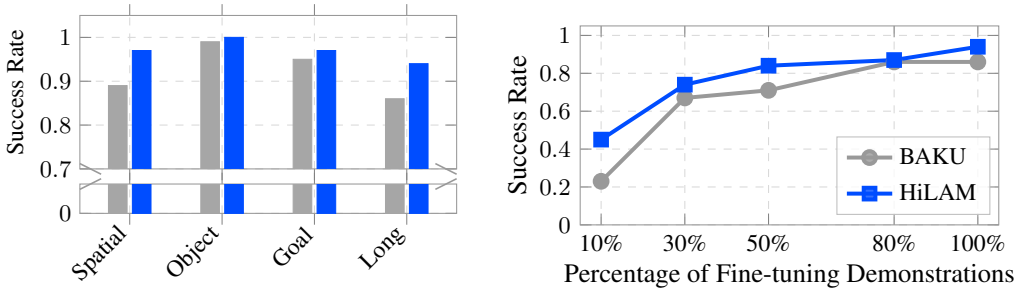
4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Datasets We train HiLAM on observation-only video datasets spanning both human and robot behaviors. For human videos, we use Something-Something V2 (Goyal et al., 2017), which contains short clips of humans performing diverse object-centric actions. For real-world robot videos, we use Droid (Khazatsky et al., 2024) and BridgeV2 (Ebert et al., 2022), which are large-scale datasets collected with Franka and WidowX-250 robot arms, respectively.

Implementation details Following H-Net (Hwang et al., 2025), HiLAM uses a two-stage H-Net. To extract latent actions from actionless videos, we use UniSkill (Kim et al., 2025)'s inverse dynamics model (IDM) as the latent-action tokenizer, and its forward dynamics model (FDM) for frame prediction conditioned on the predicted latent actions. For latent skills $\mathbf{z}^h$, we use the stage-$s = 2$ representations, i.e., $\mathbf{z}^h \equiv \mathbf{z}^2$. For hierarchical policy learning, both the high-level and low-level policies are based on the BAKU (Haldar et al., 2024) architecture, and we use a T5 encoder (Raffel et al., 2020) as the language encoder. For pretraining, we use either human videos (Something-Something V2) or robot videos (BridgeV2). In both cases, we treat the data as observation-only: we discard any available action annotations, extract latent actions/skills solely from each video, and use them as pseudo-labels to train both the high-level and low-level policies. For fine-tuning, we freeze the high-level policy and train only the low-level policy using expert demonstrations. Unless otherwise stated, both pretraining and fine-tuning are run for 100k gradient steps.

Benchmark We evaluate downstream control on the LIBERO benchmark (Liu et al., 2023). We report results on four suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. LIBERO-Spatial emphasizes spatial reasoning, while LIBERO-Object tests generalization by varying the manipulated objects. LIBERO-Goal uses consistent objects and backgrounds, requiring the policy to follow the language instruction to succeed. LIBERO-Long is the most challenging suite, consisting of long-horizon tasks with multiple sub-goals. Each suite contains 10 tasks, and each task provides 50 demonstration trajectories. We fine-tune on the provided expert demonstrations and report success rates using the official evaluation rollouts.



(a) (b)

Figure 3: LIBERO benchmark results. (a) Performance of BAKU (gray) and HiLAM (blue) on the LIBERO benchmark. (b) LIBERO-Long success rate as a function of the fraction of expert demonstrations used for fine-tuning.

Table 1: Ablations on LIBERO-Long, effects of pretraining data and the stage used for latent skill/action conditioning.

Method	Pretraining dataset	Latent skill	Latent action	Success rate
BAKU	Robot	–	$\bar{z}^0$	0.87
	Robot	–	$\bar{z}^2$	0.81
	Human	–	$\bar{z}^0$	0.91
	Human	–	$\bar{z}^2$	0.87
HiLAM	–	–	–	0.67
	Robot	$\bar{z}^1$	$\bar{z}^0$	0.90
	Robot	$\bar{z}^2$	$\bar{z}^0$	0.90
	Robot	$\bar{z}^2$	$\bar{z}^1$	0.87
	Human	$\bar{z}^1$	$\bar{z}^0$	0.89
	Human	$\bar{z}^2$	$\bar{z}^0$	**0.94**
	Human	$\bar{z}^2$	$\bar{z}^1$	0.89

4.2 Results

4.2.1 LIBERO Benchmark Results

Figure 3 summarizes performance on the LIBERO benchmark. In Figure 3a, we compare HiLAM against BAKU (Haldar et al., 2024), a recent state-of-the-art baseline. Across all four suites, HiLAM consistently outperforms BAKU, demonstrating the effectiveness of learning temporally extended latent skills from actionless videos.

LIBERO-Long and data efficiency. Figure 3b focuses on LIBERO-Long, which consists of long-horizon, multi-stage tasks and therefore provides a more stringent test of whether the learned latent skills capture meaningful temporal structure. We pretrain a hierarchical policy using pseudo-labels (latent skills and latent actions) extracted by HiLAM from diverse actionless videos, and then fine-tune with varying fractions of expert demonstrations. With only 10% of the demonstrations, BAKU achieves a 23% success rate, whereas HiLAM achieves 45%, nearly doubling performance. With 50% of the demonstrations, HiLAM reaches 84%, comparable to BAKU trained with 100% of the data. With the full 100% of demonstrations, HiLAM achieves 94%, outperforming BAKU by a large margin.

4.2.2 Ablation Studies

To validate the effectiveness of HiLAM, we conduct ablation studies on LIBERO-Long by varying key components, as shown in Table 1. Because HiLAM uses a hierarchical policy to jointly leverage latent skills and latent actions, we study which combinations are most effective for pretraining.

First, we vary the pretraining dataset (human vs. robot videos). Both improve performance, but human videos perform best; we therefore use human pretraining by default.

Next, we vary the stage index s used for conditioning via the unfolded representations $\bar{z}^s$ for latent skills and latent actions. As shown in Table 1, using $s = 2$ for latent skills and $s = 0$ for latent actions yields the best performance across both human and robot pretraining. The stage-2 representation is produced by the deepest encoder and thus captures longer-range temporal context with more semantically clustered segments; we adopt this setting as default.

We also evaluate whether a non-hierarchical (flat) policy can benefit from latent conditioning by pretraining BAKU with latent actions from either $s = 0$ or $s = 2$. While this improves performance, it still lags behind the hierarchical policy, highlighting both the benefit of latent actions and the need for hierarchical policy learning. Finally, training the hierarchical policy only on the target tasks (without large-scale pretraining) significantly degrades performance, indicating that the gains are not due to the policy architecture alone.

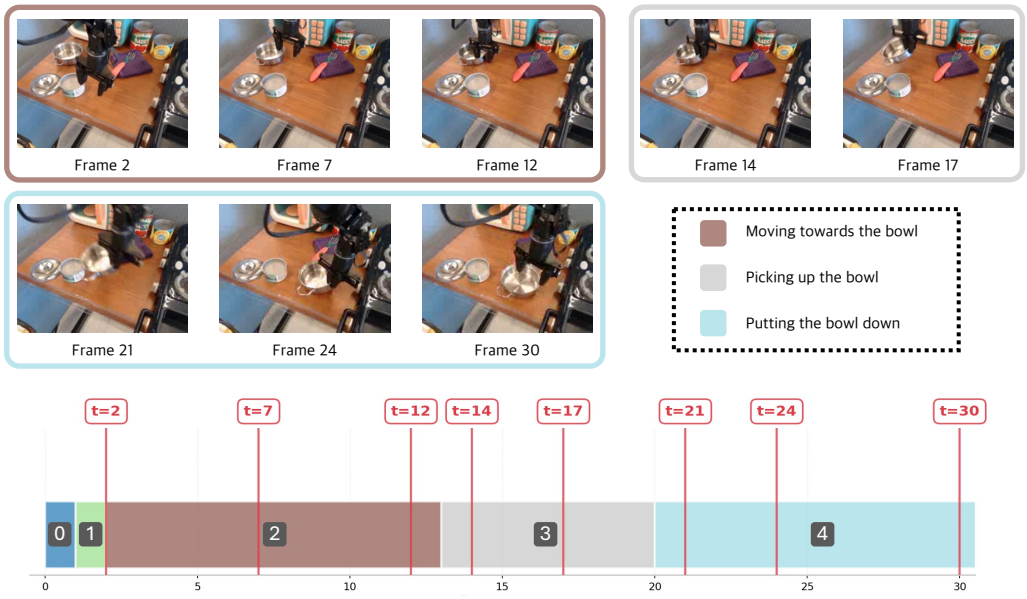


Figure 4: Qualitative results for skill boundary prediction. Using the predicted boundary indicators b_t^s, we assign each frame to a skill segment k_t^s and display the segment ID for each segment.

4.2.3 Dynamic Skill Chunking

During training, HiLAM naturally identifies segment boundaries that correspond to individual, semantically meaningful skills. As described in Equation (5), we utilize the predicted boundary indicators to partition the trajectory into segments. Each resulting segment consists of a variable-length sequence of latent actions and is interpreted as a distinct latent skill.

To assess this discovery capability, we visualize the inferred chunks in Figure 4. Each segment is assigned a distinct color and ID corresponding to the boundary predictions from HiLAM. In Segment 2, the gripper moves across the workspace toward the bowl. In Segment 3, HiLAM predicts a new boundary as the gripper picks up the bowl. Finally, in Segment 4, the gripper moves to the target location and places the bowl down.

Despite being trained in a fully unsupervised manner without any labels or ground-truth actions, HiLAM consistently groups latent action sequences into coherent skills. This qualitative result indicates that the proposed dynamic chunking mechanism captures meaningful temporal structure.

4.2.4 Latent Action Prediction

As described in Section 3.2, HiLAM predicts the next latent action from z_t^l to z_{t+1}^l. Therefore, it should preserve the action-like property of the latent representation, i.e., the motion pattern between two frames I_t and I_{t+k}. To verify that the predicted latent actions retain this motion information, we evaluate them via future-frame prediction. Using the pretrained FDM, we generate the future frame I_{t+k} from the current frame I_t and the predicted latent action $\hat{z}_t^l$. Figure 5 shows qualitative results. Although $\hat{z}_t^l$

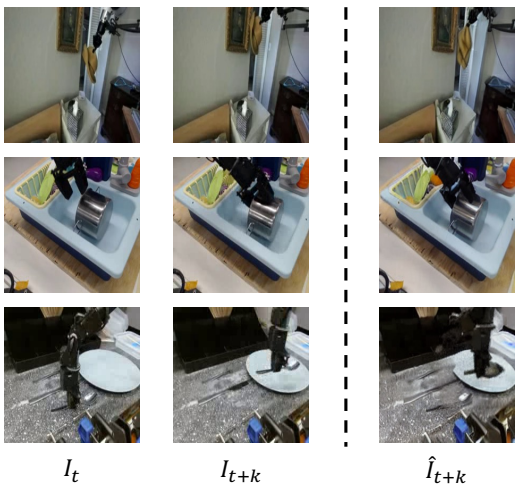


Figure 5: Qualitative results for future frame prediction using a pretrained FDM. Given the current image I_t and the predicted latent action $\hat{z}_t^l$, the model predicts the future frame $\hat{I}_{t+k}$.

is predicted from the history $z^l_{:t-1}$, it still yields a consistent future-frame prediction $\hat{I}_{t+k}$. This indicates that the predicted latent actions retain meaningful motion information, and that HiLAM implicitly models temporal dynamics through next-latent prediction.

5 Conclusion and Limitations

We present HiLAM, a hierarchical latent action model that learns temporally extended latent skills from sequences of low-level latent actions. Unlike prior work, our approach extracts high-level motion structure from actionless videos without requiring action labels or pre-defined skill sets. By leveraging a hierarchical architecture with dynamic chunking, HiLAM segments variable-length trajectories and encodes each segment into a representative latent skill. These learned skills improve downstream performance, particularly on long-horizon and multi-stage tasks, while preserving interpretability through next-latent prediction and future frame prediction. Finally, we demonstrate that using the discovered skills to pretrain a hierarchical policy yields significant data efficiency during fine-tuning.

While our work focuses on encoding latent skills from latent action sequences that capture low-level motion patterns, incorporating language represents a promising direction for future research. Motion cues and language instructions provide orthogonal rather than parallel information. Utilizing both signals could lead to a complementary synergy rather than one replacing the other. For example, imitating motion is often more effective for complex tasks such as furniture assembly, whereas following language instructions can enhance generalizability by imposing fewer constraints compared to specific motion cues. Therefore, combining hierarchical latent action modeling with natural language is a promising future step.

Limitations. While HiLAM introduces a novel approach for skill discovery, our experiments are primarily conducted in simulated environments such as LIBERO. Validating the framework through real-world experiments would further demonstrate the effectiveness of the proposed method. Additionally, to ensure computational efficiency during temporal modeling, HiLAM utilizes a pretrained IDM. However, training the entire architecture end-to-end could potentially lead to a deeper joint understanding of both low-level motion patterns and high-level skills.

References

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-JEPA: Latent video prediction for visual representation learning, 2024.

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 *Forty-first International Conference on Machine Learning*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems*, 2025.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. In *Forty-second International Conference on Machine Learning*, 2025.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pp. 5842–5850, 2017.

Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. BAKU: An efficient transformer for multi-task policy learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Sukjun Hwang, Brandon Wang, and Albert Gu. Dynamic chunking for end-to-end hierarchical sequence modeling. *arXiv preprint arXiv:2507.07955*, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. In *9th Annual Conference on Robot Learning*, 2025.

Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, qiang liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *6th Annual Conference on Robot Learning*, 2022.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Lyubaykin Nikita, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. In *Forty-second International Conference on Machine Learning*, 2025.

Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pp. 188–204. PMLR, 2021a.

Karl Pertsch, Youngwoon Lee, Yue Wu, and Joseph J Lim. Demonstration-guided reinforcement learning with learned skills. In *5th Annual Conference on Robot Learning*, 2021b.

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. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *The Twelfth International Conference on Learning Representations*, 2024.

Lucy Xiaoyang Shi, brian ichter, Michael Robert Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. In *Forty-second International Conference on Machine Learning*, 2025.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

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 *International Conference on Learning Representations*, 2025.

Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.