FLARE: Robot Learning with Implicit World Modeling

Ruijie Zheng[1,2*], **Jing Wang**[1,3*], **Scott Reed**[1*]
Johan Bjorck[1†], **Yu Fang**[1†], **Fengyuan Hu**[1†], **Joel Jang**[1†], **Kaushil Kundalia**[1†]
Zongyu Lin[1†], **Loic Magne**[1†], **Avnish Narayan**[1†], **You Liang Tan**[1†], **Guanzhi Wang**[1†]
Qi Wang[1†], **Jiannan Xiang**[1†], **Yinzhen Xu**[1†], **Seonghyeon Ye**[1†]
Jan Kautz[1], **Furong Huang**[2], **Yuke Zhu**[1, 4‡], **Linxi Fan**[1‡]
[1]NVIDIA [2]University of Maryland, College Park
[3]Nanyang Technological University [4]University of Texas, Austin
[*]equal contribution [†]alphabetical order [‡]equal advising

***Abstract*—We introduce *Future LAtent REpresentation Alignment* (FLARE*), a novel framework that integrates predictive latent world modeling into robot policy learning. By aligning features from a diffusion transformer with latent embeddings of future observations, FLARE enables a diffusion transformer policy to anticipate latent representations of future observations, allowing it to reason about long-term consequences while generating actions. Remarkably lightweight, FLARE requires only minimal architectural modifications—adding a few tokens to standard vision-language-action (VLA) models—yet delivers substantial performance gains. Across two challenging multitask simulation imitation learning benchmarks spanning single-arm and humanoid tabletop manipulation, FLARE achieves state-of-the-art performance, outperforming prior policy learning baselines by up to 26%. Moreover, FLARE unlocks the ability to co-train with human egocentric video demonstrations lacking action labels, significantly boosting policy generalization to a novel object with unseen geometry with as few as 1 robot demonstration. Our results establish FLARE as a general and scalable approach for combining implicit world modeling with high-frequency robotic control.**

I. Introduction

Several recent works [41, 5, 21, 52, 45, 9] have explored jointly learning world models and policies by generating future visual frames in parallel with actions. While intuitive, this approach faces notable practical and conceptual challenges. High-fidelity visual prediction typically requires large-scale generative models, introducing significant computational overhead and latency. Moreover, optimizing simultaneously for pixel-level reconstruction and action prediction places competing demands on model capacity: visual generation emphasizes detailed spatial fidelity and texture synthesis, whereas action modeling benefits from compact, abstract, task-relevant representations, often leading to diluted learning efficiency. In this work, we show that a surprisingly simple and flexible recipe, fully compatible with existing VLA architectures, can surpass prior SoTA VLA policy learning methods by a substantial margin.

We propose **F**uture **LA**tent **RE**presentation Alignment (**FLARE**), a lightweight yet highly effective extension to diffusion or flow-matching policies that introduces latent-space world modeling via a future alignment objective, eliminating the need for full-frame reconstruction. At its core, **FLARE** predicts a compact representation of the robot's future observation from the hidden states of the action denoising network. **FLARE** operates in two key stages. First, we pretrain a compact, action-aware observation embedding model. While general-purpose embedding models could be used for the target future embeddings, we find that an action-aware embedding explicitly optimized for downstream control tasks offers superior performance and efficiency due to its compactness and task alignment. Next, we co-train the diffusion transformer by introducing a minimal set of additional tokens, which are optimized to predict the future observation embeddings. This approach requires minimal modifications to existing VLA architectures [2, 28], making it broadly applicable and easy to deploy.

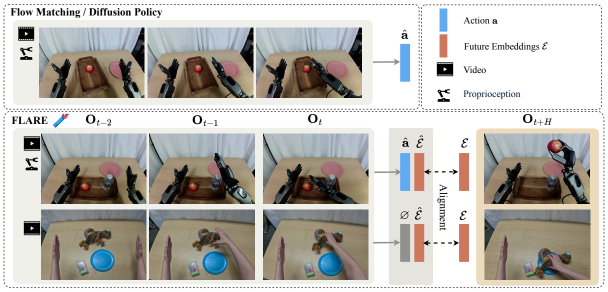


Fig. 1: Comparison of **FLARE** to a conventional flow-matching (or diffusion) policy. **FLARE** can train using both action flow-matching and future latent alignment objectives, leading to improved performance as well as enabling learning from video-only data such as human ego-view demonstrations.

Despite its simplicity, **FLARE** achieves state-of-the-art performance across two multitask imitation-learning benchmarks spanning single-arm and humanoid tabletop manipulation. Notably, when trained on diverse cross-embodiment robot data, our action-aware embedding model generalizes effectively to unseen embodiment and tasks. With just 100 post-training trajectories per task collected on a real GR1 humanoid, the **FLARE** policy achieves a 95% success rate in real-world evaluations. Finally, **FLARE** enables learning from action-free data sources, such as human egocentric videos. By leveraging GoPro-collected human demonstrations and only a single real

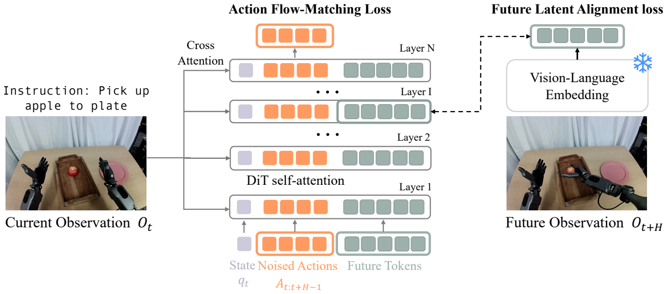

Action Flow-Matching Loss
Future Latent Alignment loss
Cross Attention
Layer N
Layer I
Layer 2
Layer 1
DiT self-attention
Instruction: Pick up apple to plate
Current Observation O_t
Vision-Language Embedding
Future Observation O_{t+H}
State q_t
Noised Actions $A_{t:t+H-1}$
Future Tokens

Fig. 2: FLARE architecture. State and action token embeddings are concatenated into a sequence with learnable future token embeddings. The flow matching DiT blocks perform self-attention on this sequence, and cross-attention to the current vision and text observation embeddings. At a middle layer, the activations corresponding to the future token embeddings are used to compute a future latent alignment loss, which is the cosine similarity with vision-language embeddings from a future observation.

robot demonstration per object, **FLARE** successfully learns novel grasping strategies, highlighting its potential for scalable robot learning from less structured data sources.

II. Background

In this work, following Pi0 and GR00T N1 [2, 28], we adopt **flow-matching** [24] as the learning objective for fitting actions from human demonstrations. Let o_t denote the robot's observation, which includes image inputs (potentially from multiple views) and a language instruction; let q_t be the robot's proprioceptive state; and let $A_t = (a_t, \ldots, a_{t+H})$ be an action chunk drawn from expert demonstrations. We define $\phi_t = VL(o_t)$ as the vision-language embedding of the observation.

Given the VL embedding ϕ_t, an action chunk A_t, a flow-matching timestep $\tau \in [0, 1]$, and sampled noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we construct the noised action chunk as:

$$A_t^\tau = \tau A_t + (1 - \tau)\epsilon.$$

Then the model prediction $V_\theta(\phi_t, A_t^\tau, q_t)$ is trained to approximate the denoising direction $\epsilon - A_t$. Following GR00T N1 [28], we use the same Diffusion Transformer (DiT) architecture [31] for V_θ with alternating cross-attention and self-attention layers to condition on the robot's vision language embedding ϕ_t.

III. Method

A. Latent World Modeling through Future Latent Representation Alignment

To enable the latent representation within the DiT blocks to predict future latent states, we add M learnable future token embeddings to the input sequence, such that the sequence contains three components: (1) the current proprioceptive state q_t encoded via a state encoder, (2) noised action chunk $A_t^\tau = \{\tau a_t + (1 - \tau)\epsilon\}_t^{t+H}$ encoded by an action encoder, and (3) a set of M learnable future tokens. Next, we slice out the intermediate DiT representations corresponding to the M future tokens at an internal layer L, project those features using an MLP, and finally align these with the frozen vision-language embeddings of the *future* observation ϕ_{t+H} (See Figure 2). In this way, we encourage the DiT modules to internally reason about the future latent state while maintaining its action prediction capability through action flow-matching. Letting B indicate batch dimension and D indicate embedding dimension, we can write the latent alignment objective as

$$\mathcal{L}_{align}(\theta) = -\mathbb{E}_\tau \left[cos(f_\theta(\phi_t, A_t^\tau, q_t), g(\phi_{t+H})\right] \tag{1}$$

where $f_\theta \rightarrow \mathbb{R}^{B\times M\times D}$ outputs the DiT activations for the M future tokens at layer L, and $g \rightarrow \mathbb{R}^{B\times M\times D}$ is the encoder of the future observation ϕ_{t+H}. The overall loss function is

$$\mathcal{L} = \mathcal{L}_{fm} + \lambda\mathcal{L}_{align} \tag{2}$$

Empirically, we found $\lambda = 0.2$ worked the best in our experiments.

B. Action-aware Future Embedding Model

While our future latent alignment framework is broadly compatible with various embedding models, we find that incorporating an *action-aware* future embedding yields further improvements in both performance and efficiency. To this end, we propose a compact vision-language embedding of the robot's current observation, explicitly optimized for policy learning. The design objective is twofold: achieving **compactness** while ensuring **action-awareness**.

Specifically, we leverage both the vision and text encoders from SigLIP-2 [37] to encode the robot's image observations and text instructions. The encoded tokens are then fused using four layers of self-attention transformer blocks to capture cross-modal dependencies. Subsequently, we apply a Q-former [20] module to compress the fused sequence into $M = 32$ learnable query tokens, producing a compact, fixed-size representation that naturally generalizes to multi-camera inputs. To ensure action-awareness, we train the vision language embedding end-to-end with the regular action flow-matching objective to predict the robot's actions by attaching 8 DiT blocks, following [28]. In this way, all task-relevant information is guaranteed to be captured within the latent token embeddings.

To pretrain the embedding model, we leverage a diverse mixture of cross-embodiment robot datasets, comprising both simulated and real-world humanoid tabletop manipulation data from GR00T N1 [28] and seven additional datasets from Open X-Embodiment [30], totaling approximately 2,000 hours of robotic data. We refer the readers to Appendix A for more details about the data mixture. Following pretraining, we posttrain the downstream policy jointly with the latent world model and the action prediction objective across downstream domains and tasks. Specifically, for posttraining, we initialize the downstream policy's encoder with the pretrained embedding model, while also using the pretrained embedding model to define the prediction targets for future latent representations. To mitigate distribution shifts between pretraining and downstream visual observations, rather than keeping the embedding model entirely frozen, we adopt an exponential moving average (EMA) update with respect to the policy's encoder. This strategy allows the embedding model to gradually adapt in tandem with the evolving vision and language encoders during policy fine-tuning. Empirically, we find that an EMA update rate of 0.995 performs the best.

IV. Experiments

A. Multitask Benchmark Performance

In this section, we evaluate our latent world model on two multitask imitation learning benchmarks. For single-arm manipulation benchmark, we adopt Robocasa [27], consisting of 24 atomic tasks in a simulated kitchen environment, including pick-and-place, door manipulation, faucet operation, and more. Robot's observations include three RGB images captured from cameras mounted on the left, right, and wrist of the robot. Next, we incorporate 24 GR-1 tabletop simulation tasks from [28], which emphasize dexterous hand control with the GR-1 humanoid robot. This suite includes 18 object rearrangement tasks—picking up and placing objects between source and target containers—and 6 tasks involving interaction with articulated objects such as cabinets, drawers, and microwaves. Observation consists of a single RGB image from an egocentric camera positioned on the robot's head.

To ensure a fair comparison between our method and the baseline, for experiments in this section, we do not use the pretrained embedding model mentioned in Section III-B. Instead, we pretrain the embedding model exclusively on the same in-domain multitask dataset for 80,000 gradient steps, ensuring that any performance gains cannot be attributed to pretraining data with the embedding model.

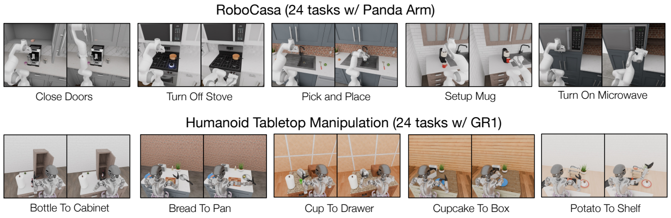


Fig. 3: Multitask Simulation Benchmarks: We use 24 Robocasa [27] and 24 GR-1 tabletop manipulation tasks as multitask simulation benchmark suite in this paper.

In particular, we include the following baselines for the experimental results:

1. Diffusion Policy [8]: Diffusion Policy models action distributions via a diffusion-based generative process, rather than using flow matching. It uses a U-Net architecture that progressively denoises random noise to generate the final action.
2. UWM [52]: We select UWM as the main baseline for methods that jointly learn video and action prediction objectives. UWM predicts image VAE latents and actions jointly with a diffusion objective.
3. GR00T N1 (Scratch) [28]: Since GR00T N1 is pretrained on a much broader data mixture, we ensure a fair comparison by using the same architecture but initializing the DiT layers from scratch, while only loading the pretrained Eagle VLM [23] model weights.
4. **FLARE** with Policy Only: We use the exact same model architecture as **FLARE**, as mentioned in Section III-B, but train it solely with the policy learning objective.

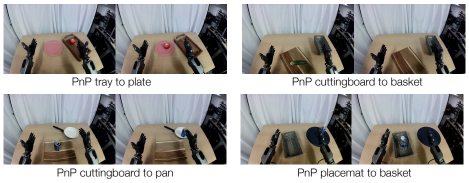


Fig. 4: Real GR1 Tasks Setup: We evaluate four tabletop manipulation tasks on a real GR1 humanoid robot.

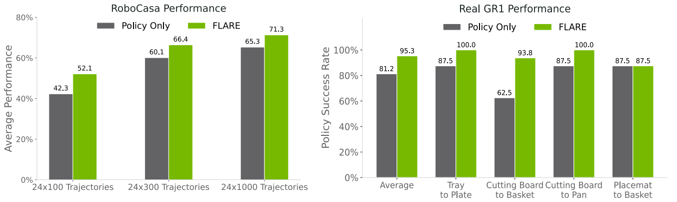


Fig. 5: **(Left)**: Post-training results on 24 Robocasa tasks. **(Right)**: Post-training results on 4 Real GR1 humanoid tasks.

All methods are trained for 80,000 gradient steps on the multitask robot dataset, except for UWM. We noticed that UWM performance is still improving at the end of 80k gradient steps, and thus we extend its training to 400k steps—five times the training budget allocated to the other methods. Following [28], we evaluate each model checkpoint for 50 episodes per task every 1000 gradient steps, and report the maximum success rate over the final five checkpoints for each method.

As shown in Table I, we draw two key observations. First, **FLARE** consistently outperforms all baseline methods including both the policy-only baselines and UWM. This highlights the strength of our compact, action-aware latent world modeling objective in enabling more effective policy learning. Additionally, in our experiments, we also observe that **FLARE** with the policy-only objective, trained for 160k gradient steps, achieves only 44.1% success rate, resulting in no performance difference compared with 80k gradient steps. Thus, the improved results cannot simply be attributed to more training steps with **FLARE**. Second, even when trained with only the policy objective, FLARE still achieves performance on par with GR00T N1 initialized from scratch, despite GR00T N1 using a larger VLM backbone. This result underscores the quality of our Q-former-based vision-language embedding model in capturing action-relevant information.

B. Data-efficient Post-training with Cross-embodiment Pretrained Embedding Model

While the latent world model demonstrates substantial performance gains, as shown in the previous section, it requires training a separate embedding model for each domain. In this section, we evaluate **FLARE** with the pretrained embedding model mentioned in Section III-B as the future prediction target, focusing on unseen embodiments and tasks with data limited posttraining settings. Specifically, we select 24 Robocasa arm tasks and 4 real world GR1 humanoid tabletop manipulation tasks as the evaluation benchmarks, and post-train the policy jointly with the latent world model and policy objectives, comparing it against a baseline that is post-trained using only the policy objective. In particular, for the policy-only

Methods	FLARE	Policy Only	UWM	GR00T N1 (Scratch)	Diffusion Policy
Pick and Place	**53.2%**	43.8%	35.6%	44.1%	29.2%
Open & Close Doors / Drawers	**88.8%**	78.7%	82.0%	80.0%	78.7%
Others	**80.0%**	75.2%	74.2%	69.6%	61.3%
24 RoboCasa Tasks Average	**70.1%**	61.9%	60.8%	60.6%	51.7%
Pick and Place Tasks	**58.2%**	46.6%	30.1%	51.8%	40.4%
Articulated Tasks	**51.3%**	47.4%	38.4%	42.8%	50.1%
24 GR1 Tasks Average	**55.0%**	44.0%	29.5%	45.1%	40.9%

TABLE I: Task Success Rate Breakdown for Multitask Policy on Robocasa and GR1 Tabletop Manipulation

baseline, we initialize both the Q-former-based vision language embedding and the policy's DiT model weights from the cross-embodiment pretrained model. For **FLARE**, we only warm start the vision language embedding model.

For the evaluation protocol, we follow the same procedure described in IV-A for the 24 RoboCasa tasks. For the 4 real-world GR-1 tasks shown in Figure 4, we define 8 reference initial frames per task, each involving 4 distinct objects (apple, can, bottled water, cucumber) to manipulate, and report the success rate of the final policy checkpoint for each method.

As shown in Figure 5, across both the 24 Robocasa simulation tasks and the real-world GR-1 humanoid tasks, **FLARE** consistently outperforms the policy-only baseline. The improvement is especially pronounced under limited data conditions, achieving a 10% gain on Robocasa with 100 trajectories per task for posttraining. Notably, although the pretrained embedding model has never seen Robocasa tasks during pretraining, using it as the future embedding achieves comparable performance with 1000 trajectories to an embedding model trained exclusively on the 24 Robocasa arm tasks (71.3% vs. 70.2% as reported in Section IV-A).

On the real GR-1 humanoid robot, we achieve a success rate of up to **95.1%**, averaging 14% higher than the baseline method. Qualitatively, we observe that in scenarios where a can or water bottle is placed close to the robot's hand, the baseline method trained with only the policy objective often knocks over the object. In contrast, **FLARE** policy learns to maneuver around or over the object and successfully grasping, highlighting the benefits of future latent reasoning enabled by **FLARE**.

C. Leveraging Human Egocentric Trajectories without Action Labels

While our previous experiments demonstrate that the proposed future latent alignment objective significantly enhances policy performance when trained on action-labeled data, we further show that it can be naturally extended to trajectories without action annotations, such as human egocentric demonstrations. To evaluate this, we select five novel objects with distinctive geometries that are absent from our GR1 in-house pretraining dataset, each requiring novel grasping strategies. For instance, the blue tape object is large and thus requires a top-down grasp by the robot hand. For each object, we collect 150 human egocentric demonstrations per object, by mounting a GoPro on the demonstrator's head while they perform the similar tasks as the humanoid robot. On the robot side, we collect only 10 teleoperated demonstrations per object and train the policy using a mixture of these limited demonstrations, our GR-1 pretraining dataset, and the egocentric human videos.

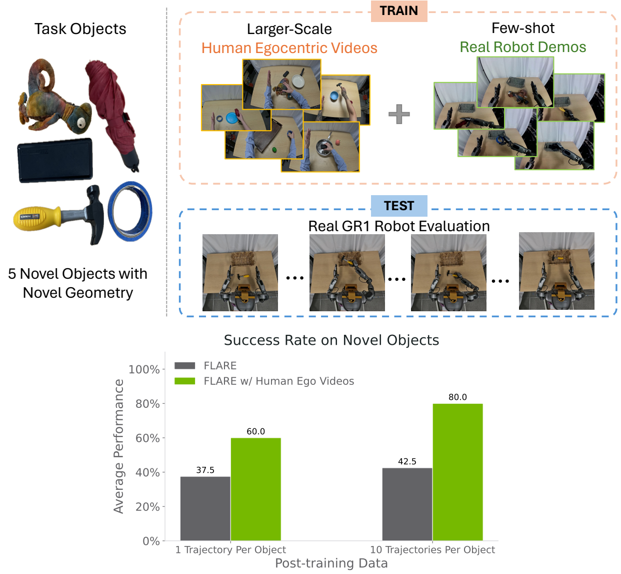


Fig. 6: Generalizing to unseen objects with human egocentric videos and few-shot real robot demos

For real-robot demonstrations with actions, we apply both the action flow-matching loss and the future alignment objective. In contrast, for the human egocentric videos without action labels, we rely solely on the future alignment loss to learn the latent dynamics. At evaluation time, we select five initial poses as reference images for each object and measure the robot's success rate. Partial credit (0.5) is given when the robot successfully grasps the object but fails to place it into the basket.

As shown in Figure 6, with only **1** teleoperated trajectory per object, **FLARE** already achieves up to a 60% success rate on novel objects. When provided with 10 trajectories per object, and jointly trained with human videos, **FLARE** further improves to an 80% success rate—roughly doubling the performance of a baseline trained solely on action-labeled data. These results highlight that **FLARE** not only enhances learning from action-labeled demonstrations, but also effectively leverages unlabeled human demonstrations to improve generalization by capturing latent task dynamics.

V. Conclusion

We present Future Latent Representation Alignment (**FLARE**), a simple yet effective framework for jointly learning robot policy and latent world dynamics. By aligning the future representations of the robot's observations with the hidden states of the action denoising network, **FLARE** enables the policy to implicitly reason about future states while predicting actions. This approach leads to state-of-the-art performance on challenging robotic manipulation benchmarks. Furthermore, **FLARE** unlocks co-training with human egocentric video demonstrations that lack action labels, significantly improving generalization to novel objects with minimal real-robot teleoperation data.

Appendix

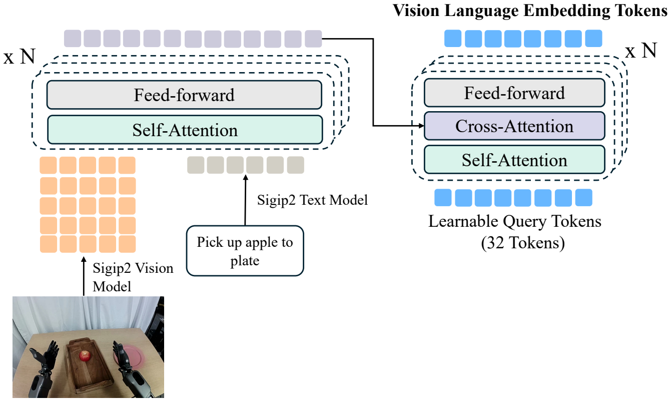


Fig. 7: Our Q-former based Vision Language Embedding Module

We present the architectural details of our compact Q-former-based vision-language embedding module. Specifically, we adopt `siglip2-large-patch16-256` as the backbone for both vision and language encoders. The SigLIP2 vision encoder processes 256×256 resolution robot images into 256 patch tokens, while the language encoder encodes padded robot instructions into 32 language tokens. hese 256 vision tokens and 32 language tokens are concatenated and passed through four layers of self-attention transformers to yield 288 fused vision-language tokens. To obtain a compact representation, we apply a Q-former architecture [20], where 32 learnable query tokens—randomly initialized—interact with the 288 fused tokens through interleaved self-attention and cross-attention layers, producing 32 compressed vision-language tokens.

A. Pretraining Data Mixture

Details of pretraining data mixture weight and statistics is presented in Figure 8, Table II.

B. Training Details

For the pretraining of action-aware vision language embedding module, we use 256 NVIDIA H100 GPUs with batch size 8192 for 150,000 gradient steps. We use AdamW [25] optimizer with $\beta_1 = 0.95, \beta_2 = 0.999$, and $\epsilon = $ 1e-8. A weight decay of 1e-5 is applied, and the learning rate follows a cosine scheduling strategy with a warmup ratio of 0.05. Following [2, 28], we sample the flowmatching denoising timestep from $p(\tau) = \text{Beta}(\frac{s-\tau}{s}; 1.5, 1)$, $s = 0.999$.

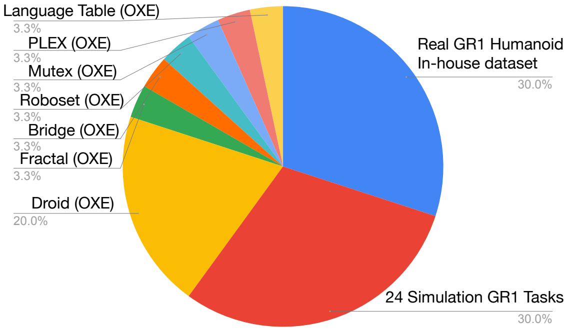


Fig. 8: Data mixture of pretrained action-aware vision language embedding model

TABLE II: Action-Aware Vision Language Embedding Pre-training Dataset Statistics

Dataset	Length (Frames)	Duration (hr)	FPS	Camera View	Category
GR-1 In-house Dataset	6.4M	88.4	20	Egocentric	Real robot
DROID (OXE) [17]	23.1M	428.3	15	Left, Right, Wrist	Real robot
RT-1 (OXE) [3]	3.7M	338.4	3	Egocentric	Real robot
Language Table (OXE) [26]	7.0M	195.7	10	Front-facing	Real robot
Bridge-v2 (OXE) [38]	2.0M	111.1	5	Shoulder, left, right, wrist	Real robot
MUTEX (OXE) [35]	362K	5.0	20	Wrist	Real robot
Plex (OXE) [36]	77K	1.1	20	Wrist	Real robot
RoboSet (OXE) [1]	1.4M	78.9	5	Left, Right, Wrist	Real robot
GR-1 Simulation	125.5M	1,742.6	20	Egocentric	Simulation
Total	169.5M	2,989.5	–	–	–

For the multitask experiments of **FLARE** conducted in Section IV-A and IV-B, we use 32 NVIDIA H100 GPUS with batch size 1024 for 80,000 gradient steps, while keeping the rest of the hyperparameter setups exactly the same.

C. Using the Pretrained Siglip2 as Future Embedding model

Method	Success Rate (%)
No FLARE loss	43.9
SigLIP2	49.6
SigLIP2 (Average Pooled)	50.9
Action-aware Embedding	**55.0**

TABLE III: Ablation of target embedding models.

While leveraging a policy-oriented future embedding model results in strong policy performance and enhanced training efficiency, we also explore an alternative setting that employs pretrained SigLIP2-Large vision tokens at timestep $t + 16$ as prediction targets. Specifically, we experiment with using both raw SigLIP2 vision tokens (256 tokens per image) and 2x2 average-pooled tokens (64 tokens per image). As illustrated in Table III, our **FLARE** framework maintains compatibility with diverse teacher encoder models beyond the policy-oriented embedding model. Although we get the optimal performance with embedding model pretrained specifically on the target domain, using a more general-purpose vision encoder such as SigLIP2 still yields a significant 7% improvement over baseline methods. A key design decision in **FLARE** is selecting the DiT layer at which to apply the future latent alignment loss, and the coefficient λ of **FLARE** loss. In our main experiments, we apply this objective at layer 6 out of 8 total layers in the DiT architecture. Applying it at deeper layers allows a larger portion of the model weights to benefit from the supervision of future latent prediction, but may also lead to conflicts between the action prediction and future alignment objectives. To evaluate the effect of these two hyperparameters, we evaluate **FLARE** on the GR1 simulation benchmark with different layer indexes and coefficients used for alignment. As shown in Figure 9, the model maintains strong performance across a range of hyperparameter setups. However, we do notice that applying the alignment objective too early—e.g., at layer 4—leads to a notable drop in performance, highlighting the importance of aligning the future prediction objective with the action denoising process.

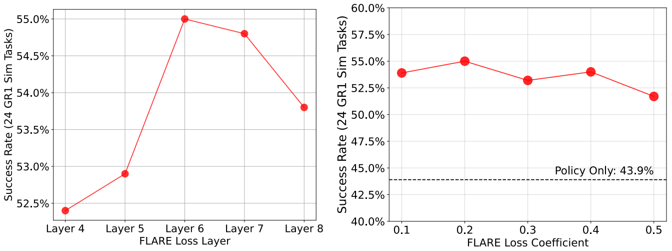


Fig. 9: (**Left**): Ablation of the DiT Layer used in FLARE loss (**Left**): Ablation of FLARE loss coefficient.

D. Exponential Moving Average (EMA) of Pretrained Action-aware Embedding Model

As discussed in Section III-B, to address the distribution shift between pretraining and downstream tasks for our action-aware vision-language target embedding model, we incorporate an exponential moving average (EMA) update. Specifically, at each gradient step, the target embedding model parameters are updated as follows:

$$\theta_{\text{target\_vl\_embedding}} \leftarrow \rho\theta_{\text{target\_embedding}} + (1-\rho)\theta_{\text{policy\_vl\_embedding}}$$

While the policy's vision-language encoder is initialized from the target vision-language encoder, i.e. pretrained action-aware vision-language embedding, it gradually adapts to the downstream task during training via the action flow-matching objective. The EMA update enables the prediction target to adapt slowly in tandem with the evolving policy encoder, providing stability across training.

We evaluate several choices of the EMA coefficient $\rho \in \{0.99, 0.995, 0.999, 1.0\}$, each using 24×300 trajectories to train the **FLARE** policy. The final average success rates are reported in Figure 10. We find that while all EMA variants outperform the baseline method without **FLARE** future latent alignment objective, $\rho = 0.995$ yields the best performance and is used in all experiments. Notably, even with $\rho = 1.0$ (i.e., no EMA), **FLARE** still surpasses the baseline, whereas $\rho = 0.99$ performs the worst, likely due to the instability caused by frequent target updates.

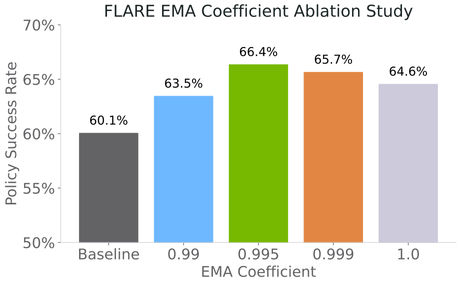


Fig. 10: Effect of EMA Coefficient ρ: We report the policy success rate using 24×300 training trajectories across 24 RoboCasa tasks. Baseline is trained without **FLARE** future alignment loss, i.e. a policy only objective.

E. Related Work

Generative World Models for Robotics. There has been a rich body of research on world models for robotics, ranging from model-based control to model-based reinforcement learning [16, 12, 13, 7, 39, 48]. More recently, with advances in image and video generation, several works have explored the integration of generative modeling into policy learning [41, 5, 9, 52, 21, 10]. One line of work [9, 15] uses image diffusion models with inverse dynamics models to close the perception-to-action loop. The GR1 and GR2 families introduce end-to-end models that jointly predict discrete image tokens and actions using a unified next-token prediction objective. Other approaches [52, 21, 51, 34, 33, 47, 49] instead aim to jointly predict continuous image latents and actions. For instance, UWM [52] and UVA [21] jointly denoise VAE latents of future frames along with robot actions. DINO-WM [51] utilizes DINO features [51] to train a latent dynamics model for model-based planning.

Our work builds upon recent advances in representation learning, particularly Representation Alignment [44], which has shown remarkable success in accelerating the convergence of diffusion transformers for image generation and is key to state-of-the-art flow-matching models like Seedream-3.0 [11]. However, our approach differs in two crucial ways: we train a flow-matching *policy* rather than an image model, and we align the DiT representation with features from *future* observations rather than current ones. In contrast to existing works, **FLARE** introduces an implicit latent world model objective that bypasses explicit reconstruction of future frames or latents. This simple design enables reasoning over a compact, action-aware latent space and avoids the computational burden of high-fidelity generation, while maintaining compatibility with standard VLA architectures—without requiring major architectural redesign. While DINO-WM focuses on zero-shot planning, **FLARE** is designed for policy and world model co-training, though planning could be a valuable future extension.

Vision Language Action Models. A growing body of recent work [3, 4, 2, 18, 50, 40, 6, 22, 46, 14, 43, 42] has focused on developing general-purpose foundational vision-language-action (VLA) models by fine-tuning vision-language models for downstream robotics tasks. Among these works, models such as [18, 42, 32, 19] autoregressively predict sequences of discrete action tokens using next-token prediction objective. In contrast, methods like [29, 2, 28] leverage diffusion-based or flow-matching policy heads to bridge pretrained VLMs with continuous action generation. In this work, inspired by the architecture of GR00T-N1 [28], we adopt a flow-matching policy head built with diffusion transformer blocks, using interleaved self-attention and cross-attention layers to condition on the fused vision-language embeddings.

F. Pseudocode of **FLARE**

Here we present a Python-style pseudocode of **FLARE** loss calculation as well as the entire training loop.

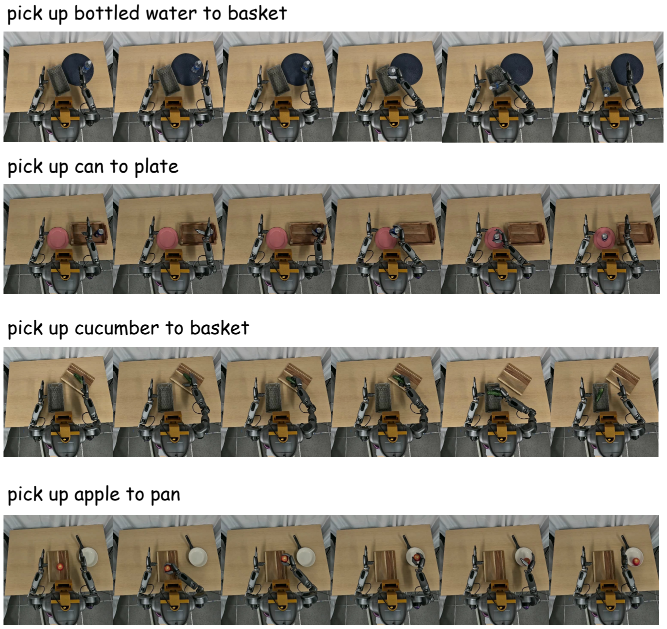
Algorithm 1 Python-style pseudocode for FLARE training

```
# target_vl_embedding: pretrained action-aware vision
    language embedding
# vl_embedding:  vision language embedding of the current
    policy
# dit: diffusion transformer of the current policy
# action_embedding: 2-layer MLP to embed noisy actions
# state_embedding:  2-layer MLP to embed prioprioceptive
    state
# action_decode:    2-layer MLP to decode robot's actions
# embedding_decode: 2-layer MLP to decode predicted
    embeddings
# N: Number of gradient steps
# M: Number of tokens in VL
# lambda: coefficient of FLARE loss (default is 0.2)

### Initialization
future_tokens = nn.Embedding(M, hiddem_dim)
vl_embedding.load_state_dict(vl_embedding.state_dict())
target_vl_embedding.requires_grad = False

for n in range(N):
    obs, proprio, actions, future_obs = dataset.next()

    ### Prepare noisy action inputs
    noise = gaussian.sample()
    timestep = beta.sample() # sample flowmatching timestep
    noisy_action = timestep * actions + (1-timestep) * noise
    velocity = actions - noise

    ### Get state, action, and observation embedding tokens
    action_tokens = action_embed(noisy_action, timestep)
    state_token   = state_embed(state)
    vl_tokens     = vl_embedding(obs)

    ### Pass through DiT layers
    sa_tokens   = torch.concat([state_token, action_tokens,
     future_tokens], dim=1)
    policy_outputs = dit(sa_tokens, vl_tokens)

    ### Calculate action flowmatching loss
    action_outputs = action_decoder(policy_outputs[:, 1:1 +
     action_tokens.shape[1]])
    action_loss = MSE(action_outputs, velocity)

    ### Calculate FLARE loss
    with torch.no_grad():
        embedding_to_align = target_vl_embedding(future_obs)
    predict_embedding = decode_embedding(policy_outputs[:,
     -M:])
    flare_loss = 1-COSINE_SIMILARITY(predict_embedding,
     embedding_to_align)

    ### Optimize the combined loss
    loss = action_loss + lambda * flare_loss
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

G. Real GR1 Humanoid Rollouts

H. 4 Pick-and-place Tasks

Below, we present policy rollouts from the **FLARE** trained policy on 4 real-world GR1 humanoid pick-and-place tasks, together with the task's language instructions. Qualitatively, we observe that when manipulating objects such as bottled water or coke can, the **FLARE** policy learns to maneuver the hand around the object, climbing overt the water bottle, rather than striking and knocking it over.


pick up bottled water to basket

pick up can to plate

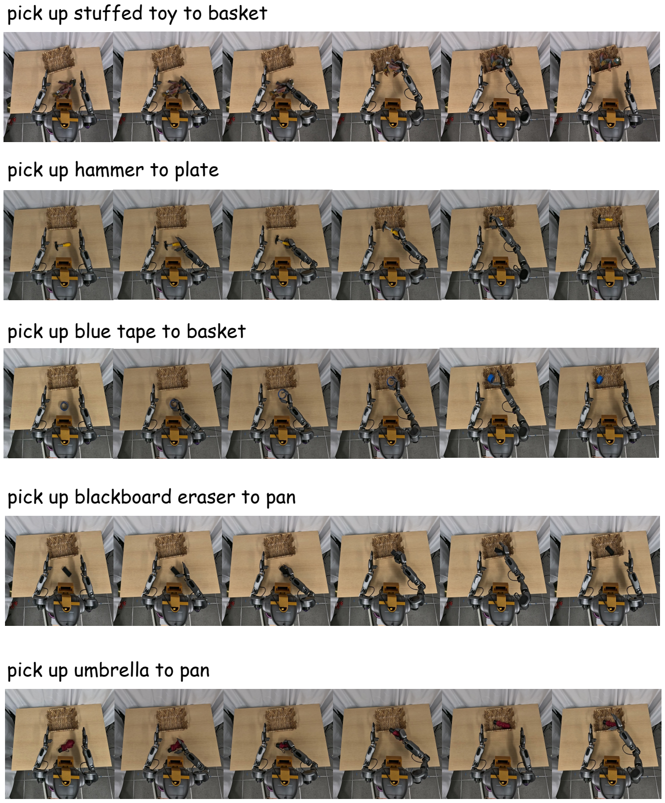
pick up cucumber to basket

pick up apple to pan

Fig. 11: FLARE policy rollout on real GR1 humanoid robot with 4 pick-and-place tasks

I. Manipulating Novel Objects

Below, we present policy rollouts from the **FLARE** trained policy manipulating 5 novel objects.


pick up stuffed toy to basket

pick up hammer to plate

pick up blue tape to basket

pick up blackboard eraser to pan

pick up umbrella to pan

Fig. 12: FLARE policy rollout manipulating 5 novel objects

References

[1] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[5] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[6] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[7] Jin Cheng, Dongho Kang, Gabriele Fadini, Guanya Shi, and Stelian Coros. Rambo: Rl-augmented model-based optimal control for whole-body loco-manipulation, 2025. URL https://arxiv.org/abs/2504.06662.

[8] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[9] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bo8q5MRcwy.

[10] 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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9pKtcJcMP3.

[11] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. Seedream 3.0 technical report. *arXiv preprint arXiv:2504.11346*, 2025.

[12] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[13] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. 2022.

[14] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[15] Shuaiyi Huang, Mara Levy, Zhenyu Jiang, Anima Anandkumar, Yuke Zhu, Linxi Fan, De-An Huang, and Abhinav Shrivastava. Ardup: Active region video diffusion for universal policies, 2025. URL https://arxiv.org/abs/2406.13301.

[16] Xiaoyu Jiang, Qiuxuan Chen, Shiyi Han, Mingxuan Li, Jingyan Dong, and Ruochen Zhang. When to trust your model: Model-based policy optimization, 2020. URL https://openreview.net/forum?id=SkgPIpcGar. Submitted to NeurIPS 2019 Reproducibility Challenge.

[17] 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, Kevin Lin, 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. 2024.

[18] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[19] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[20] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/li23q.html.

[21] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[22] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[23] Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, et al. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025.

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[26] 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, 2022.

[27] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

[28] NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

[29] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[30] Open X-Embodiment Collaboration et al. Open X-Embodiment: Robotic learning datasets and RT-X models. International Conference on Robotics and Automation, 2024.

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[32] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

[33] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?

id=uCQfPZwRaUu.

[34] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R Devon Hjelm, Philip Bachman, and Aaron C Courville. Pretraining representations for data-efficient reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12686–12699. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/69eba34671b3ef1ef38ee85caae6b2a1-Paper.pdf.

[35] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023.

[36] Garrett Thomas, Ching-An Cheng, Ricky Loynd, Felipe Vieira Frujeri, Vibhav Vineet, Mihai Jalobeanu, and Andrey Kolobov. Plex: Making the most of the available data for robotic manipulation pretraining. In *CoRL*, 2023.

[37] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[38] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[39] Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. COPlanner: Plan to roll out conservatively but to explore optimistically for model-based RL. In *NeurIPS 2023 Workshop on Generalization in Planning*, 2023. URL https://openreview.net/forum?id=9lkkqGagDF.

[40] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024.

[41] 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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NxoFmGgWC9.

[42] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, Lars Liden, and Jianfeng Gao. Magma: A foundation model for multimodal ai agents, 2025. URL https://arxiv.org/abs/2502.13130.

[43] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VYOe2eBQeh.

[44] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

[45] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models, 2025. URL https://arxiv.org/abs/2503.22020.

[46] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

[47] Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. Taco: Temporal latent action-driven contrastive loss for visual reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 48203–48225. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/96d00450ed65531ffe2996daed487536-Paper-Conference.pdf.

[48] Ruijie Zheng, Xiyao Wang, Huazhe Xu, and Furong Huang. Is model ensemble necessary? model-based RL via a single model with lipschitz regularized value function. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hNyJBk3CwR.

[49] Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Shuang Ma, Hal Daumé III, Huazhe Xu, John Langford, Praveen Palanisamy, Kalyan Shankar Basu, and Furong Huang. Premier-taco is a few-shot policy learner: pretraining multitask representation via temporal action-driven contrastive loss. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[50] 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 *The Thirteenth International Conference on Learning Representations*, 2025.

[51] 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.

[52] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. 2025.