2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

RynnVLA-001: Using Human Demonstrations to Improve Robot Manipulation

Yuming Jiang[1], Siteng Huang[1,2], Shengke Xue[1,2], Yaxi Zhao[1], Jun Cen[1], Sicong Leng[1], Kehan Li[1], Jiayan Guo[1], Kexiang Wang[1], Mingxiu Chen[1,2], Fan Wang[1], Deli Zhao[1,2] and Xin Li[1,2]

***Abstract*— This paper presents RynnVLA-001, a vision-language-action (VLA) model built upon large-scale video generative pretraining from human demonstrations. We propose a novel two-stage pretraining methodology. The first stage, Ego-Centric Video Generative Pretraining, trains an Image-to-Video model to predict future frames based on an image and a language instruction. The second stage, Human-Centric Trajectory-Aware Modeling, extends this by jointly predicting future keypoint trajectories, thereby bridging visual frame prediction with action prediction. Furthermore, to enhance action representation, we propose ActionVAE, a variational autoencoder that compresses sequences of actions into compact latent embeddings, reducing the complexity of the VLA output space. When finetuned on the same downstream robotics datasets, RynnVLA-001 achieves superior performance over state-of-the-art baselines, demonstrating that the proposed pretraining strategy provides a more effective initialization for VLA models.**

I. Introduction

The past few years have witnessed rapid progress in large language models [1], [2], [3], [4], [5], [6], large multimodal models [7], [8], [9], [10], vision-based recognition models [11], [12], [13], [14], [15], and generative models [16], [17], [18], [19]. The success in these fields is attributed to the availability of large-scale datasets. For instance, large language models benefit from abundant training data readily accessible from web sources. In contrast, progress in Vision-Language-Action (VLA) models is constrained by the scarcity of large-scale robot manipulation data. Collecting such data typically relies on human teleoperation on physical robots to record manipulation trajectories, making large-scale dataset construction both labor-intensive and costly. There have been some early attempts to address the challenges of data scarcity. Some methods propose to build large-scale robot manipulation datasets [20], [21], [22], [23], which collect manipulation data under diverse environment and even with different embodiments. However, the size of these datasets still remains far smaller than those used in LLMs, VLMs, and generative models. Another line of studies works on exploiting prior knowledge from pretrained generative models [24], [25] or vision-language models [26], [27], [28], [29], [30] to alleviate the data scarcity.

In this work, we propose RynnVLA-001, a VLA model enhanced by video generation pretraining. The key insight of RynnVLA-001 is to implicitly transfer the manipulation skills learned from human demonstrations in ego-centric videos to robot manipulation. The overall training pipeline is shown in Fig. 1. In the first stage, **Ego-Centric Video Generative Pretraining**, we train an Image-to-Video (I2V) model that takes a single image and a language instruction as inputs and predicts subsequent frames. To capture general manipulation dynamics, this stage relies on ego-centric human manipulation videos, which emphasize first-person hand operations. Trained with ego-centric videos, the model is capable of predicting manipulations at the visual level. However, a gap remains between the high-level visual observations and the low-level action spaces required to control real robots. To bridge the gap, we introduce another stage of **Human-Centric Trajectory-Aware Video Modeling**, where we further train the I2V model on ego-centric videos paired with human keypoint annotations. In this stage, in addition to future frames, the model is also trained to predict the keypoint trajectories in future frames conditioned on current observations and language instructions. These keypoint-based patterns share similarities with robot actions, thereby facilitating the transfer from visual dynamics to robot manipulation with low-level actions.

Following previous pretraining stages, we further adapt the model using self-collected robot datasets. In this phase, the model is trained to predict action chunks rather than a single-step action, conditioned on RGB observations and language instructions. To ensure the smoothness and temporal coherence of predicted actions, we propose **ActionVAE**, a variational autoencoder that encodes action chunks into compact embeddings. Once trained, the ActionVAE is fixed and employed to extract latent representations of future actions. The model is then optimized to predict these action embeddings alongside future visual observations. During inference, given an observation and a language instruction, the model outputs a single action embedding, which is subsequently decoded by ActionVAE into a sequence of executable robot actions.

Our proposed RynnVLA-001 enables a robot arm to execute complex pick-and-place and long-horizon tasks by accurately following high-level language instructions. To evaluate the effectiveness of the pretraining weights of RynnVLA-001, we compare our proposed RynnVLA with state-of-the-art models, including GR00T-N1.5 [31] and Pi0 [29], by finetuning on the same robot manipulation data. Our proposed RynnVLA-001 consistently achieves higher success rates, demonstrating that the proposed pretraining framework provides more effective initialization for VLA modeling.

[1]DAMO Academy, Alibaba Group, [2]Hupan Lab, *Corresponding: xinting.lx@alibaba-inc.com

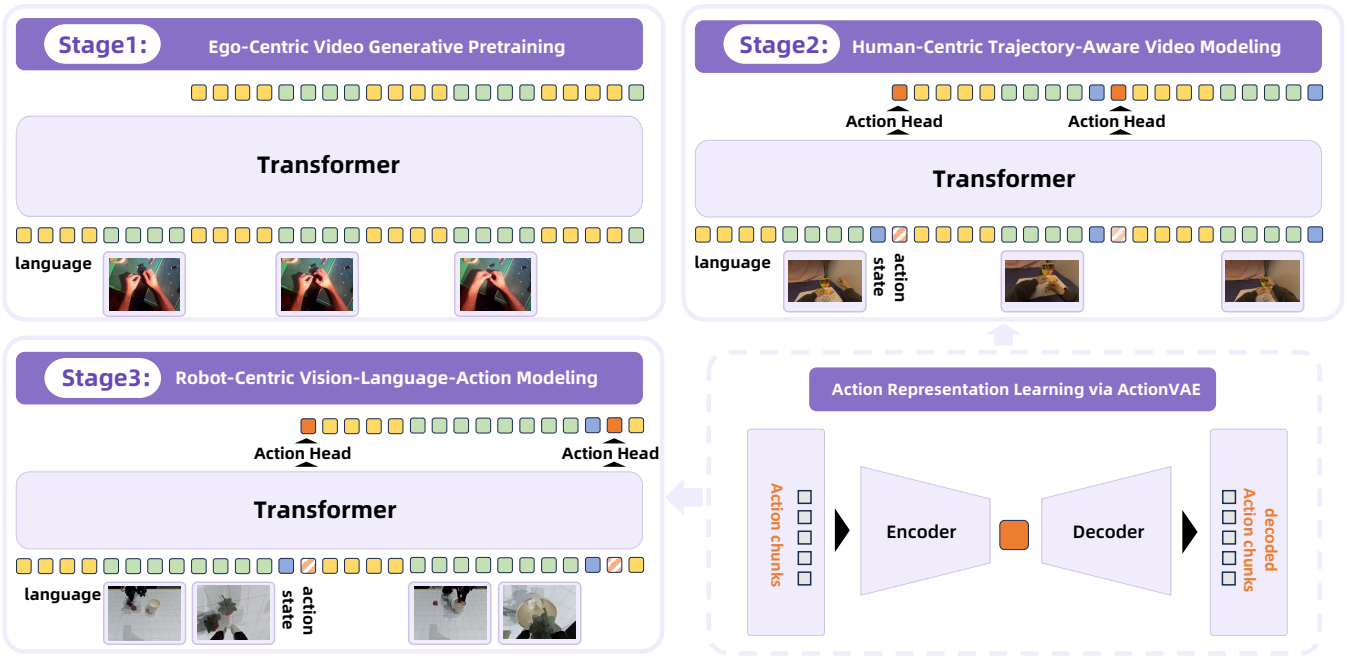


Fig. 1: **Training Pipelines of RynnVLA-001.** The training pipeline consists of: 1) Ego-Centric Video Generative Pretraining to learn future frame prediction. 2) Human-Centric Trajectory-Aware Video Modeling to incorporate action trajectory prediction. 3) Robot-Centric VLA Modeling, where the final model inherits the weights of pretraining stages and uses vision and language inputs to generate action embeddings that are decoded into robot actions. The action embeddings are extracted by ActionVAE. For human-centric trajectory-aware video modeling and robot-centric VLA modeling, besides action embeddings, we also include the state embeddings(blue blocks) as input.

II. Related Work

Vision-Language-Action Models. The development of intelligent robotic agents increasingly relies on harnessing the sophisticated capabilities of pre-trained Vision-Language Models (VLMs), enabling embodied systems to effectively perceive, reason, and act within complex environments. In this context, RT-2 [26] pioneered the concept of vision-language-action models (VLAs), which co-fine-tuned VLMs with both robotic trajectory data and large-scale Internet vision-language tasks. Following RT-2's strategy of discretizing continuous actions into bins for compatibility with LLM decoders, OpenVLA [27] advanced generalization through extensive pretraining on the Open X-Embodiment dataset [20], and FAST [32] further explored the use of the discrete cosine transform to facilitate efficient and scalable training. However, the inherent precision loss associated with action discretization has prompted a shift towards approaches that integrate policy heads specifically designed for continuous action generation. LCB [33] pioneered the adoption of such a dual-system structure, leveraging a pre-trained 3D Diffusion Actor [34] as the policy head for LLaVA [35]. HiRT [36] optimized real-time interaction by operating VLMs at lower frequencies while maintaining high-frequency vision-based control. CogACT [28] applied a diffusion transformer (DiT) to model complex and temporally-correlated actions. DexVLA [37] incorporated diffusion action experts with embodiment curriculum learning across diverse robot platforms. Pi0 [29] integrated PaliGemma [38] with conditional flow matching [39] for continuous action generation. More recently, GR00T N1 [30] presented an end-to-end trained dual-system specifically tailored for humanoid robots. GR00T N1.5 [31] further improved GR00T N1 by using an advanced VLM with more powerful grounding capabilities. Besides, OpenHelix [40] provided a comprehensive summary of existing dual-system architectural designs and conducted systematic empirical evaluations of their core design elements.

Future Prediction for Robot Learning. Existing VLM methods are limited in inadequate modeling of visual dynamics, given their typical dependence on only one or two sampled images from the current observation. In light of this perspective, a stream of research has consequently focused on incorporating future prediction to explicitly model physical dynamics and improve policy learning. Early contributions in this area, such as SuSIE [41], formulated their control policies based on a future keyframe predicted by a text-conditioned image editing model [42]. Similarly, UniPi [43] approached inverse dynamics by considering two generated future frames. While these pioneering efforts leveraged single-step future predictions to guide action generation, they inherently struggled to fully capture the rich complexity of continuous physical dynamics over extended horizons. Recognizing this limitation, more recent endeavors have explored multi-step or autoregressive prediction. GR-1 [44] generated future frames and corresponding actions in an autoregressive fashion, while suffering from poor prediction quality. Advancing beyond this, PAD [45] utilized diffusion models

to concurrently forecast future images and multi-step action sequences, offering a more robust approach to sequential prediction. GR-2 [24] emphasized extensive pretraining on a vast collection of internet-sourced videos, subsequently fine-tuning for both video generation and action prediction using real robotic trajectories. DREAMGEN [46] utilizes neural trajectories, synthetic robot data generated from video world models, for training robot policies that generalize across behaviors and environments. Seer [47] proposed an end-to-end framework that predicts actions by conditioning inverse dynamics models on a series of forecasted visual states. VPP [25] leveraged representations refined from large-scale video foundation models to establish a generalist robotic policy, and GEVRM [48] integrated Open-Sora [49] for highly expressive pre-execution goals.

III. Methodology

As shown in Fig. 1, the training of RynnVLA-001 consists of three stages: **1) Ego-Centric Video Generative Pretraining**: An ego-centric I2V model is trained using ego-centric human manipulation videos. This stage enables the model to predict future frames. **2) Human-Centric Trajectory-Aware Video Modeling**: A trajectory-aware ego-centric video generation model is then trained to predict the future frames as well as keypoint trajectories. The motivation of this stage is to bridge the gap of the prediction spaces between I2V model and the final VLA model. **3) Robot-Centric Vision-Language-Action Modeling**: A VLA model is trained by inheriting weights of the aforementioned pretraining models. Language instructions and the current observations are fed into the VLA model, the VLA model is expected to generate the action embedding, which is further decoded by the pretrained ActionVAE to generate actions for robot arms to execute.

A. Ego-Centric Video Generative Pretraining

The challenge in scaling up VLA models lies in that large-scale paired data for VLA training is hard to collect. In this work, we transfer priors learned from human demonstrations in video pretraining to VLA models. This stage is to train an I2V model that closely mimics the inference process of a VLA model. In a typical VLA setting, actions are predicted conditioned on current observations (*e.g.*, visual inputs, robot states) and a language instruction. Accordingly, our I2V model is trained to predict future video frames based on an initial visual observation and a corresponding language description. This pretraining task forces the model to learn the physical dynamics of object manipulation from an ego-centric perspective.

For the network architecture, we adopt an autoregressive (AR) Transformer. Due to the limited availability of AR-based video generation frameworks, we extend a powerful AR image generation model, Chameleon [50], [51], to perform the image-to-video task. To ensure the learned priors are directly relevant for robotic manipulation, we curated 11.93M ego-centric human manipulation videos for training. These videos contain first-view human operations and focus on hand manipulations. Besides, we also filter 244K robotic manipulation videos from open-source datasets. These videos feature human hand manipulations that are analogous to robot gripper movements and operations. As shown in Fig. 1, instead of providing the instruction only once, we interleave the language tokens with the visual tokens of the video frames. The input sequence format is [language, visual tokens$_t$, language, visual tokens$_{t+1}$, ...]. The repetition of language tokens is to mimic the inference scenarios of VLA models, where each action prediction is based on language tokens and visual tokens. The training is supervised by the cross-entropy loss over discrete visual tokens and language tokens.

B. Human-Centric Trajectory-Aware Video Modeling

While the I2V model from Stage 1 learns to predict dynamics at the visual level, it lacks an explicit understanding of actions. To bridge this gap between purely visual prediction and action generation, Stage 2 refines the pretrained model into a trajectory-aware framework. The core idea of this stage is to finetune the pretrained model to concurrently predict future visual frames and the corresponding human keypoint trajectories, where human trajectories can be regarded as another form of the actions. By learning to associate visual changes with their underlying motion trajectories, the model develops a more holistic understanding of dynamics of manipulations.

To this end, we utilize the EgoDex dataset [52], which provides trajectories of all upper-body joints captured via Apple Vision Pro devices. From all the joints provided, we selectively use only the wrist keypoints, as these can mimic the end-effector positions of robots. Crucially, rather than predicting raw coordinates, the model is trained to predict a compact, continuous embedding for a chunk of trajectory data. This dense representation is generated by a pretrained ActionVAE (detailed in Stage III-C), which effectively compresses the action sequence.

To provide the model with proprioceptive information, we introduce state embeddings (blue blocks in Fig. 1). These embeddings represent the current keypoint positions of the human wrists and are fed into the model at each timestep. A linear layer is employed to project the dimension of states into that of transformer's input feature. The input sequence is now structured as: [language, visual tokens$_t$, state embedding$_t$, <ACTION_PLACEHOLDER>, ...], where <ACTION_PLACEHOLDER> is the signal to generate continuous action embeddings. This sequence explicitly provides the model with three crucial pieces of information: the high-level goal (language), the current visual scene (visual tokens), and the current physical configuration of the arm (state embedding).

The architecture from Stage 1 is extended to handle the new and continuous prediction target. While the Transformer backbone remains, a lightweight action head (a single linear layer) is introduced. The Transformer's main output remains discrete visual tokens, whereas the action representation is a continuous embedding. The action head takes the Transformer's last hidden state and maps it to

the continuous latent space of the action embeddings. The training of the action head is supervised by L1 loss, which is computed exclusively for the outputs at token positions of <ACTION_PLACEHOLDER>. The prediction of visual tokens remains the same as Stage 1.

C. ActionVAE: Action Representaton via VAE

In VLA models, predicting action chunks (*i.e.*, short sequences of actions) is more effective than predicting single, step-by-step actions. This design choice is motivated by two key factors: 1) Avoiding repetitive predictions: Single-step action execution often results in negligible visual changes, which can cause the model to repeatedly output the same action and become stuck. Predicting a chunk encourages more substantial progress. 2) Efficiency: Generating multiple actions in a single forward pass reduces computational overhead and inference latency.

To facilitate this chunk-level prediction while ensuring the generated actions are smooth and coherent, we introduce an Action Variational Autoencoder (ActionVAE). As illustrated in Fig. 1, the ActionVAE consists of an encoder that compresses an "action chunk" into a compact, continuous latent embedding, and a decoder that reconstructs the original action sequence from this embedding.

Since our training pipeline involves both human demonstrations and robot executions, and their kinematic spaces differ, we train two domain-specific ActionVAEs: one for compressing human trajectories (used in Stage 2) and another for compressing robot actions (used in Stage 3). This ensures that each domain has a tailored and accurate action representation. Importantly, the ActionVAE is embodiment-specific. As it encodes chunk-level actions that often correspond to atomic motion primitives, a well-trained model can be directly used to extract action embeddings from new data on the same embodiment without retraining.

D. Robot-Centric Vision-Language Action Modeling

In the final stage, we adapt the pretrained trajectory-aware model into a VLA model for robot control. This is achieved by integrating the robot-specific ActionVAE representations and finetuning the model on robot-centric data. The primary objective is to predict the embedding of the next robot action chunk, which is then decoded by the ActionVAE into an executable action sequence.

The architecture largely inherits the multi-task framework from Stage 2 but is now adapted for the robot domain. A key distinction lies in the action head. Since human hand trajectories differ substantially from robot arm kinematics, the action head pretrained in Stage 2 is discarded. Instead, a new lightweight action head (a single linear layer) is initialized to predict robot action embeddings.

The input sequence for the VLA model is structured to mirror the real-world robotic deployment scenario, using the same placeholder-based design defined in our pretraining stages: [language, front tokens$_t$, wrist tokens$_t$, robot state embedding$_t$, <ACTION_PLACEHOLDER>, ...] In our setting, for visual observations, we will have two views, one is the front camera view and the other is the wrist camera view. During the forward pass, the model processes this sequence and is trained on two concurrent objectives: 1) Robot Action Prediction: The hidden state corresponding to the output of the <ACTION_PLACEHOLDER> token is fed into the newly initialized action head to regress the hidden state to a continuous embedding. The training of this head is supervised by an L1 loss between the predicted embedding and the ground-truth embedding from the robot-specific ActionVAE. 2) Future Visual Prediction: The model also continues its pretraining task of autoregressively predicting the visual tokens for the next frame. This part of the training is supervised by a cross-entropy loss.

E. Inference

During inference, the VLA model operates within a closed-loop cycle to perform tasks. At each step, the model receives the language instruction, the current RGB observations from the robot's cameras and current robot states as inputs. To optimize for efficiency, at inference time, the model only predict the action embedding, and discards the generation of future vision tokens, which is computationally expensive and unnecessary for control. The predicted action embedding is then passed to the frozen decoder of the ActionVAE. The decoder reconstructs a chunk of low-level robot actions from this single embedding. The robot executes this entire action chunk. Upon completion, a new observation is captured and fed back into the model along with the language instruction, initiating the next cycle. This process repeats until the task is successfully completed.

IV. Experiments

A. Experimental Setup

Dataset. To train and evaluate our proposed RynnVLA-001 model, we collected a new real-world manipulation dataset using a LeRobot SO100 robotic arm. The entire dataset comprises demonstrations collected through human teleoperation. To ensure our dataset covers a diverse range of manipulation skills, we designed and collected data for three representative tasks: 1) Pick up and place green blocks, focusing on fundamental object recognition and grasping abilities. We collected 248 demonstrations. 2) Pick up and place strawberries, requiring precise localization and grasp point estimation, challenging the model's fine-grained perception. We collected 249 demonstrations. 3) Grab pen and put it into holder, demanding advanced 3D spatial reasoning, specifically the ability to infer object orientation and height for a precise insertion action. We collected 301 demonstrations. To enhance the richness and complexity of the data, we introduced significant variations during collection. The scenes vary from containing only target objects to more complex arrangements that include other irrelevant, distractor objects. During teleoperation, the human operator's goal was to move all target objects to their destination. Furthermore, the data was collected using three different SO100 arms across various environments.

TABLE I: **Performance comparison on three manipulation tasks.** We report task-specific success rates, average success rate over three tasks, and success rate@1. Each number represents the average across the three evaluation settings in Tab. II.

Method	Task-Specific Success Rate (%)			Average (%)	Success@1 (%)
	Pick up and place green blocks	Pick up and place strawberries	Grab pen and put it into holder		
GR00T N1.5	65.0	53.3	48.3	55.6	37.2
Pi0	75.6	71.1	64.4	70.4	56.3
RynnVLA-001 (Ours)	**90.0**	**91.7**	**90.0**	**90.6**	**56.7**

TABLE II: **Performance comparison on three different evaluation settings.**

Method	Single-Target Manipulation	Multi-Target Manipulation	Instruction-following with Distractor Objects
GR00T N1.5	63.3	46.7	56.7
Pi0	80.0	71.1	60.0
RynnVLA-001 (Ours)	**93.3**	**86.7**	**91.7**

Baselines. We compare our model with two strong open-source baseline, GR00T N1.5 [31] and Pi0 [29]. We initialize the model with the pretrained weights and then finetune the model with the same SO100 data as our model. We use the official code of GR00T N1.5 and Pi0 and strictly follow the instructions to finetune the model.

Evaluation. We evaluate the performance under three different scenarios: 1) Single-target Manipulation, where only a single target object is on the desktop, 2) Multi-target Manipulation, where multiple target objects are on the desktop, 3) Instruction-following with Distractors, where target objects and distractor objects appear on the desk. For all scenarios, a trial is considered a success if the model correctly places at least one target object in its target location within a predefined time limit. A trial is marked as a failure under any of the following conditions: 1) The time limit is exceeded. 2) The model makes more than five consecutive failed attempts to grasp a target object. 3) Specifically for the Instruction-Following with Distractors scenario, the model attempts to manipulate any distractor objects. We report the success@1 metric, defined as the percentage of tasks successfully completed within a single uninterrupted trial. To evaluate generalization, each task is evaluated on multiple robotic arms, each operating in a unique physical environment.

B. Comparison with SoTA methods

Table I presents a detailed comparison of task-specific and average success rates. Our model, RynnVLA-001, demonstrates substantially higher overall performance, outperforming both GR00T N1.5 and Pi0 across all three tasks. As for the success rate@1 metric, the performance of our proposed RynnVLA-001 is comparable to Pi0. The relatively low success@1 rates for all three models suggest that further improvements in object localization accuracy are necessary for achieving reliable single-trial success.

In Table II, we report success rates in three different evaluation settings. The task becomes more challenging when more objects are introduced. For GR00T N1.5, the success rates of multi-target manipulation and instruction-following with distractor objects become lower than that of single-target manipulation. For Pi0, when distractor objects appear on the desk, the success rates drops significantly, which indicates the instruction-following capability is limited. In contrast, the performance of our proposed RynnVLA-001 remains stable across three settings.

C. Effectiveness of Pretraining Weights

In RynnVLA-001, we propose two pretraining stages: 1) ego-centric video generative pretraining and 2) human-centric trajectory-aware video modeling. To investigate the effectiveness of the two-stage pretraining pipeline, we conduct ablation studies as shown in Table III.

First, we evaluate the impact of Stage 1: Ego-centric Video Generative Pretraining. We compare three initialization strategies for the final VLA model: 1) RynnVLA-001-Scratch: A baseline initialized from random weights, skipping all pretraining. 2) RynnVLA-001-Chameleon: A stronger baseline initialized directly from the pretrained weights of the Chameleon Text-to-Image model [50], thus bypassing our video pretraining stage. 3) RynnVLA-001-Video: Our model after completing Stage 1, which starts from Chameleon weights but is further pretrained on ego-centric videos. The results clearly demonstrate the importance of video-centric pretraining. The RynnVLA-001-Scratch model is incapable of correlating language instructions with meaningful actions, resulting in an extremely low success rate. The RynnVLA-001-Chameleon baseline, benefiting from T2I pretraining, can perform some basic tasks. However, it exhibits a limited localization capability, capping its performance at a success rate of 50.0%. In contrast, RynnVLA-001-Video achieves a significant performance improvement, indicating that priors learned from ego-centric videos are crucial for effective VLA finetuning.

Next, we build upon this successful video-pretrained model to evaluate the contribution of Stage 2: Human-centric Trajectory-Aware Video Modeling. By incorporating this second pretraining stage where the model learns to predict human trajectories, our full model, RynnVLA-001, achieves the best performance among all variants. This final improvement shows the benefit of explicitly bridging the gap between visual prediction and action generation by pretraining the model to predict human trajectories.

D. Ablation Study on Model Designs

To systematically evaluate the impact of key components, we conduct a series of ablation studies on the Calvin Benchmark. For experimental efficiency, our baseline model

TABLE III: **Effectiveness of Pretrained Weight.** We train four variants of RynnVLA-001 with different initializations.

Method	Task-Specific Success Rate (%)			Average (%)
	Pick up and place green blocks	Pick up and place strawberries	Grab pen and put it into holder	
RynnVLA-001-Scratch	0	6.7	6.7	4.4
RynnVLA-001-Chameleon	56.6	50.0	43.3	50.0
RynnVLA-001-Video	81.7	86.7	85.0	84.4
RynnVLA-001 (Full)	**90.0**	**91.7**	**90.0**	**90.6**

TABLE IV: **Ablation Study of VLA Components on the Calvin Benchmark.** All models were trained with reduced epochs for efficiency; scores are for relative comparison.

Method	Task	Task Success Rate (%)					Avg. Len.
		1	2	3	4	5	
256 × 256	Task ABC - D	92.7	83.7	73.5	62.1	53.2	3.652
Raw Actions Prediction	Task ABC - D	93.8	86.5	80.4	74.2	67.0	4.019
Deeper Action Head	Task ABC - D	90.2	77.9	65.3	54.6	44.3	3.323
Full Model	**Task ABC - D**	**95.4**	**88.2**	**82.2**	**78.2**	**72.1**	**4.161**

and its ablated variants are trained for a reduced number of epochs, initialized from weights pretrained on RynnVLA-001-Video. We also modified the evaluation for the "place in slider" task because the original prompt, "store the grasped block in the sliding cabinet", resulted in extremely low performance. To better assess action prediction capabilities, we revised it to "place the grasped object in the sliding cabinet". Consequently, the results presented in this section are intended for comparative analysis to demonstrate the relative importance of each component.

Image Resolution. In this study, we investigate the impact of image resolution on our video-pretrained VLA model. As shown in Tab. IV, a substantial performance drop was observed when the resolution was decreased from our proposed 384×384 to 256×256. This degradation is attributed to the resolution mismatch with the VQGAN component, which was pretrained exclusively on 512×512 images. At a lower resolution of 256×256, the VQGAN's reconstruction quality degrades, the VQGAN fails to generate high-fidelity reconstructions, resulting in imprecise visual tokens that cannot faithfully represent the source content. Consequently, a VLA model trained on these imprecise tokens exhibits reduced performance. Furthermore, our choice of 384×384 strikes a balance: 1) it maintains high reconstruction fidelity by using the resolution closer to the VQGAN's native resolution. 2) it offers a significant reduction in computational overhead compared to the 512×512 resolution, making it a more practical choice for deployment.

Action Representations. In this work, we propose using ActionVAE to compress action chunks into compact latent embeddings. This approach contrasts with prior methods that directly predict raw action sequences. To evaluate our method, we conducted an ablation study on the Calvin ABC->D benchmark, comparing the performance of predicting VAE embeddings against predicting raw actions. As shown in Tab. IV, predicting actions in the VAE's latent space outperforms the direct prediction of raw actions. The performance gain stems from two key advantages by using ActionVAE: 1) it provides an efficient and compressed representation of complex action sequences, and 2) the inherent structure of its latent space promotes temporal consistency, yielding smoother predicted actions.

Size of Action Head. Our action prediction module utilizes a simple action head: a single linear layer that projects the transformer's final hidden state into the action embedding space. To assess the impact of head complexity, we performed an ablation study comparing this design to a deeper, five-layer MLP head on a subset of the Calvin Task ABC->D benchmark. As shown in Tab. IV, increasing the head's depth is surprisingly detrimental to performance, causing the evaluation score to decrease substantially from 4.019 to 3.323. This result indicates that the transformer's output representation is already highly effective for the task. A direct linear mapping is sufficient for decoding, while the additional complexity of a deeper head appears to introduce noise or overfitting, ultimately impairing performance. This underscores the value of architectural simplicity in the action prediction stage of our model.

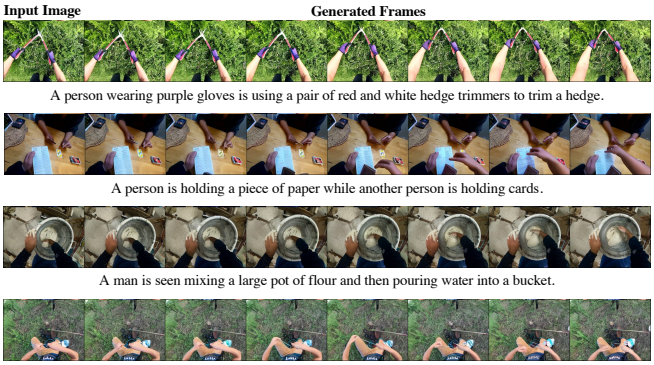


Fig. 2: **Visualization of Video Generative Pretraining.**

E. Further Analysis

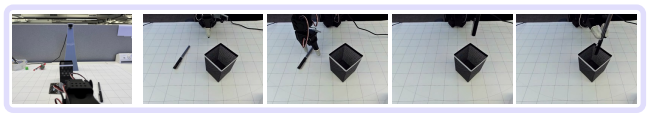
Visualization of Video Pretraining Model. The first stage involves pretraining an ego-centric Image-to-Video (I2V) model. This I2V paradigm is chosen to align with the typical input for Vision-Language-Action (VLA) models: an initial image observation and a text-based instruction. As illustrated in Fig. 2, the pretrained model can generate video frames with plausible motion and consistent content from a given image and text prompt. Although the model is prone to generate subtle visual changes between frames, we find it sufficient for its role as a pretrained backbone for the subsequent VLA training stage.

Improving Instruction-Following Capabilities through Data Collection. Our evaluation protocol for instruction-following capabilities involves placing distractor objects in the desktop to test the model's robustness against visual ambiguity. We hypothesize that training exclusively on data with isolated target objects leads to a simplistic and vision-driven policy, where the model learns to grasp any objects without performing actions in the provided language instruction. To validate this hypothesis, we perform an ablation study. A variant of RynnVLA-001 is trained solely on data

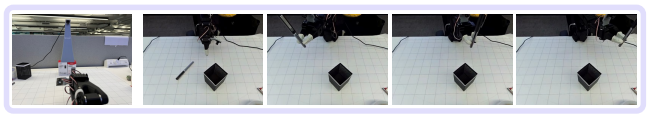
(a) Views with disabled front camera

(b) Trajectory of wrist camera views when grab and place the strawberries

Fig. 3: **Analysis on the front camera's function for coarse localization.** (a) The front camera is masked, leaving only the wrist camera effective. (b) The robot can still complete the task if the target is within the wrist camera's initial field of view. However, task success rate drops to 0% when the target is outside the wrist camera's view (on the left side).



(a) Normal camera setting and trajectories of front camera when grab and put pens



(b) High front camera setting and trajectories of front camera when grab and put pens

Fig. 4: **The front camera provides 3D projective information for manipulation.** (a) With the standard camera configuration, the robot can insert the pen into the holder. (b) When the front camera is elevated, the altered projective geometry of the scene causes the model to fail the task.

without distracted objects. When evaluated on the task "pick up the strawberry" in a scene cluttered with pens and green blocks, this ablated model demonstrates a 0% success rate over 10 trials. A total of 5 failure cases of the 10 trials consistently select a distractor object. In contrast, our full RynnVLA-001 model, trained on our comprehensive dataset including distractors, achieves a 90% success rate (9/10) on this task. These results quantitatively underscore the critical importance of diverse data collection with distractors for developing reliable language-conditioned VLA models.

Functional Analysis of Front and Wrist Cameras. We hypothesize that the front camera provides coarse object localization and 3D projective context, while the wrist camera is responsible for precise local adjustments. 1) To validate the front camera's role in coarse localization, we perform the evaluation where it is disabled. When the front camera is masked (Figure 3(a)), the model could still succeed as long as the target is within the wrist camera's initial field of view (Figure 3(b)). However, if the target (*e.g.*, strawberries on the left) is outside the wrist camera's view, the robot fails to initiate any action. Quantitative results confirm this: for targets on the right, the success rate drops slightly from 100% (5/5) to 80% (4/5) after masking. For targets on the left, the rate degrades from 80% (4/5) to 0%. These findings strongly suggest that the front camera's primary function is to guide the end-effector to the target's general location. 2) Furthermore, we explore the front camera's function in providing 3D information for tasks requiring depth perception, such as inserting a pen into a holder. As shown in Figure 4(a), the robot succeeds with the normal camera setup. However, when we elevate the front camera, altering the scene's projective geometry, the model fails to correctly insert the pen (Figure 4(b)). This shows that the front camera provides critical 3D projective information that the model relies on for spatial reasoning and manipulation.

V. Discussion and Conclusion

In this work, we propose RynnVLA-001, a VLA model enhanced by human demonstrations. We introduce human demonstrations in two pretraining stages. The first stage trains an I2V model by learning dynamics through predicting next frames. The second stage bridges the gaps between I2V models and VLA models by learning to predict keypoint trajectories of human. Moreover, we propose ActionVAE to embed action chunks into a compacted embeddings. Owing to our dedicated designs, our proposed RynnVLA-001 outperforms state-of-the-art models.

Limitation. In this work, we validate the performance of RynnVLA-001 on the LeRobot SO100 robot arm. However, the scope of our current evaluation presents several limitations. Our experiments are limited to a single robot embodiment and a similar evaluation environment to the training data. Furthermore, the front camera is mounted in a fixed position. To assess and enhance the model's generalization capabilities, future efforts will focus on: (1) extending the evaluation to a more diverse range of robot arms; (2) testing the model in more varied and unstructured environments; and (3) diversifying camera viewpoints.

References

[1] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen, *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[2] Anthropic, "System card: Claude opus 4 and claude sonnet 4." [Online]. Available: https://www.anthropic.com/news/claude-4

[3] OpenAI, "Introducing gpt-4.1 in the api," 2025. [Online]. Available: https://openai.com/index/gpt-4-1/

[4] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[5] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[6] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[7] OpenAI, "Gpt-4o system card," 2024. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[8] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[9] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, *et al.*, "Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models," *arXiv preprint arXiv:2504.10479*, 2025.

[10] D. Guo, F. Wu, F. Zhu, F. Leng, G. Shi, H. Chen, H. Fan, J. Wang, J. Jiang, J. Wang, *et al.*, "Seed1. 5-vl technical report," *arXiv preprint arXiv:2505.07062*, 2025.

[11] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[12] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *CVPR*, 2023, pp. 15 619–15 629.

[13] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[14] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[15] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, *et al.*, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," *arXiv preprint arXiv:2502.14786*, 2025.

[16] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.

[17] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *ICML*, 2024.

[18] K. Tian, Y. Jiang, Z. Yuan, B. Peng, and L. Wang, "Visual autoregressive modeling: Scalable image generation via next-scale prediction," *NeurIPS*, vol. 37, pp. 84 839–84 865, 2024.

[19] J. Baldridge, J. Bauer, M. Bhutani, N. Brichtova, A. Bunner, L. Castrejon, K. Chan, Y. Chen, S. Dieleman, Y. Du, *et al.*, "Imagen 3," *arXiv preprint arXiv:2408.07009*, 2024.

[20] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration," in *ICRA*, 2024, pp. 6892–6903.

[21] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," *arXiv preprint arXiv:2410.07864*, 2024.

[22] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[23] Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang, *et al.*, "Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems," *arXiv preprint arXiv:2503.06669*, 2025.

[24] C. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, *et al.*, "GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation," *arXiv preprint arXiv:2410.06158*, 2024.

[25] Y. Hu, Y. Guo, P. Wang, X. Chen, Y. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen, "Video prediction policy: A generalist robot policy with predictive visual representations," *arXiv preprint arXiv:2412.14803*, 2024.

[26] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "RT-2: vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*, 2023, pp. 2165–2183.

[27] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, *et al.*, "OpenVLA: an open-source vision-language-action model," in *Conference on Robot Learning*, 2024, pp. 2679–2713.

[28] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang, *et al.*, "CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," *arXiv preprint arXiv:2411.19650*, 2024.

[29] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[30] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, Linxi, Y. Fang, D. Fox, F. Hu, S. Huang, *et al.*, "GR00T N1: an open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[31] J. Bjorck, V. Blukis, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, *et al.*, "Gr00t n1.5: An improved open foundation model for generalist humanoid robots," https://research.nvidia.com/labs/gear/gr00t-n1_5/, 2025.

[32] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "FAST: efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.

[33] Y. Shentu, P. Wu, A. Rajeswaran, and P. Abbeel, "From LLMs to actions: Latent codes as bridges in hierarchical robot control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 8539–8546.

[34] T. Ke, N. Gkanatsios, and K. Fragkiadaki, "3D Diffuser Actor: policy diffusion with 3D scene representations," in *Conference on Robot Learning*, 2024, pp. 1949–1974.

[35] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Adv. Neural Inform. Process. Syst.*, 2023, pp. 34 892–34 916.

[36] J. Zhang, Y. Guo, X. Chen, Y. Wang, Y. Hu, C. Shi, and J. Chen, "HiRT: enhancing robotic control with hierarchical robot transformers," in *Conference on Robot Learning*, 2024, pp. 933–946.

[37] J. Wen, Y. Zhu, J. Li, Z. Tang, C. Shen, and F. Feng, "DexVLA: vision-language model with plug-in diffusion expert for general robot control," *arXiv preprint arXiv:2502.05855*, 2025.

[38] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, *et al.*, "PaliGemma: A versatile 3B VLM for transfer," *arXiv preprint arXiv:2407.07726*, 2024.

[39] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *ICLR*, 2023.

[40] C. Cui, P. Ding, W. Song, S. Bai, X. Tong, Z. Ge, R. Suo, W. Zhou, Y. Liu, B. Jia, H. Zhao, S. Huang, and D. Wang, "OpenHelix: A short survey, empirical analysis, and open-source dual-system VLA model for robotic manipulation," *arXiv preprint arXiv:2505.03912*, 2025.

[41] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pretrained image-editing diffusion models," *arXiv preprint arXiv:2310.10639*, 2023.

[42] T. Brooks, A. Holynski, and A. A. Efros, "InstructPix2Pix: learning to follow image editing instructions," in *CVPR*, 2023, pp. 18 392–18 402.

[43] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," in *NeurIPS*, 2023.

[44] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, "Unleashing large-scale video generative pre-training for visual robot manipulation," in *Int. Conf. Learn. Represent.*, 2024.

[45] Y. Guo, Y. Hu, J. Zhang, Y. Wang, X. Chen, C. Lu, and J. Chen, "Prediction with action: Visual policy learning via joint denoising process," in *NeurIPS*, 2024.

[46] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin, *et al.*, "Dreamgen: Unlocking generalization in robot learning through neural trajectories," *ArXiv*, vol. abs/2505.12705, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:278739925

[47] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang, "Predictive inverse dynamics models are scalable learners for robotic manipulation," in *Int. Conf. Learn. Represent.*, 2025.

[48] H. Zhang, P. Ding, S. Lyu, Y. Peng, and D. Wang, "GEVRM: goal-expressive video generation model for robust visual manipulation," in *Int. Conf. Learn. Represent.*, 2025.

[49] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You, "Open-Sora: democratizing efficient video production for all," *arXiv preprint arXiv:2412.20404*, 2024.

[50] C. Team, "Chameleon: Mixed-modal early-fusion foundation models," *CoRR*, vol. abs/2405.09818, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.09818

[51] D. Liu, S. Zhao, L. Zhuo, W. Lin, Y. Qiao, H. Li, and P. Gao, "Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining," *CoRR*, vol. abs/2408.02657, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2408.02657

[52] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang, "Egodex: Learning dexterous manipulation from large-scale egocentric video," *CoRR*, vol. abs/2505.11709, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.11709