ByteDance | Seed

World Value Models for Robotic Manipulation

Zhihao Wang[1,2,3], **Jianxiong Li**[1,3,†], **Yu Cui**[1,§], **Yuan Gao**[3], **Xianyuan Zhan**[3], **Junzhi Yu**[2,§], **Xiao Ma**[1]

[1]ByteDance Seed, [2]Peking University, [3]Tsinghua University

[†]Project Lead, [§]Corresponding Author

Abstract

Generalist value models play a pivotal role in scaling robotic policy learning from large-scale, mixed-quality data. Mathematically, accurate value estimation demands deep temporal understanding, requiring models to both ground the current belief using historical context and plan over future outcomes. However, most existing robotic value models are built on Vision-Language Model (VLM) backbones that are pretrained primarily on static or temporally sparse visual observations, lacking the requisite temporal modeling capabilities for value estimation. Unlike VLMs, world models naturally excel at temporal modeling and future planning, making them ideal foundations for learning generalizable value functions. Driven by this insight, we marry world models with value estimation to construct a new generalist robotic value model, **World Value Model** (**WVM**), that offers accurate task progressions to assess data quality. On standard benchmarks, WVM delivers state-of-the-art (SOTA) Value-Order Correlation (VOC) results. Complementing standard evaluation suites that contains only expert data, we further introduce **Suboptimal-Value-Bench**, a multi-embodiment benchmark consisting of 800 suboptimal trajectories with high-fidelity, human-labeled frame annotations. Our evaluations show that WVM maintains its SOTA performance on Suboptimal-Value-Bench, establishing its robustness in handling both expert and suboptimal data. When deployed for policy learning, WVM improves manipulation performance across various policy extraction approaches in both simulated and real-world deployment, providing robust guidance for learning from mixed-quality data.

Date: June 18, 2026
Correspondence: cuiyu.0627@bytedance.com, yujunzhi@pku.edu.cn
Project Page: zh1hao.wang/wvm

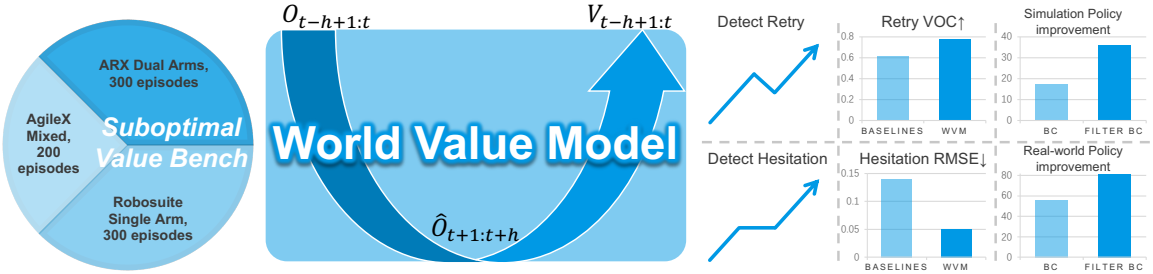


Figure 1: Overview of World Value Models and Suboptimal-Value-Bench. WVM leverages the world models to estimate temporally grounded task progress from videos, enabling the reliable detection of suboptimal behaviors such as hesitation and retry. Evaluated on Suboptimal-Value-Bench, our new 800-trajectory multi-embodiment benchmark, WVM significantly enhances both high-fidelity value estimation and downstream policy learning.

1 Introduction

Generalist value models serve as a cornerstone for scaling robotic policy learning from vast and heterogeneous datasets [7, 15, 28, 75, 76], providing learning signals for large scale real-world RL systems [30, 49, 69] as well as offline data filters [19, 21]. At its core, accurate value estimation requires a dual capability: a thorough comprehension of past temporal contexts [14, 22, 23, 67], combined with prospective forward-looking planning over long-term future outcomes [13, 46, 57, 59]. However, merging these temporal dimensions into a single value estimator remains a formidable challenge in practice [2, 4, 18, 50, 55].

Existing value models, despite their notable promise, struggle to deliver high-fidelity progress estimations because they are hindered by three primary bottlenecks: (1) inefficiencies in value learning due to a heavy dependence on scalar value supervision [61, 73], (2) limited generalization resulting from narrow, task-specific customization [19, 30, 39, 69], and (3) impaired temporal understanding and future planning capabilities, a direct consequence of the sparse visual modeling characteristic of underlying VLMs [6, 26, 43, 61, 73].

World models have achieved notable success in modeling temporal dynamics and forecasting future states across both video generation [38, 58, 64] and robotic manipulation [29, 48, 71], demonstrating strong capabilities in spatial and temporal understanding. Consequently, they inherently possess the dual properties required by a generalist value estimator. Driven by this, our key insight is that the spatiotemporal priors embedded in world models can be repurposed as a powerful foundation for value learning. We introduce WVM, a World Value Model engineered to inherit rich spatial-temporal priors from a pretrained video world model. Specifically, WVM couples the video stream with a lightweight value Diffusion-Transformer (DiT) [52] via a Mixture-of-Transformers (MoT) [33] architecture. This design allows value tokens to selectively attend to structural video latents while minimizing representation interference with the video generation stream during training. To achieve comprehensive value learning over large-scale data corpus, WVM formulates the value function as a distributional value chunk trained by flow matching [34]. This formulation provides dense training signals and enhanced expressiveness, thereby yielding superior value estimation performance compared to traditional scalar supervision and conventional categorical distributions [3, 10]. Finally, a suite of augmentations, including video rewinding [74] and value prefix randomization, are applied during training, empowering WVM to robust prediction both optimal and suboptimal task progress during inference.

When evaluating generalist value models, current practices mainly rely either on qualitative human visual comparisons or on non-scalable downstream policy performance. Another common choice is VOC [43], but it can only reflect value models' task progress awareness on optimal trajectories. To address these limitations, we introduce Suboptimal-Value-Bench, a multi-embodiment benchmark comprising 800 trajectories paired with human-annotated ground-truth task progress. Featuring two prevalent suboptimal behavioral modes—retries and hesitations—Suboptimal-Value-Bench enables a comprehensive evaluation of generalist value models that extends far beyond the scope of existing metrics.

Experiments show that WVM can generate values with higher qualities than value-model baselines on both Suboptimal-Value-Bench and expert VOC. Also the ablations validate the necessity of our world-model prior and core architectural choices. Beyond standalone value quality evaluation, integrating WVM into downstream policy learning yields substantial performance gains with only noisy-data in both simulation and real-world manipulation tasks. We will make WVM and Suboptimal-Value-Bench available to support the community.

Our main contributions are threefold:

(1) We repurpose world models as foundational backbones for robotic value learning, leveraging their rich spatiotemporal priors to overcome the limitations of standard VLMs.

(2) WVM is the first large-scale value flow model that formulates value functions as distributional chunks. Complemented by simple-yet-effective design choices, WVM achieves SOTA performance across diverse benchmarks and proven to be effective for policy improvement.

(3) We introduce Suboptimal-Value-Bench, a new evaluation suite featuring dense, human-labeled suboptimal trajectories tailored for value model evaluations.

2 Related Work

Value models for robotic manipulation. Existing robotic value models face three persistent bottlenecks as generalist progress estimators. First, scalar value regression on high-dimensional observations provides a sparse, low-information supervision signal, yielding sample-inefficient training and brittle predictions when scaled to heterogeneous video corpora [61, 73]. Second, many value models [19, 30, 69] are tightly tailored to a single task and thus cannot serve as a generalist progress estimator. Third, generalist value estimators built on pretrained VLM backbones [6, 26, 31, 43, 61, 73] inherit a representation optimized for static or temporally sparse images, and thus cannot capture dense temporal dynamics. While ViVa [39] represents the closest attempt to ours by building upon video models, it is confined to single-task settings and remains fundamentally restricted to action-annotated data. To address these limitations, WVM reformulates value estimation as a distributional chunk, enabling scalable, multi-task learning across massive, action-free video datasets by leveraging expressive visual-physical priors from a pretrained video world model.

World models for robotic manipulation. World models [1, 8, 12] have recently gained prominent traction in robotic manipulation through the emergence of World Action Models (WAMs) [65, 71], which jointly model action-conditioned visual dynamics. Prior works [25, 29, 40, 48, 62, 70, 71] demonstrate that such video priors significantly boost learning efficiency by leveraging the world model's innate capacities for both temporal reasoning and forward prediction. Building upon this, Fast-WAM [72] illustrates that operating future predictions entirely within the latent space retains these representational benefits without the need for explicit pixel-level decoding [36, 44]. Shifting away from these predominantly policy-centric deployments, WVM repurposes latent video priors for value estimation. Through a MoT design, our framework preserves the backbone's intrinsic video-modeling capacity while successfully deriving robust value predictions from its latent temporal features.

Evaluation of robotic value models. Quantifying the performance of robotic value models remains a critical challenge [31, 43]. Early works [27, 41, 42, 45] rely on qualitative curve inspection, which does not scale to systematic comparison. Another line of work [61, 73] measures value quality indirectly through downstream policy success, entangling value fidelity with policy choice and incurring substantial computational overhead. GVL [43] proposes Value-Order Correlation (VOC), but its monotonicity criterion only applies to expert trajectories, and thus lacks assessment for suboptimal segments. Complementing all three, our Suboptimal-Value-Bench evaluates value models on human-annotated hesitation and retry trajectories, directly reflecting their ability to flag suboptimal segments.

3 Method

3.1 Problem Formulation

We formulate value estimation as a chunk-wise prediction problem. Given an h-frame observation sequence $o_{t-h+1:t}$ and a language instruction l, the value model defines a conditional distribution over a length-h sequence of per-frame values:

$$p_\psi(\hat{v}_{t-h+1:t} \mid o_{t-h+1:t}, l), \tag{1}$$

where $\hat{v}_{t-h+1:t} \in [0,1]^h$. Here, $v_t = t/T$ denotes the normalized task progress, where T is the total trajectory length. Modeling the full length-h chunk instead of an isolated scalar enables the value model to capture local progress profiles—such as plateaus and regressions—that are pivotal for tracking temporal dynamics. Classically, a value function in reinforcement learning (RL) is defined as the expected discounted sum of future rewards [56, 60]:

$$V(o_t) = \mathbb{E}\left[\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \,\middle|\, o_t\right], \tag{2}$$

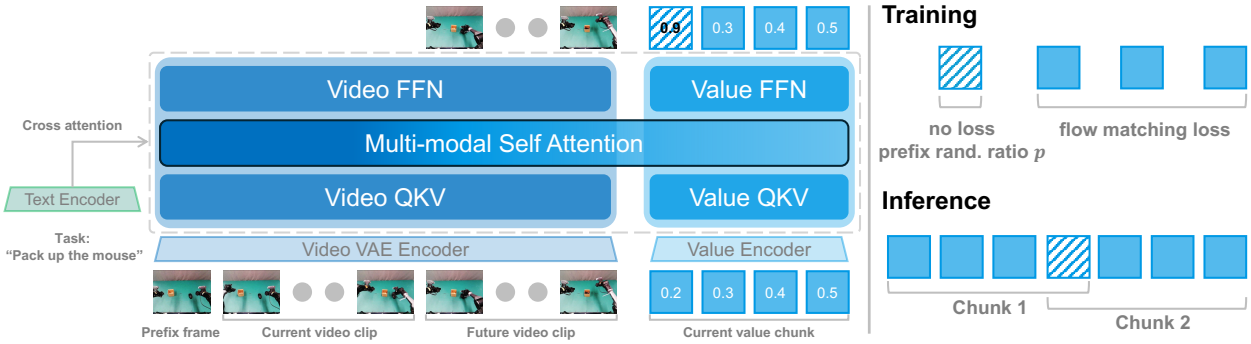


Figure 2 WVM's architecture, prefix randomization and chunk overlapping scheme.

where $r_{t'}$ is the step-level reward and $\gamma \in (0, 1]$ is the discount factor. Under the canonical sparse-reward setting where $r_{t'} = -1$ for non-terminal steps and 0 at task completion, $V(o_t)$ reduces to the negative expected distance-to-goal, rendering value estimation equivalent to task-progress prediction. By construction, the value function thereby intrinsically focuses on future outcomes [11]. This perspective naturally motivates utilizing a video world model M_ω as a rich feature extractor for value estimation:

$$p_\psi(\hat{v}_{t-h+1:t} \mid o_{t-h+1:t},\, l) = p_\psi(\hat{v}_{t-h+1:t} \mid M_\omega(o_{t-h+1:t},\, l)). \tag{3}$$

3.2 WVM Architecture

As illustrated in Fig. 2, WVM instantiates Eq. 3 via a video DiT, a value DiT, and Multi-model self attention to enable the value stream capitalize temporal video features.

Video stream. WVM uses the video VAE and video DiT of Wan2.2 [64] as the world-modeling stream. For a value chunk anchored at the time window $[t-h+1, t]$, we first feed the video VAE a clean video clip with length $(2h+1)$ consisting of one prefix frame, the current observation frames, and the target future frames:

$$\underbrace{o_{t-h}}_{\text{1-frame prefix}} \;\|\; \underbrace{o_{t-h+1:t}}_{h \text{ current frames}} \;\|\; \underbrace{o_{t+1:t+h}}_{h \text{ future frames}} . \tag{4}$$

The VAE encodes the clip into three temporal latents: we discard the prefix latent, keep the current latent as context, and corrupt the future latent for video-generation denoising.

Value stream and MoT coupling. The value stream is a lightweight DiT that mirrors the architecture of the video DiT with substantially fewer parameters. It predicts the value chunk $\hat{v}_{t-h+1:t}$ from noisy value tokens while attending to intermediate video-DiT features through MoT multi-modal self-attention. We adopt an asymmetric attention mask: value tokens attend to current video tokens but video tokens never attend to value tokens [72].

3.3 Training

Training objective. We apply flow matching [34, 35, 37] to the supervised video and value tokens. Let y denote either the future video latents $\xi_{t+1:t+h}$ or the value chunk $v_{t-h+1:t}$, and let f_ψ denote the corresponding velocity predictor. We sample noise $\epsilon \sim \mathcal{N}(0, I)$ and a time step $\tau \in (0, 1)$, construct the interpolated sample $y_\tau = \tau\, y + (1-\tau)\, \epsilon$, and train f_ψ to predict the velocity field $y - \epsilon$:

$$\mathcal{L}_{\mathrm{FM}}(y) \;=\; \mathbb{E}_{y,\,\epsilon,\,\tau}\Big[\, \big\| f_\psi(y_\tau,\, \tau,\, o_{t-h+1:t},\, l) - (y - \epsilon) \big\|_2^2 \Big]. \tag{5}$$

Instantiating Eq. 5 on the value chunk $v_{t-h+1:t}$ and the future video latents $\xi_{t+1:t+h}$ gives

$$\mathcal{L}_{\mathrm{value}} = \mathcal{L}_{\mathrm{FM}}(v_{t-h+1:t}), \qquad \mathcal{L}_{\mathrm{video}} = \mathcal{L}_{\mathrm{FM}}(\xi_{t+1:t+h}). \tag{6}$$

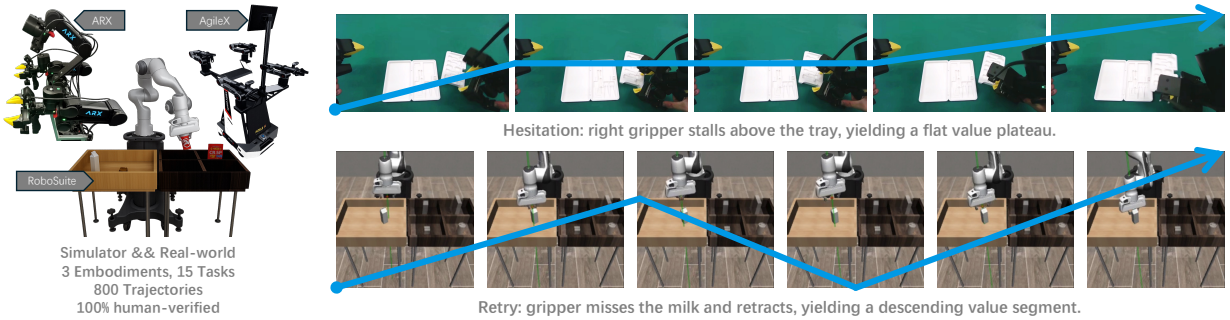


Figure 3 Setup of Suboptimal-Value-Bench, and blue arrows (⟶) schematically illustrate the per-frame human-annotated value curve under suboptimal modes (hesitation and retry).

The overall objective is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{value}} + \lambda\, \mathcal{L}_{\text{video}}, \tag{7}$$

where λ controls the weight of video co-training; see Appendix A for details.

Prefix randomization. Inference with value chunk overlapping improves chunk continuity but can introduce a shortcut, allowing the value stream to extrapolate from this prefix without visual evidence. To prevent this behavior, we apply prefix randomization during training, analogous to the conditioning dropout in classifier-free guidance (CFG) [16, 32]: with probability p, the prefix value is replaced by a random scalar sampled uniformly from $[0, 1]$; otherwise, the prefix is retained. The loss is applied only to the remaining value tokens [5]. Mixing clean and randomized prefixes preserves inter-chunk continuity while preventing the value stream from taking the prefix as a shortcut. We ablate p in Section 5.3.

Video rewinding. Expert trajectories provide solely monotonic progress labels, offering limited supervision for plateaus or regressions. Following ReWiND [74], we apply chunk-level rewind augmentation: for each window $o_{t-h+1:t}$, we sample one of three temporal patterns over the h frames—rising, plateau or descending—by preserving, repeating or reversing the frames, with $v_{t-h+1:t}$ relabeled accordingly. This exposes the value stream to local progress profiles associated with smooth advancement, hesitation and retry.

4 Suboptimal-Value-Bench

Real robot datasets commonly exhibit suboptimal segments (e.g., hesitation and retry) that are pivotal for practical value estimation. We present Suboptimal-Value-Bench, a benchmark consisting of 800 human-annotated trajectories across three embodiments and 15 tasks, with each frame augmented by a dense value curve focusing on hesitation and retry (Figure 3). We refer the reader to Appendix B for comprehensive dataset details.

4.1 Hesitation Segments

During a hesitation segment, the robot either remains stationary or executes task-irrelevant micro-movements rather than advancing the task progress. This behavior typically arises from teleoperator cognitive pauses (e.g., searching for a visual target) or physical hardware constraints (e.g., deceleration near joint limits). Therefore the task progress remains invariant throughout the segment. Because the standard VOC metric is ill-defined for invariant target trajectories [51], we evaluate hesitation segments using the Root Mean Squared Error:

$$\text{Hesitation-RMSE} \;=\; \sqrt{\frac{1}{|\mathcal{H}|}\sum_{t\in\mathcal{H}}(\hat{v}_t - v_t)^2}, \tag{8}$$

where $\mathcal{H}$ denotes the set of frames within a hesitation segment, and v_t represents the constant ground-truth value over that interval. This metric explicitly penalizes prediction drift; a model that maintains a constant,

Hesitation RMSE ↓	GVL	VLAC	Robometer	TopReward	RoboReward	Robo-Dopamine	WVM
Suboptimal-AgileX	0.11	0.47	0.13	0.36	0.12	0.41	**0.07**
Suboptimal-ARX	0.14	0.50	0.12	0.24	0.17	0.52	**0.05**
Suboptimal-RoboSuite	0.16	0.54	0.16	0.33	0.31	0.51	**0.04**
Average	0.14	0.51	0.14	0.31	0.21	0.49	**0.05**

Table 1 Evaluation of Hesitation-RMSE on Suboptimal-Value-Bench.

accurate prediction achieves an error of zero, whereas fluctuating predictions incur a higher RMSE proportional to their tracking deviation.

4.2 Retry Segments

A retry episode is characterized by a failed manipulation attempt—such as an unsuccessful grasp—followed by a release and retraction phase prior to the subsequent attempt. Since capturing the resulting drop in value is paramount to identifying retries, our evaluation isolates temporal windows that exhibit monotonically decreasing ground-truth progress. We assess a value model's capacity to track this downward trend by restricting the VOC calculation exclusively to these windows, reporting the metric as Retry-VOC. A perfectly tracking, monotonically decreasing prediction yields a maximum score of $+1$, whereas an inverse, monotonically increasing prediction receives the worst-case score of -1.

5 Experiments

We first analyze the quality of value predictions across both Suboptimal-Value-Bench and standard expert trajectories (Section 5.1). We then investigate whether WVM can facilitate downstream policy acquisition (Section 5.2). Finally, we conduct comprehensive ablation studies to dissect the contributions of our key design choices (Section 5.3).

5.1 Value Estimation Quality

Benchmarks and baselines. We evaluate value estimation performance on both Suboptimal-Value-Bench and standard VOC benchmarks. We compare our approach against six competitive baselines: GVL [43], VLAC [73], Robometer [31], TopReward [6], RoboReward [26], and Robo-Dopamine [61]. Detailed standard VOC dataset settings and baseline implementation details are provided in Appendix C and Appendix D, respectively.

Performance on Suboptimal-Value-Bench. As shown in Table 1, WVM consistently achieves the lowest Hesitation-RMSE across all three embodiments. Crucially, it reduces the average error to 0.05, outperforming the strongest baselines, GVL and Robometer (both 0.14), by a substantial margin. These findings validate that WVM provides superior value estimation stability, effectively mitigating prediction drift during periods of task-invariant stagnation. Similarly, Table 2 reveals a comparable performance advantage for WVM within retry phases. Specifically, our method secures the top Retry-VOC score across all embodiments, boosting the average metric from 0.62 to 0.78. Finally, qualitative comparisons in Figure 4 show that the hesitation and

Retry VOC ↑	GVL	VLAC	Robometer	TopReward	WVM
Suboptimal-AgileX	0.73	-0.37	0.32	0.15	**0.79**
Suboptimal-ARX	0.76	/[1]	-0.27	-0.19	**0.79**
Suboptimal-RoboSuite	0.43	/	-0.37	0.00	**0.75**
Average	0.62	-0.37	-0.16	0.00	**0.78**

Table 2 Evaluation of Retry-VOC on Suboptimal-Value-Bench.

[1] "/" marks an ill-defined VOC, which holds consistently for RoboReward and Robo-Dopamine.

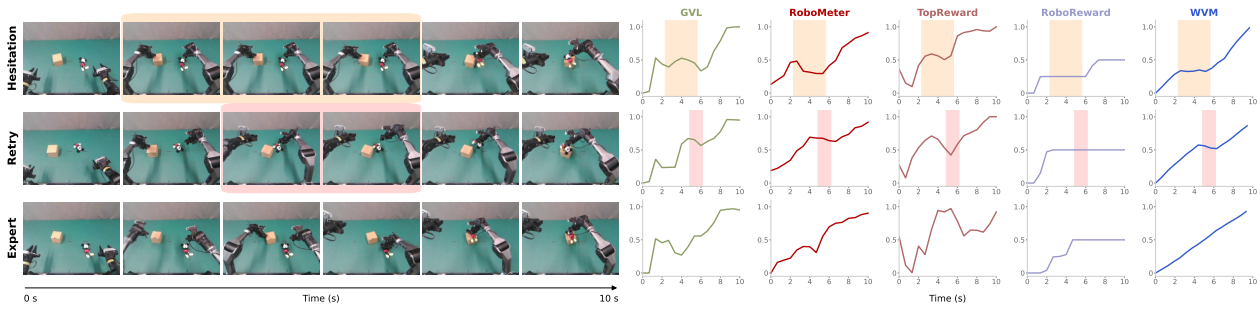


Figure 4 Qualitative value comparison results. From top to bottom: (a) Hesitation: gripper pauses before catching the doll; (b) Retry: gripper backtracks after an unsuccessful grasp; (c) Expert: gripper successfully catches the doll and puts it on the box.

Expert VOC ↑	GVL	VLAC	Robometer	TopReward	RoboReward	Robo-Dopamine	WVM
OXE	0.67	0.48	0.63	0.19	0.92	0.72	**0.94**
RoboCOIN	0.70	0.60	0.77	0.47	0.85	0.75	**0.95**
EgoDex	0.82	0.62	0.86	0.37	**0.95**	0.88	0.92
Self-collected (3 embodiments)	0.93	0.50	0.93	0.58	0.84	0.76	**0.99**
Average	0.78	0.59	0.81	0.42	0.88	0.82	**0.95**

Table 3 Value-Order Correlation on expert demonstrations.

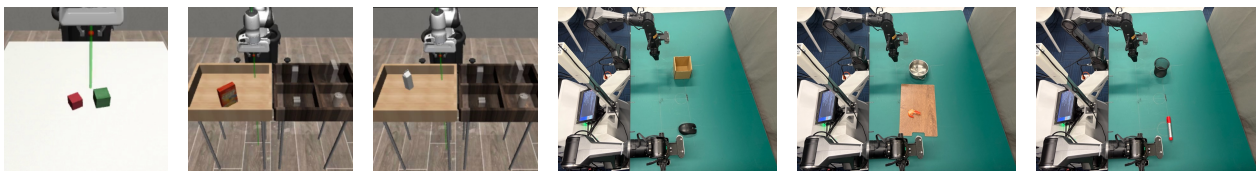
retry segments flagged by WVM align closely with human intuition, consistent with its superior performance on Suboptimal-Value-Bench.

Performance on Expert-VOC. As shown in Table 3, WVM also leads on clean expert trajectories, achieving the highest average VOC score of 0.95 versus 0.88 for the strongest baseline and ranking first on five of the six datasets. It exceeds 0.99 on all three self-collected datasets, confirming strong monotonic tracking on clean demonstrations. The sole exception is EgoDex, where RoboReward slightly outperforms WVM (0.95 vs. 0.92), exposing limitations of Expert-VOC as a metric for value models that we revisit in Section 5.3.

5.2 Downstream Policy Learning

Experimental setup. We evaluate downstream policy learning across three simulated RoboSuite tasks and three real-world AgileX bimanual manipulation tasks, with their experimental setups illustrated in Figure 5. We employ $\pi_{0.5}$-base [20] as our foundational policy. For policy finetuning, we exclusively utilize *suboptimal* data, consisting of only 10 trajectories for each simulated task and 50 trajectories for each real-world task. Additional implementation details are provided in Appendix E.

Policy improvement. Building upon vanilla Behavioral Cloning (BC), we evaluate Advantage Weighted Regression (AWR) [53, 54] alongside two variants of Filtered BC: a binary filter that exclusively retains



(a) Stack cubes (b) Put cereal in tray (c) Put milk in tray (d) Put mouse in box (e) Put shrimp in pot (f) Put marker in holder

Figure 5 Task setups for downstream policy learning experiments.

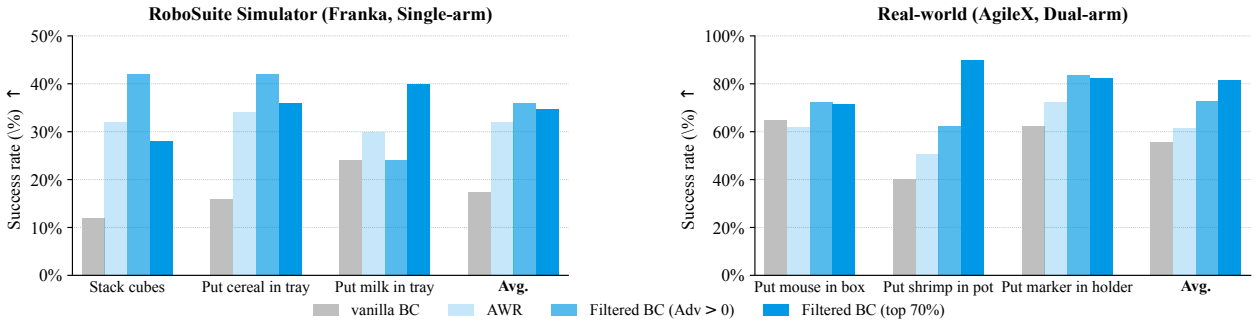


Figure 6 Policy improvement results by combining WVM with AWR and Filtered BC.

Metric	Ours	Video DiT variants			Prefix rand.		Value head
		w/o $\mathcal{L}_{\text{video}}$	scratch	frozen	p=0	p=1	HL-Gaussian
Hesitation-RMSE ↓	**0.05**	0.08	0.08	0.12	0.09	**0.05**	0.06
Retry-VOC ↑	**0.78**	0.68	0.62	0.45	0.67	0.75	0.59
Expert-VOC ↑	0.95	0.95	0.96	0.92	**0.98**	0.91	0.87

Table 4 Ablation results of WVM's design choices.

trajectory segments with positive advantages [30], and a percentile filter that preserves the top 70% of segments ranked by WVM values [19]. As demonstrated in Figure 6, all three WVM-guided variants consistently outperform the vanilla BC baseline across both simulated and physical environments. These experimental results validate that WVM can truly distinguish genuine task progress from suboptimal behaviors, thereby enabling the downstream policy to leverage imperfect data more effectively.

5.3 Ablation Study

To evaluate key design choices of WVM, we conduct ablation studies and summarize results in Table 4. Unless specified otherwise, other components maintain the default configurations.

Video co-training (λ). Omitting the video co-training objective ($\mathcal{L}_{\text{video}}$) consistently degrades performance across all Suboptimal-Value-Bench metrics. Specifically, Hesitation-RMSE increases from 0.05 to 0.08 and Retry-VOC drops from 0.78 to 0.68, even though the pretrained video DiT continues to be optimized via the value-gradient pathway. Furthermore, training the video stream from random initialization reduces Retry-VOC to 0.62, whereas completely freezing the video weights yields the most severe performance degradation (0.12 Hesitation-RMSE and 0.45 Retry-VOC). These empirical findings underscore that explicit video co-training is indispensable for the value model to capture underlying temporal dynamics, thereby validating the central premise of WVM: a continuously co-trained video world model serves as a principled and effective backbone for value estimation.

Prefix randomization (p). The prefix randomization ratio p manages the trade-off between temporal shortcut suppression and inter-chunk continuity. Without randomization (p=0), Hesitation-RMSE worsens to 0.09 and Retry-VOC drops to 0.67, whereas Expert-VOC increases to 0.98. This divergence exposes a pathological over-reliance on prefixes as a causal shortcut, confirming Expert-VOC is an insufficient standalone metric for value models. Conversely, full masking (p=1) recovers retry detection (0.75) but drops Expert-VOC to 0.91 due to disrupted cross-chunk consistency. WVM's p=0.5 achieves the optimal balance, maximizing robustness on Suboptimal-Value-Bench and expert trajectories.

Value head design. Distributional value heads are widely adopted in value learning [3, 10, 19, 30]. Replacing our flow-matching head with an HL-Gaussian alternative (Appendix F) degrades all metrics, with a marginal increase in Hesitation-RMSE ($0.05 \rightarrow 0.06$) and a sharp decline in discriminative scores. This underscores a core limitation of categorical heads: their fixed, pre-specified bin support preserves the conditional mean but discards the fine-grained density variations needed by ordinal metrics. Conversely, our flow-matching head captures a continuous return density without bounding the support or resolution, retaining the local value differentials critical for ranking temporally adjacent chunks [9].

6 Conclusion

We introduce WVM, a generalist robotic value flow model rooted in the predictive capabilities of a pretrained world model. Inheriting its native strengths in historical grounding and future planning, WVM delivers SOTA results across both standard benchmarks and our new Suboptimal-Value-Bench —a multi-embodiment suite of 800 high-fidelity suboptimal trajectories with frame-level human annotations, complementing expert-only evaluations. Downstream deployments in simulation and reality confirm that this world-model-derived architecture offers robust and effective guidance for learning from mixed-quality data.

7 Limitations

While WVM demonstrates strong performance, several limitations remain. First, due to computational constraints, our training dataset is currently limited in scale; consequently, WVM exhibits restricted zero-shot capacity when confronted with entirely unseen tasks and scenes. Second, although Suboptimal-Value-Bench broadens evaluation beyond expert-only demonstrations, its scope is primarily focused on pick-and-place tasks. Expanding the benchmark to more dexterous [30, 76] and long-horizon manipulations [63, 66] is a critical next step. We plan to scale up both training mixture and evaluation diversity in future work.

Appendix

A Implementation Details

Architecture. We build the video stream of WVM upon the publicly released Wan2.2-TI2V-5B checkpoint [64]. The accompanying Wan2.2-VAE compresses input videos by a factor of $4 \times 16 \times 16$ along the temporal and spatial axes, yielding a 48-channel spatiotemporal latent. This latent is further patchified with a patch size of $(1, 2, 2)$ before entering the transformer. The video DiT is a 30-layer Diffusion Transformer with a hidden dimension of 3072 (24 attention heads, head dimension 128) and an FFN width of 14336, totaling approximately 5.0B parameters. The value DiT shares the same depth but operates at a reduced hidden width of 512, configured with 8 self/cross-attention heads (head dimension 64) and a matching FFN width of 14336. At each layer, the two streams are coupled via Mixture-of-Transformers (MoT) self-attention: video tokens retain the inherited Wan2.2 query/key/value projections, whereas value tokens are linearly projected from 512 to the shared attention width of 3072 (24 heads, head dimension 128) to participate in joint attention, before being projected back to 512 at the layer output. Crucially, the MoT attention mask is asymmetric, enabling value tokens to attend to the video latents while ensuring the video stream remains completely unaffected by the value stream. The additional value-side components (projections, cross-attention, FFN, adaptive-norm tables, and input/output heads) contribute roughly 0.7B trainable parameters, bringing the full two-stream transformer to approximately 5.7B parameters. During training, both streams are optimized jointly, while the Wan2.2-VAE is frozen as a tokenizer and the T5 text encoder is executed offline to precompute static text embeddings.

Training. The complete set of hyperparameters used for the main WVM run is summarised in Table A.1.

Category	Setting
Hardware	32× NVIDIA A100-SXM4-40 GB
Wall-clock training time	$\sim$ 40 hours
Optimizer	AdamW
β_1	0.9
β_2	0.95
Weight decay	0
Gradient clipping (max norm)	1.0
Peak learning rate	1×10^{-4}
LR schedule	Cosine decay to 0.1× peak
Warm-up steps	500
Global batch size	1024
Total training steps	30,000
Mixed precision	bfloat16 (bf16)
Prefix randomization ratio p	0.5
Rewind ratio	0.5
Rewind plateau ratio	0.1
Value chunk length h	4
Latent target FPS	2.0 (3.0 for AgileX / ARX self-collected data)
Video co-training weight λ	1.0

Table A.1 Training hyperparameters used for the main WVM run.

Inference. At test time, WVM denoises the value chunk with an explicit Euler solver applied to the learned flow-matching velocity field, and we use only a single denoising step for all reported results. Empirically, using more denoising steps does not yield measurable gains on either Suboptimal-Value-Bench or Expert-VOC. We attribute this to the regime in which WVM operates: relative to its model capacity, the training corpus is moderate in size, so the learned velocity field is sufficiently smooth that one Euler step already lands close to the ground-truth value chunk, and additional refinement provides no further signal. Inference is performed with a chunk size of h=4 and the overlapping-window averaging scheme described in Section 3.2.

Training dataset mixture. The composition of the training mixture used to pre-train WVM is summarised in Table A.2. We report each source along two granularities: "Subsets" counts the distinct subsets enumerated within the source, while "Trajectories" counts demonstrations. The semantics of a subset depends on the source: for the four self-collected sources (RoboSuite, AgileX single-arm, AgileX dual-arm, ARX) , one subset corresponds to exactly one task; for RoboCOIN, EgoDex, and RoboReward, one subset may bundle one or several tasks—specifically, a scene–language pair for RoboCOIN, a top-level task category covering many distinct language instructions for EgoDex, and a constituent sub-dataset for RoboReward. The same convention applies to Table C.4.

Data source	Type	Subsets	Trajectories	Hours	FPS
RoboCOIN [68][2]	Real-world	432	98,171	673.80	30, 50
EgoDex [17]	Real-world	111	299,100	688.56	30
RoboReward [26][3]	Real-world	29	7,428	36.01	10
RoboSuite (ours)	Simulation	6	1,865	11.32	10
AgileX single-arm (ours)	Real-world	4	160	0.39	15
AgileX dual-arm (ours)	Real-world	3	120	0.26	15
ARX (ours)	Real-world	5	242	0.50	15
Total	—	**590**	**407,086**	**1,410.83**	—

Table A.2 Composition of the training mixture used to pre-train WVM. "Subsets" uses the source-specific semantics defined in the surrounding text. "FPS" lists the native frame rates available in each source before resampling.

B Suboptimal-Value-Bench Details

Per-task statistics. Suboptimal-Value-Bench covers three embodiments (AgileX, ARX, RoboSuite) and 15 manipulation tasks, with each task instantiated as two trajectory groups corresponding to the two suboptimal modes (hesitation and retry). The full per-task breakdown is provided in Table B.3.

Annotation tool. Manually labelling every frame of 800 trajectories is prohibitively expensive. To accelerate annotation while preserving label quality, we adopt a two-stage pipeline. In the first stage, each trajectory is fed to a proprietary large vision-language model through its public API to obtain an initial coarse segmentation of non-progress intervals. Frames are sampled at a downsampled frame rate, prefixed with their index, and submitted together with the natural-language task description using the following prompt:

```
Below are frames sampled from a {fps:.0f}fps robot manipulation video
(total {frames[-1][0]} frames). Each frame is preceded by its frame
index label [frame=N].

{frames_block}

The task is: "{task_description}".

Please analyze the full trajectory and identify segments where the
robot is NOT making forward progress toward completing the task.
A segment is "not forward progress" if during that time the task
state is stalled or regressing, for example:
- The robot is stuck, hesitating, or repeating a motion without
  advancing the task.
- The task state is going backwards (e.g. an object was dropped,
  knocked away, or released unnecessarily).
- The robot is performing motions that do not bring it closer to
  task completion.

Do NOT flag segments where the robot is actively and successfully
```

[2]We use the snapshot of all RoboCOIN data available as of 2026-01-12.

[3]We use the train split of the RoboReward open-source release and retain only the successful demonstrations with the maximum reward label of 5. For the DROID portion, which provides both left- and right-camera views, we keep only the left view.

Embodiment	Task	Arm	Hesitation	Retry	Total	Duration (min)
AgileX (real)	carrot off plate	single	25	25	50	9.9
	carrot on plate	single	25	25	50	10.1
	mickey box	dual	25	25	50	11.1
	sausage pot	dual	25	25	50	11.0
	Subtotal	—	*100*	*100*	*200*	*42.0*
ARX (real)	flip bottles	dual	15	15	30	5.1
	open box	dual	20	20	40	7.4
	split cups	dual	15	15	30	5.9
	stack bowls	dual	50	50	100	16.3
	stack cups	dual	50	50	100	16.6
	Subtotal	—	*150*	*150*	*300*	*51.3*
RoboSuite (sim)	lift	single	25	25	50	14.0
	pick & place bread	single	25	25	50	20.4
	pick & place can	single	25	25	50	23.3
	pick & place cereal	single	25	25	50	23.3
	pick & place milk	single	25	25	50	21.3
	stack	single	25	25	50	17.6
	Subtotal	—	*150*	*150*	*300*	*119.9*
Total	—	—	**400**	**400**	**800**	**213.2**

Table B.3 Per-task composition of Suboptimal-Value-Bench. Each task contributes two trajectory groups, one per suboptimal mode (hesitation and retry); the columns "Hesitation" and "Retry" report the trajectory count in each group.

```
advancing toward the goal, even if the motion is slow.

Output a JSON object with the following format (NO extra text
outside the JSON):
{
  "non_progress_segments": [
    {
      "start_frame": <int>,
      "end_frame": <int>,
      "description": "<brief description of why this segment is
                      not forward progress>"
    }
  ],
  "task_completed": <true or false>,
  "summary": "<one-sentence summary>"
}

If the entire trajectory is efficient with no wasted time, return
an empty non_progress_segments list.
```

The model returns, for every trajectory, a set of candidate non-progress intervals together with a one-sentence rationale. In the second stage, human annotators inspect each candidate interval in a custom labelling interface (Figure B.1), correct its boundaries at frame-level resolution. The interface displays the trajectory video together with a frame-aligned timeline of the VLM-proposed segments, and allows annotators to drag segment endpoints, split or merge intervals, and re-play any sub-clip. Trajectories whose VLM proposal is empty are

still presented to annotators in full to avoid silent omissions. This pre-segmentation step substantially reduces the search effort of human labellers while leaving all final boundaries and labels under human control.

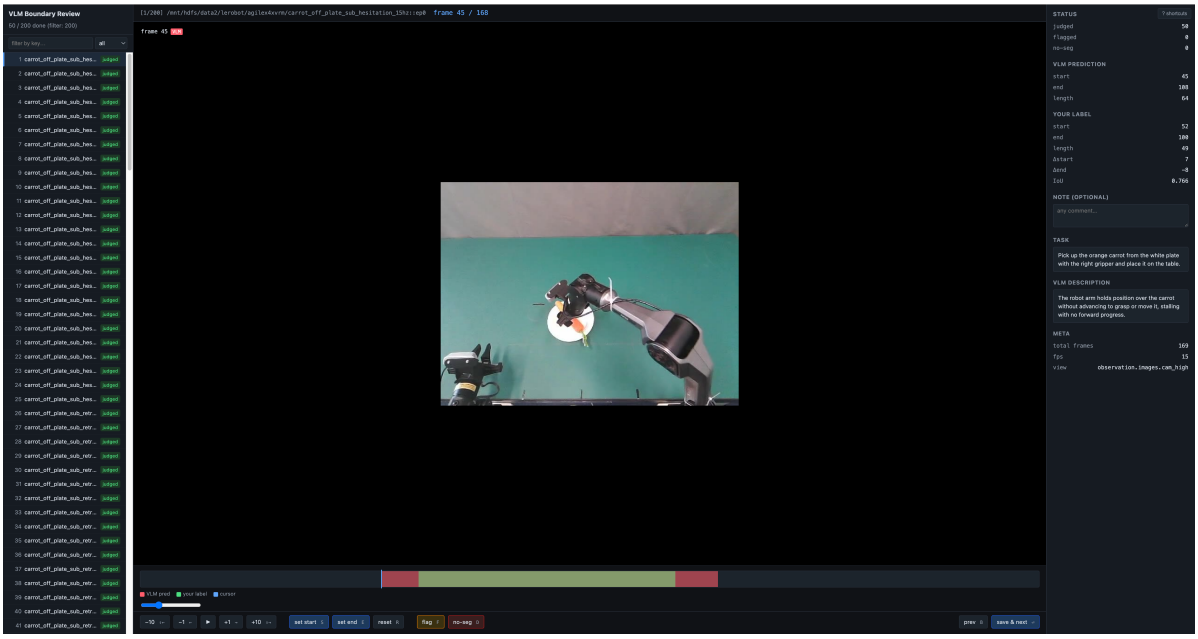


Figure B.1 Human verification interface used to refine the VLM-proposed non-progress intervals. Annotators can replay the trajectory video, adjust segment boundaries at frame-level resolution.

Per-frame label to ground-truth value. For every trajectory the human annotation provides a suboptimal type (*hesitation* or *retry*) together with three integers: the inclusive endpoints m, n $(0 < m < n < T-1)$ of the suboptimal segment and the total trajectory length T. We then construct the per-frame ground-truth value $v_t \in [0, 1]$ as a four-point piecewise-linear curve through the control points

$$(0,\, 0), \quad (m,\, v_m), \quad (n,\, v_n), \quad (T-1,\, 1), \tag{B.1}$$

with v_t obtained by linear interpolation between adjacent control points (frames outside $[0, T-1]$ are clamped to 0 or 1). Let $x = n - m$ denote the length of the suboptimal segment. The values v_m, v_n are chosen so that the resulting curve reflects the semantics of each suboptimal mode:

Hesitation. During $[m, n]$ the robot does not advance the task—it either remains stationary or executes task-irrelevant micro-movements—and then resumes at the same effective speed, so the segment is a plateau in progress space and the remaining $T - 1 - x$ effective frames evenly cover the unit progress:

$$v_m \;=\; v_n \;=\; \frac{m}{T-1-x}. \tag{B.2}$$

The three slopes become $\frac{1}{T-1-x}$, 0, and $\frac{1}{T-1-x}$, giving a piecewise curve that advances at a constant effective rate before and after the plateau.

Retry. We model the retry as a uniform-speed retraction: during $[m, n]$ the robot moves backward at the same per-frame rate r at which it normally advances forward, an assumption we find well supported by the retry trajectories in our data. Out of the T trajectory frames, x are spent retracting and a further x forward frames are needed to re-earn the lost progress, leaving only $T - 2x$ frames of net forward motion to cover the unit interval; hence $r = 1/(T - 2x)$. Concretely,

$$v_m \;=\; \frac{m}{T-2x}, \qquad v_n \;=\; \max\!\Big(0,\; \tfrac{m-x}{T-2x}\Big). \tag{B.3}$$

When the retry happens late enough that $n \leq 2m$, the three segments share the same absolute slope $\frac{1}{T-2x}$ with the middle segment negative, producing a symmetric "V" over the retry interval. When the retry is unusually long ($n > 2m$), v_n would otherwise be negative and is clamped to 0, so the curve drops linearly from v_m to 0 and then re-climbs to 1 over the remaining $(T-1-n)$ frames; this preserves the semantics that progress cannot fall below zero.

We use these piecewise-linear curves as the ground truth for both the Hesitation-RMSE and Retry-VOC metrics in Section 4.

C Expert VOC Dataset Composition

The composition of the evaluation set used for Expert-VOC is summarised in Table C.4. It covers three public corpora (OXE, RoboCOIN, EgoDex) and four self-collected expert sets that match the embodiments used in Suboptimal-Value-Bench. All trajectories are held out from the WVM training mixture in Table A.2 at the trajectory level to avoid leakage.

Data source	Type	Subsets	Trajectories	Hours	FPS
RoboCOIN [68]	Real-world	318	641	3.02	30
EgoDex [17]	Real-world	13	588	1.04	30
OXE [47]	Real-world	10	146	0.58	10
AgileX single-arm (ours)	Real-world	4	40	0.10	15
AgileX dual-arm (ours)	Real-world	3	30	0.07	15
ARX (ours)	Real-world	5	61	0.13	15
RoboSuite (ours)	Simulation	6	99	0.60	10
Total	—	**359**	**1,605**	**5.52**	—

Table C.4 Composition of the evaluation set used for Expert-VOC. "Subsets" follows the same source-specific semantics as in Table A.2 (see the "Training dataset mixture" paragraph for the per-source definition). The four self-collected sources cover the same three embodiments (AgileX, ARX, RoboSuite) used by Suboptimal-Value-Bench, and are reported as a single "Self-collected (3 embodiments)" row in the main paper (Table 3).

Overall protocol. Unless stated otherwise, the Expert-VOC pool for each data source is obtained by uniformly sampling 5% of its trajectories at random; the remaining 95% are retained in the WVM training mixture (Table A.2), so the two splits are disjoint at the trajectory level and the held-out set is never seen during training. The source-specific deviations from this default are detailed below.

OXE [47]. Our OXE trajectories are drawn from the RoboReward [26] open-source release rather than the raw OXE distribution; concretely, we re-use exactly the same filtered pool (train split, reward = 5 successes, DROID left view only) described in the footnote of Table A.2, and the per-source selection protocol below operates on top of that pool. OXE itself aggregates many sub-datasets whose trajectory quality varies substantially, so a uniform random 5% sample over this pool would still pull low-quality trajectories into the expert set. We adopt the per-dataset VOC ranking reported by GVL [43] as a quality proxy. GVL observes that several large sub-datasets, most notably DROID [24], are ranked very low, consistent with prior reports that removing DROID from large action-model training improves final performance; on inspection of the low-VOC DROID clips, GVL attributes the low scores to poor camera placement that fails to capture the robot's motion and to heavy occlusions of the arm and the manipulated object, both of which would directly bias an expert-progress evaluator. We therefore restrict OXE to the ten sub-datasets with the highest VOC scores reported by GVL, and then apply the default 5% random sample within those sub-datasets, yielding the 146 expert trajectories listed in Table C.4.

AgileX and ARX (ours). The self-collected AgileX (single-arm and dual-arm) and ARX trajectories are teleoperated by trained operators and are uniformly of high quality, so no additional quality filtering is needed. To obtain a sufficiently large Expert-VOC pool for stable per-embodiment estimates given the small absolute size of these sources, we raise the sampling ratio from the default 5% to 20% for all three subsets.

RoboSuite (ours). RoboSuite trajectories are generated by a scripted oracle policy and are uniformly expert, so no quality filtering is required either. Because RoboSuite already contributes a large number of trajectories, the default 5% random sample alone produces a sufficiently large expert pool (99 trajectories), and we apply no additional adjustment.

RoboCOIN [68]. RoboCOIN provides multiple successful demonstrations per language instruction and the per-demonstration execution speed varies substantially. Following the common heuristic that, among successful demonstrations of the same instruction, the faster ones tend to be closer to expert behaviour, we enumerate the distinct language instructions in RoboCOIN and, for each instruction, keep the single shortest-duration trajectory as the Expert-VOC sample (rather than using the default random 5%).

EgoDex [17]. EgoDex covers a wide range of egocentric human activities, many of which are unrelated to robotic manipulation. We therefore first shortlist the EgoDex task categories that correspond to basic pick-and-place manipulation, since these are the activities most directly relevant to manipulation-oriented value estimation. Within each shortlisted category we then enumerate the distinct language instructions and, for each instruction, keep the single shortest demonstration, using the same speed-as-expertise heuristic as for RoboCOIN.

D Value Model Baseline Reproduction

All six baselines are evaluated under a unified video-sampling pipeline: each trajectory is first downsampled to a common target frame rate (2 fps, or 3 fps for the AgileX and ARX datasets to keep per-trajectory frame counts comparable across embodiments), and we then evaluate each baseline using its officially recommended sampling protocol (multi-anchor prefix evaluation for Robometer and RoboReward; single-pass full-trajectory evaluation for the remaining four). Our evaluation harness directly builds on the public Robometer [31] codebase.

GVL [43]. GVL casts value estimation as autoregressive completion-percentage prediction over shuffled video frames, where a close-source VLM is prompted with a task description together with a sequence of frame–index pairs and asked to output a task-progress percentage for each frame. We use the public API of `gpt-5.4` as the backbone, with a per-call frame budget of 32 frames.

VLAC [73]. VLAC fine-tunes an InternVL backbone into a unified action–critic model that, given a pair of image observations and a language goal, autoregressively emits a signed progress delta together with a task-completion (done) signal, providing dense rewards for downstream RL. We use the public checkpoint `InternRobotics/VLAC` in single-pass mode with 32 frames per trajectory and the default decoding configuration (temperature 0.5, batch size 32); we keep the original `frame_skip` schedule and disable the auxiliary image branch (`use_images=false`).

Robometer [31]. Robometer fine-tunes Qwen3-VL-4B with a composite objective combining a frame-level progress loss, a per-frame success loss, and an inter-trajectory preference loss, enabling it to learn from both expert and suboptimal/failed trajectories and to emit dense per-frame progress estimates over short prefix clips. We use the public checkpoint `robometer/Robometer-4B` with the official multi-anchor evaluation: 5 uniformly spaced anchors per trajectory, 8 frames per anchor (`max_frames=8`, `use_frame_steps=true`, `subsample_n_frames=5`), and a model batch size of 32.

RoboReward [26]. RoboReward fine-tunes Qwen3-VL at the 4B and 8B scales to predict a discrete end-of-episode progress score in $\{1, \ldots, 5\}$ from a task description and a full rollout video; following the original setup, dense per-frame rewards are obtained by re-querying the model on partial-trajectory prefixes. We use the public checkpoint `teetone/RoboReward-4B` with the same multi-anchor protocol as Robometer (5 anchors, `use_frame_steps=true`, `subsample_n_frames=5`) but raise `max_frames` to 32 and cap the generation length at 128 tokens, matching the original setup.

TopReward [6]. TopReward turns a frozen video VLM into a zero-shot temporal value function by reading off the log-probability that the model assigns to an affirmative completion token (e.g., `True`) when asked whether a video prefix has completed the instruction; in our setup, we follow the official protocol and mean-aggregate this score over K prefix samples per trajectory. We use `Qwen/Qwen3-VL-8B-Instruct` as the backbone with $K = 15$ prefix samples, mean reduction, the official chat template, and a 2 fps temporal sampling; each trajectory is consumed once with `max_frames=32`.

Robo-Dopamine [61]. Robo-Dopamine is a 3B-parameter step-aware generative process reward model (the GRM in the Robo-Dopamine framework) that, conditioned on a task description and multi-view "before"/"after" frames, predicts a discretised relative progress hop and supports an incremental inference mode that emits a per-frame progress signal as frames are streamed in. We use the public checkpoint `tanhuajie2001/Robo-Dopamine-GRM-3B` in incremental mode (`eval_mode="incremental"`, `frame_interval=1`, batch size 1) and feed up to 32 uniformly sampled frames per trajectory in a single pass.

E Downstream Policy Learning Details

The hyperparameters used for all downstream policy-learning experiments in Section 5.2 are summarised in Table E.5.

Chunk-level advantage proxy. All weighting schemes operate on the same chunk-level advantage proxy derived from WVM. For a sample anchored at frame t with action-chunk length H, let $t^{\text{head}} = t$ and $t^{\text{tail}} = \min(t + H - 1, T - 1)$, and let $V(\cdot) \in [0, 1]$ denote the per-frame value emitted by WVM. We define

$$\Delta_i \;=\; V(t_i^{\text{tail}}) \;-\; V(t_i^{\text{head}}), \tag{E.4}$$

which approximates the value improvement contributed by the action chunk and plays the role of an advantage estimate in the weighting schemes below. Given per-sample BC losses ℓ_i (e.g. flow-matching loss for $\pi_{0.5}$-base) and weights w_i, the weighted-BC objective used throughout this section is

$$\mathcal{L} \;=\; \frac{\sum_{i=1}^{B} w_i\, \ell_i}{\sum_{i=1}^{B} w_i + \varepsilon}, \qquad \varepsilon = 10^{-6}. \tag{E.5}$$

Filtered BC. The two Filtered-BC variants in Section 5.2 both use a hard-threshold indicator on Δ_i:

$$w_i \;=\; \mathbf{1}[\,\Delta_i \;\geq\; \kappa\,]. \tag{E.6}$$

The *binary* variant uses $\kappa = 0.0$, which simply discards chunks on which the value does not improve. The *percentile* variant retains the top 70% of chunks ranked by Δ_i over the training set; the corresponding threshold κ is reported per task suite in Table E.5.

Advantage-Weighted Regression (AWR). AWR replaces the hard indicator with a clipped exponential weight on the same advantage proxy:

$$w_i \;=\; \min\Big(\exp\big(\tau \cdot \Delta_i\big),\; \delta\Big). \tag{E.7}$$

This recovers the standard AWR/AWAC weighting [53, 54] with temperature $1/\beta = \tau$, advantage $A = \Delta_i$, and clipping ceiling $M = \delta$. We fix $\delta = 2.0$ across both task suites and adjust the temperature τ per suite (see Table E.5) to match the empirical scale of Δ_i at each suite's action-chunk length H. The clip caps the highest-advantage chunks at $2\times$ their baseline contribution, preventing a few high-advantage outliers from dominating the gradient. We further renormalise the per-batch weights so that $\frac{1}{B}\sum_i w_i = 1$, keeping the gradient scale aligned with vanilla BC and allowing us to reuse the same learning rate and weight decay as the BC baseline.

For evaluation, each RoboSuite task is rolled out for 50 trials, while each real-world AgileX task is rolled out for 30 trials, and we report the average success rate over these trials.

Category	Setting
Shared	
Base policy	$\pi_{0.5}$-base
Hardware	16× NVIDIA A100-SXM4-40 GB
Optimizer	AdamW
β_1	0.9
β_2	0.95
Peak learning rate	2.5×10^{-5}
Global batch size	256
AWR clip ceiling δ	2.0
Filtered-BC (binary) threshold κ	0.0
Task-suite specific (RoboSuite / AgileX)	
Control mode	EEF / Joint
Total training steps	5,000 / 10,000
Wall-clock training time	$\sim$ 4.5 h / $\sim$ 9 h
Action-chunk length H	10 / 50
AWR temperature τ	10 / 2
Filtered-BC (top-70%) threshold κ	0.02 / 0.06

Table E.5 Hyperparameters used for downstream policy fine-tuning in Section 5.2. The action-chunk length, AWR temperature, and top-70% Filtered-BC threshold are set per task suite to account for the different per-frame value-change scales induced by different chunk lengths.

F HL-Gaussian Value Head Details

The HL-Gaussian ablation in Section 5.3 replaces WVM's flow-matching value head with a discrete distributional head [10] that reformulates per-frame value regression as classification over K fixed bins. All non-head components (video stream, value stream, MoT coupling, prefix randomization, video rewinding) are kept identical to WVM, so the only varying factor is the value head itself.

Bin support and soft targets. We discretise the support $[0, 1]$ with $K = 51$ equally-spaced bin centres $c_k = (k-1)/(K-1)$ for $k = 1, \ldots, K$. Given a ground-truth value $v \in [0, 1]$, the soft target distribution is a Gaussian-smoothed one-hot over the bins:

$$p_k(v) \;=\; \frac{\exp\!\big(-(v-c_k)^2/2\sigma^2\big)}{\sum_{j=1}^{K} \exp(-(v-c_j)^2/2\sigma^2)}, \qquad \sigma \;=\; \frac{1}{K-1}. \tag{F.8}$$

We set σ to one bin width by default, which gives smooth-but-locally-peaked targets; values within 10^{-6} of the endpoints 0 or 1 are snapped to a hard one-hot at bin 1 or bin K to avoid leaking probability mass onto neighbouring bins at the boundaries.

Training objective. The value DiT outputs K logits $z \in \mathbb{R}^K$ at every (frame, sub-step) token. We train it with a token-level soft cross-entropy against the encoded target distribution, which is equivalent to minimising the KL divergence up to a constant entropy term:

$$\mathcal{L}_{\text{value}}^{\text{HLG}} \;=\; \mathbb{E}_v\left[-\sum_{k=1}^{K} p_k(v)\, \log \operatorname{softmax}(z)_k\right]. \tag{F.9}$$

This replaces the flow-matching loss $\mathcal{L}_{\text{value}}$ in Eq. 5; the video co-training loss $\mathcal{L}_{\text{video}}$ and the overall objective weight λ are kept unchanged from WVM.

Inference. Unlike the flow-matching head, the HL-Gaussian head does not require any iterative denoising: a single transformer forward yields K logits per (frame, sub-step) token, and we decode the predicted scalar

as the expectation of the softmax distribution over the bin centres:

$$\hat{v} \;=\; \sum_{k=1}^{K} \mathrm{softmax}(z)_k \cdot c_k. \tag{F.10}$$

We then apply the same overlapping-window averaging scheme described in Section 3.2 to assemble per-frame value predictions across adjacent chunks.

References

[1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985, 2025.

[2] Bram Bakker. Reinforcement learning with long short-term memory. Advances in neural information processing systems, 14, 2001.

[3] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In International conference on machine learning, pages 449–458. Pmlr, 2017.

[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2):157–166, 1994.

[5] Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. arXiv preprint arXiv:2512.05964, 2025.

[6] Shirui Chen, Cole Harrison, Ying-Chun Lee, Angela Jin Yang, Zhongzheng Ren, Lillian J Ratliff, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Topreward: Token probabilities as hidden zero-shot rewards for robotics. arXiv preprint arXiv:2602.19313, 2026.

[7] Shivin Dass, Alaa Khaddaj, Logan Engstrom, Aleksander Madry, Andrew Ilyas, and Roberto Martín-Martín. Datamil: Selecting data for robot imitation learning with datamodels. arXiv preprint arXiv:2505.09603, 2025.

[8] Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, et al. Understanding world or predicting future? a comprehensive survey of world models. ACM Computing Surveys, 58(3):1–38, 2025.

[9] Perry Dong, Chongyi Zheng, Chelsea Finn, Dorsa Sadigh, and Benjamin Eysenbach. Value flows. arXiv preprint arXiv:2510.07650, 2025.

[10] Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, et al. Stop regressing: Training value functions via classification for scalable deep rl. arXiv preprint arXiv:2403.03950, 2024.

[11] Samuel Garcin, Trevor McInroe, Pablo Samuel Castro, Christopher Lucas, David Abel, Prakash Panangaden, and Stefano V Albrecht. Studying the interplay between the actor and critic representations in reinforcement learning. In International Conference on Learning Representations, volume 2025, pages 35590–35616, 2025.

[12] David Ha and Jürgen Schmidhuber. World models. arXiv preprint arXiv:1803.10122, 2(3):440, 2018.

[13] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603, 2019.

[14] Matthew J Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In AAAI fall symposia, volume 45, page 141, 2015.

[15] Joey Hejna, Chethan Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Re-mix: Optimizing data mixtures for large scale imitation learning. arXiv preprint arXiv:2408.14037, 2024.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022.

[17] Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. arXiv preprint arXiv:2505.11709, 2025.

[18] Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value. Nature communications, 10(1):5223, 2019.

[19] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. arXiv preprint arXiv:2511.14759, 2025.

[20] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054, 2025.

[21] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi^*_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. arXiv preprint arXiv:2604.15483, 2026.

[22] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. Artificial intelligence, 101(1-2):99–134, 1998.

[23] Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In International conference on learning representations, 2018.

[24] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. arXiv preprint arXiv:2403.12945, 2024.

[25] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. arXiv preprint arXiv:2601.16163, 2026.

[26] Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics. arXiv preprint arXiv:2601.00675, 2026.

[27] Jianxiong Li, Jinliang Zheng, Yinan Zheng, Liyuan Mao, Xiao Hu, Sijie Cheng, Haoyi Niu, Jihao Liu, Yu Liu, Jingjing Liu, et al. Decisionnce: Embodied multimodal representations via implicit preference learning. In Forty-first International Conference on Machine Learning, 2024.

[28] Jianxiong Li, Zhihao Wang, Jinliang Zheng, Xiaoai Zhou, Guanming Wang, Guanglu Song, Yu Liu, Jingjing Liu, Ya-Qin Zhang, Junzhi Yu, et al. Robo-mutual: Robotic multimodal task specification via unimodal learning. In 2025 IEEE International Conference on Robotics and Automation (ICRA), pages 4182–4189. IEEE, 2025.

[29] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. arXiv preprint arXiv:2601.21998, 2026.

[30] Yunfei Li, Xiao Ma, Jiafeng Xu, Yu Cui, Zhongren Cui, Zhigang Han, Liqun Huang, Tao Kong, Yuxiao Liu, Hao Niu, et al. Gr-rl: Going dexterous and precise for long-horizon robotic manipulation. arXiv preprint arXiv:2512.01801, 2025.

[31] Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S. Huang, Luke Zettlemoyer, Dieter Fox, Yu Xiang, Anqi Li, Andreea Bobu, Abhishek Gupta, Stephen Tu, Erdem Biyik, and Jesse Zhang. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. arXiv preprint arXiv:2603.02115, 2026.

[32] Ruiming Liang, Yinan Zheng, Kexin Zheng, Tianyi Tan, Jianxiong Li, Liyuan Mao, Zhihao Wang, Guang Chen, Hangjun Ye, Jingjing Liu, Jinqiao Wang, and Xianyuan Zhan. Dichotomous diffusion policy optimization. arXiv preprint arXiv:2601.00898, 2026.

[33] Weixin Liang, LILI YU, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. Transactions on Machine Learning Research, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Nu6N69i8SB.

[34] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. arXiv preprint arXiv:2210.02747, 2022.

[35] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. arXiv preprint arXiv:2412.06264, 2024.

[36] Dongxiu Liu, Haoyi Niu, Zhihao Wang, Jinliang Zheng, Yinan Zheng, Zhonghong Ou, Jianming Hu, Jianxiong Li, and Xianyuan Zhan. Efficient robotic policy learning via latent space backward planning. arXiv preprint arXiv:2505.06861, 2025.

[37] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003, 2022.

[38] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv preprint arXiv:2402.17177, 2024.

[39] Jindi Lv, Hao Li, Jie Li, Yifei Nie, Fankun Kong, Yang Wang, Xiaofeng Wang, Zheng Zhu, Chaojun Ni, Qiuping Deng, et al. Viva: A video-generative value model for robot reinforcement learning. arXiv preprint arXiv:2604.08168, 2026.

[40] Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. arXiv preprint arXiv:2603.10448, 2026.

[41] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. arXiv preprint arXiv:2210.00030, 2022.

[42] Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In International Conference on Machine Learning, pages 23301–23320. PMLR, 2023.

[43] Yecheng Jason Ma, Joey Hejna, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, et al. Vision language models are in-context value learners. In The Thirteenth International Conference on Learning Representations, 2024.

[44] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworldmodel: Stable end-to-end joint-embedding predictive architecture from pixels. arXiv preprint arXiv:2603.19312, 2026.

[45] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. arXiv preprint arXiv:2203.12601, 2022.

[46] Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. Advances in neural information processing systems, 30, 2017.

[47] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 6892–6903. IEEE, 2024.

[48] Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. arXiv preprint 2512.15692, 2025.

[49] Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, et al. Sop: A scalable online post-training system for vision-language-action models. arXiv preprint arXiv:2601.03044, 2026.

[50] Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In International conference on machine learning, pages 7487–7498. PMLR, 2020.

[51] Karl Pearson. Vii. note on regression and inheritance in the case of two parents. proceedings of the royal society of London, 58(347-352):240–242, 1895.

[52] 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.

[53] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. arXiv preprint arXiv:1910.00177, 2019.

[54] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In Proceedings of the 24th international conference on Machine learning, pages 745–750, 2007.

[55] Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. A survey of temporal credit assignment in deep reinforcement learning. arXiv preprint arXiv:2312.01072, 2023.

[56] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37

of Proceedings of Machine Learning Research, pages 1312–1320, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schaul15.html.

[57] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. Nature, 588(7839):604–609, 2020.

[58] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. arXiv preprint arXiv:2604.14148, 2026.

[59] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484–489, 2016.

[60] Richard S Sutton, Andrew G Barto, et al. Reinforcement learning: An introduction, volume 1. MIT press Cambridge, 1998.

[61] Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Yuheng Ji, Cheng Chi, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, et al. Robo-dopamine: General process reward modeling for high-precision robotic manipulation. arXiv preprint arXiv:2512.23703, 2025.

[62] MotuBrain Team, Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, et al. Motubrain: An advanced world action model for robot control. arXiv preprint arXiv:2604.27792, 2026.

[63] Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, et al. Mem: Multi-scale embodied memory for vision language action models. arXiv preprint arXiv:2603.03596, 2026.

[64] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314, 2025.

[65] Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, Mike Zheng Shou, Xuanjing Huang, Xipeng Qiu, and Yu-Gang Jiang. World action models: The next frontier in embodied ai, 2026. URL https://arxiv.org/abs/2605.12090.

[66] Zhihao Wang, Jianxiong Li, Jinliang Zheng, Wencong Zhang, Dongxiu Liu, Yinan Zheng, Haoyi Niu, Junzhi Yu, and Xianyuan Zhan. Physiagent: An embodied agent framework in physical world. arXiv preprint arXiv:2509.24524, 2025.

[67] Daan Wierstra, Alexander Foerster, Jan Peters, and Juergen Schmidhuber. Solving deep memory pomdps with recurrent policy gradients. In International conference on artificial neural networks, pages 697–706. Springer, 2007.

[68] Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. arXiv preprint arXiv:2511.17441, 2025.

[69] Charles Xu, Jost Tobias Springenberg, Michael Equi, Ali Amin, Adnan Esmail, Sergey Levine, and Liyiming Ke. Rl token: Bootstrapping online rl with vision-language-action models. arXiv preprint arXiv:2604.23073, 2026.

[70] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. arXiv preprint arXiv:2603.17240, 2026.

[71] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

[72] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? arXiv preprint arXiv:2603.16666, 2026. URL https://arxiv.org/abs/2603.16666.

[73] Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. A vision-language-action-critic model for robotic real-world reinforcement learning. arXiv preprint arXiv:2509.15937, 2025.

[74] Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. RewiND: Language-guided rewards teach robot policies without new demonstrations. In 9th Annual Conference on Robot Learning, 2025. URL https://openreview.net/forum?id=XjjXLxfPou.

[75] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 22508–22519, 2025.

[76] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. arXiv preprint arXiv:2510.10274, 2025.