UR-VC: Unsupervised Robotic Value Correction for Time-Derived Progress Proxies

Lirui Zhao, Modi Shi, Li Chen, Qi Liu, Ping Luo, Hongyang Li
The University of Hong Kong

***Abstract*—Modern robot learning systems increasingly rely on dense progress or value signals to evaluate intermediate states, guide policy learning, and detect task completion, making the quality of these signals critical. Since such dense labels are rarely available at scale, normalized time within a demonstration is often used as a scalable substitute: later frames are treated as higher progress. However, this time-derived label is only a noisy proxy for physical task progress. In contact-rich manipulation, a robot may make progress and then lose it through slips, failed grasps, or partial undoing, while the time-derived label continues to increase monotonically. We introduce Unsupervised Robotic Value Correction (UR-VC), an offline, training-free method for correcting time-derived progress labels. UR-VC exploits a simple regularity in demonstration data: similar states often recur across different episodes, but at different timestamps. Instead of trusting the timestamp from a single trajectory, UR-VC retrieves similar states from other episodes and aggregates their time-derived labels to obtain a corrected progress estimate. UR-VC requires no manual progress labels, reward annotations, or additional value model. We evaluate UR-VC on real bimanual cloth flatten-and-fold data, a long-horizon deformable-object manipulation task with visible intermediate progress. The corrected labels capture local regressions and non-uniform progress that normalized time cannot represent, while preserving the overall task trend. We further use the corrected signal to construct advantage labels for VLA training, following recent advantage-conditioned policy learning. UR-VC shows a positive trend in real-robot task success under matched data, model, and training settings.**

I. Introduction

Modern robot learning systems increasingly use intermediate task signals in addition to action supervision. In long-horizon manipulation, such signals estimate how far the current state has advanced toward task completion and whether recent actions have moved the task forward. They commonly take the form of task progress, value, or advantage, and are used for completion detection, value-function learning, and advantage-conditioned policy learning [1, 25, 20, 22, 6, 9]. As vision–language–action (VLA) models are trained on larger and more diverse robot datasets [5, 15, 4, 3], the quality of these dense signals becomes increasingly critical.

The difficulty is that dense progress or value labels are rarely available at scale. A common scalable substitute is normalized time within a successful demonstration [1, 25], which treats later frames as higher progress since they usually correspond to states closer to task completion. However, the resulting time-derived label is only a noisy proxy for physical task progress. It assumes that progress increases monotonically and uniformly with time, while real manipulation, especially contact-rich and deformable manipulation [16, 8], can regress, stall, or advance at different rates. For example, in the real cloth flatten-and-fold episode (Fig. 1), the robot first smooths the garment, but a later grasp slip reintroduces wrinkles and moves the task state backward, so physical progress regresses even as the time-derived label continues to increase linearly.

This mismatch matters for downstream learning in two ways. Within an episode, this directly affects advantage-style supervision: under a fixed-horizon progress difference, a linear time proxy assigns the same increase to every frame, so it cannot distinguish actions that improve the physical state from actions that stall or undo progress. Across episodes, the problem is subtle but more persistent: similar physical states can receive different labels simply because demonstrations proceed at different speeds or take different recovery motions. A model trained on these noisy labels may then explain label variation using incidental cues such as appearance, texture, or execution speed, rather than physical task progress.

A common response is to learn this signal from data, for example by training progress estimators or value models using temporal contrast, language–image objectives, value pretraining, optimal transport, or generative modelling [23, 18, 19, 2, 11, 13], or by eliciting value estimates from pretrained models [20, 7, 17]. However, when the supervision target is a systematically biased time-derived proxy, a learned estimator can inherit the same bias: it may smooth over local regressions and encode differences in execution speed or recovery behavior as differences in progress. This motivates a complementary question: *can the proxy labels themselves be corrected before they serve as supervision for estimators or policies?*

We introduce **Unsupervised Robotic Value Correction (UR-VC)**, an offline, training-free method for correcting time-derived progress labels in demonstration datasets. UR-VC is based on a simple data-driven premise: time-derived labels are noisy mainly because each demonstration has its own temporal distortion. A timestamp reflects both physical task progress and episode-specific factors such as execution speed, pauses, failures, and recovery motions. Cross-episode state matches provide anchors for reducing this distortion: when similar physical states are found in independently collected episodes, the state is approximately held fixed while the timestamp varies across trajectories. Aggregating the time-derived labels of these matched states therefore estimates where that state typically lies in the task, rather than where it happened to occur in one particular episode. UR-VC operationalizes this idea by retrieving semantically similar frames from other episodes using SigLIP-2 visual embeddings [26] and aggregating their

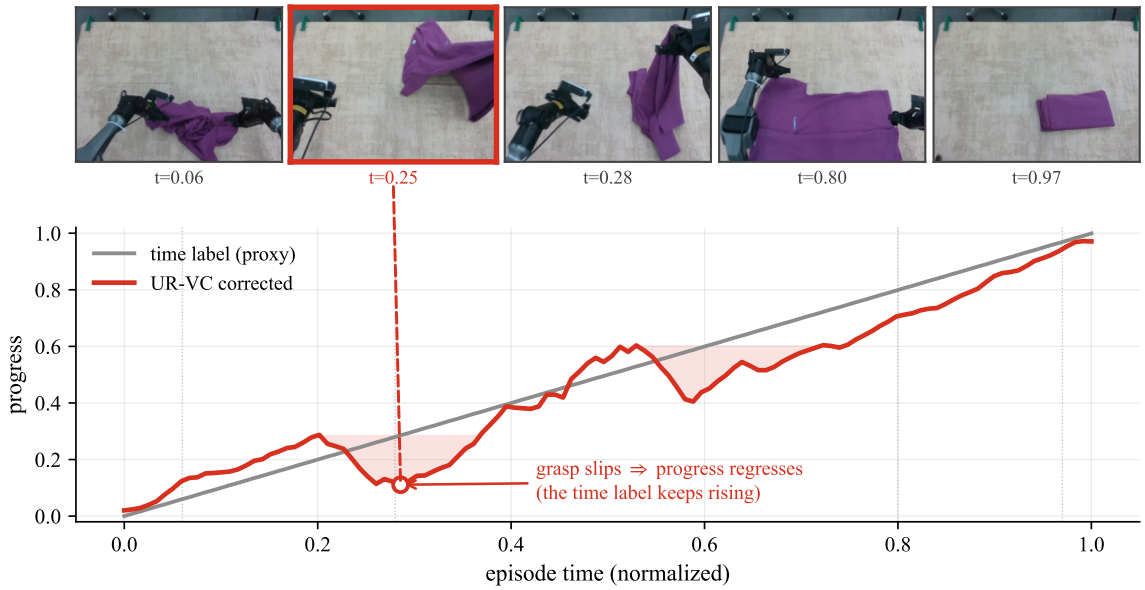

Fig. 1: **Time is not progress.** In a real cloth flatten-and-fold episode, the garment is first smoothed, but a slipping grasp reintroduces wrinkles and causes physical progress to regress. The time-derived label (gray) remains monotone by construction, whereas UR-VC estimates corrected progress by retrieving semantically similar frames from other episodes and aggregating their time-derived labels, producing a correction (red) that decreases when the visual state worsens. Across the primary evaluation set, 13.4% of frames receive a negative horizon advantage under UR-VC, capturing regressions the time proxy misses.

time-derived labels into a corrected progress estimate, with at most one matched label contributed by each episode to avoid over-weighting temporally adjacent frames.

We study UR-VC on real bimanual cloth flatten-and-fold data. Cloth manipulation is a natural testbed for this problem because progress is visually meaningful, local regressions occur frequently, and similar intermediate states recur across garments and episodes [28, 8]. Since true physical progress is not directly observed in demonstrations, we first evaluate the conditions that make correction useful: cross-episode coverage, task-state consistency, stability under aggregation, and recovery of non-monotone progress. We find that these conditions hold in our data: high-quality matches are abundant even at moderate dataset sizes (Fig. 2), retrieved frames align with folding state across different garments (Fig. 3), the corrected estimate becomes smoother as the evaluation set grows (Fig. 4), and the labels recover local regressions while preserving the overall task trend (Fig. 1). As a downstream use case, we convert the corrected signal into advantage labels for VLA training, following recent advantage-conditioned policy learning [1, 22], and observe a positive trend in real-robot task success under matched training settings (Tab. I).

Our contributions are summarized as follows:

- **Proxy-label perspective.** We formulate normalized time as a noisy proxy for latent physical task progress, and highlight its limitations both within episodes and across episodes in non-monotone manipulation.
- **Unsupervised correction.** We propose UR-VC, an offline and training-free method that corrects time-derived progress labels by aggregating semantically similar states across episodes, requiring no manual progress labels, reward annotations, or an additional value model.
- **Real-robot evidence.** On real bimanual cloth flatten-and-fold data, we validate the conditions needed for correction and demonstrate a downstream use case in advantage-conditioned VLA training.

II. Related Work

Progress and value supervision in robot learning. Vision-language-action models such as RT-2 [5], OpenVLA [15], and the $\pi_0/\pi_{0.5}$ family [4, 3] make large-scale robot policy learning practical, and recent systems consume explicit progress or value signals: $\pi^*_{0.6}$ [1] learns a value function for advantage conditioning, and Gemini Robotics 1.5 [25] uses stage progress for planning and completion judgement. A body of work learns visual progress, reward, or value estimators from contrastive video objectives, language-image representations, value pretraining, optimal transport, or generative modelling [23, 24, 18, 19, 2, 11, 13], or elicits values from large pretrained models [20, 7, 17]. UR-VC is complementary to both lines: it corrects the proxy scores *before* any estimator is trained, targeting the supervision itself.

Cross-episode redundancy and label correction. UR-VC is also related to learning with noisy labels [21, 10] and graph/nearest-neighbour label propagation [29, 14]. The difference is the source of redundancy. Rather than propagating sparse human labels over an image graph, UR-VC exploits repeated robot states across independently collected episodes, where each timestamp is a noisy measurement of the same latent physical progress.

Deformable manipulation and semantic encoders. Cloth manipulation has been studied in simulation and real-robot

settings, including SoftGym [16], ALOHA/ACT [28], Mobile ALOHA [12], and CLASP [8]. UR-VC studies real bimanual cloth flatten-and-fold data since this setting exposes both within-episode regressions and across-episode timing variation in time-derived progress labels. We use SigLIP-2 [26], an image–text pretrained semantic vision encoder, to retrieve cross-episode state matches.

III. Method

We first formalize normalized time as a noisy proxy for latent task progress and explain why its fixed-horizon difference provides a degenerate advantage signal. We then introduce UR-VC, which corrects this proxy by aggregating time-derived labels from semantically matched states across episodes. Finally, we describe how the corrected progress estimate is converted into advantage labels for policy training.

A. Time-derived progress as a noisy proxy

Let episode e contain frames $o_1^{(e)}, \dots, o_{T_e}^{(e)}$. A common time-derived proxy defines

$$g_t^{(e)} = t/T_e, \tag{1}$$

treating elapsed time as progress supervision. We assume there is also an unobserved task progress $p(o) \in [0, 1]$, determined by the physical state relevant to the task. This progress p is a conceptual target, not a quantity available to the algorithm.

The mismatch between g and p appears in two ways. First, within an episode, physical progress can be non-monotone while normalized time is monotone by construction. For instance, if a cloth fold slips or a grasp fails, $p(o_t^{(e)})$ may decrease even though $g_t^{(e)}$ continues to increase. As a result, raw time labels cannot distinguish actions that improve the task state from actions that stall or locally undo progress. Second, across episodes, the same physical state can appear at different normalized times. Operators may move at different speeds, pause for different durations, or take different recovery paths after local failures. Thus, $g_t^{(e)}$ contains not only information about task progress, but also episode-specific timing distortion. A frame that is physically close to completion may receive a lower time-derived label in a slow episode, while a less advanced state may receive a higher label in a fast episode.

These two issues directly affect downstream learning. For advantage conditioning with a fixed action horizon H, the finite difference of the raw time proxy is

$$g_{t+H} - g_t = \frac{H}{T_e}, \tag{2}$$

constant within episode e. Therefore, within the same episode, naively differencing the raw label gives every frame the same ranking signal, regardless of whether the action improves or worsens the physical state.

Across episodes the difference varies only through the episode length T_e, which primarily reflects execution speed rather than state improvement. More generally, any method that consumes raw time labels as progress or value supervision inherits these within-episode monotonicity errors and across-episode timing distortions.

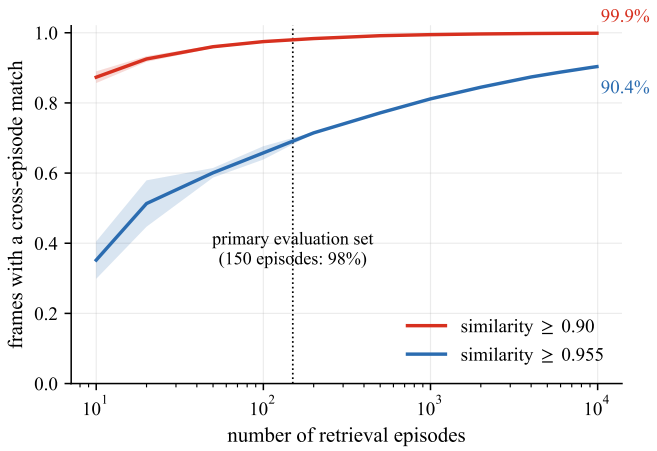

Fig. 2: **Cross-episode matches are abundant.** Fraction of task-region frames with at least one semantically similar neighbour in a *different* episode as more retrieval episodes are added. On real flatten-fold demonstrations, the 150-episode primary evaluation set already covers 98% of frames at similarity ≥ 0.90, indicating that independently collected episodes provide sufficient anchors for correcting time-derived progress labels. Coverage further improves at the 10^4-episode scale, reaching 99.9% at ≥ 0.90 and 90.4% at ≥ 0.955.

B. UR-VC: Episode-Balanced Cross-Episode Correction

UR-VC corrects a time-derived proxy by exploiting cross-episode recurrence: similar physical states often appear in independently collected demonstrations, but at different normalized times. For an idealized recurring state, episode e's time-derived score can be viewed as

$$g^{(e)} = p + \varepsilon_e, \tag{3}$$

where p is the latent progress and ε_e is an episode-specific timing error. Averaging proxy scores across independent episodes can reduces this error; averaging many nearby frames from the same episode would not, since their errors are correlated.

Concretely, let f_i be the L_2-normalized SigLIP-2 embedding of query frame i, and let $f_i^\top f_j$ denote cosine similarity. For every other episode e, we form an in-band candidate set

$$\mathcal{C}_{i,e} = \{\, j : \mathrm{ep}(j) = e,\ |g_j - g_i| \leq \tau \,\}. \tag{4}$$

We then keep only the best representative from that episode,

$$j_{i,e}^* = \arg\max_{j \in \mathcal{C}_{i,e}} f_i^\top f_j, \tag{5}$$

and discard it if its similarity is below ρ. Let $\mathcal{M}_i$ be the remaining episode representatives, optionally capped to the top m episodes by similarity. The corrected estimate is

$$\hat{g}_i = \frac{1}{|\mathcal{M}_i|} \sum_{e \in \mathcal{M}_i} g_{j_{i,e}^*}. \tag{6}$$

If no representative survives, we leave g_i unchanged.

The episode-balanced form matches the statistical assumption above: each episode contributes at most one proxy score

Fig. 3: **Cross-episode retrieval captures folding state across garment appearances.** For query frames at successive fold stages (left, black borders), nearest neighbours retrieved from *other* episodes (red borders, similarity shown) exhibit the same folding state across different garment colors.

to the average, so the estimate aggregates evidence across demonstrations rather than across temporally adjacent frames from a single trajectory. The similarity threshold ρ acts as a conservative match filter, preventing episodes without a sufficiently similar state from contributing to the estimate. The temporal band τ imposes a locality constraint on label correction: all averaged labels satisfy $|g_j - g_i| \leq \tau$, and therefore the correction magnitude is bounded as $|\hat{g}_i - g_i| \leq \tau$.

UR-VC is an offline label-correction procedure. After computing visual embeddings, it only requires similarity search, per-episode representative selection, and scalar averaging. In implementation, the per-episode argmax is computed with a single masked scatter-max over the offline corpus, so the correction reduces to a small number of matrix operations. UR-VC doesn't require manual progress labels, reward annotations, online rollouts, or training an additional value model.

C. Advantage Labels from Corrected Progress

After correcting the time-derived proxy, we use the resulting estimate $\hat{g}$ to construct advantage labels for policy training following $\pi^*_{0.6}$. This step is only one downstream use of UR-VC: the correction is performed at the progress-label level, before any policy or value model is trained.

For an action chunk starting at frame i with horizon H, we define a scalar corrected advantage proxy as

$$r_i = \hat{g}_{i+H} - \hat{g}_i. \tag{7}$$

Near the end of an episode, we compute the difference to the final frame and rescale it to the horizon rate:

$$r_i = \frac{H}{T_e - i}\left(\hat{g}_{T_e} - \hat{g}_i\right). \tag{8}$$

Here we follow the indexing convention $o_1^{(e)}, \ldots, o_{T_e}^{(e)}$. If the implementation uses zero-indexed frames, the denominator should be adjusted accordingly.

We rank frames by r_i and mark the top 20% as positive-advantage examples. The corresponding training frames receive a plain-text prompt suffix, e.g., "`..., advantage: positive`". At deployment, the policy is queried with the same positive-advantage suffix. The policy architecture, training data, and optimization schedule are otherwise unchanged.

Thus, UR-VC introduces no new policy objective and no separate value model. It only replaces the raw time-derived supervision with a corrected signal, allowing existing progress- or advantage-conditioned pipelines to use cross-episode information without additional annotation or critic training.

IV. Experiments

We evaluate UR-VC on real bimanual cloth flatten-and-fold data. Since true physical progress is not directly observed in real demonstrations, we do not claim access to ground-truth progress labels. Instead, our evaluation follows the requirements implied by the method: cross-episode state recurrences should be sufficiently dense, retrieved frames should correspond to the same semantic task state rather than superficial appearance, the corrected labels should express non-monotone progress, and the resulting signal should be usable in a downstream robot-learning pipeline.

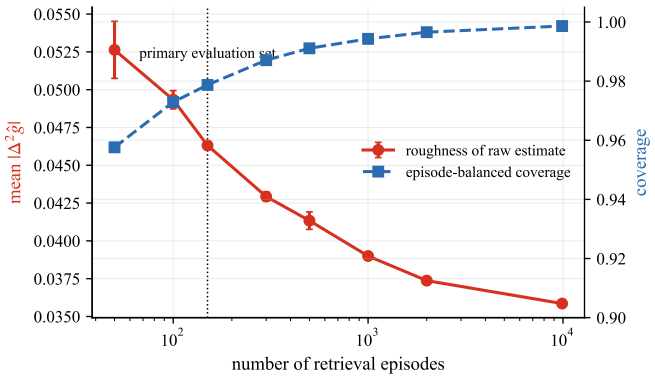

Fig. 4: **The correction improves with more retrieval episodes.** Episode-balanced UR-VC across retrieval-set sizes, reported as mean ± std over random subsets. The roughness of the corrected progress estimate, measured by mean $|\Delta^2 \hat{g}_t|$, decreases by roughly one third from 50 episodes to the 10^4-episode scale. Meanwhile, coverage remains high and rises from 96% to 99.9%, showing that larger retrieval sets provide both denser support and more stable correction. The 150-episode primary evaluation set already lies in a usable regime.

A. Experimental Setup

Task and data. We study a long-horizon bimanual cloth flatten-and-fold task. Each episode starts from a garment placed on a table and requires the robot to flatten the cloth and then fold it into the target configuration. This task is a natural setting for evaluating time-derived progress proxies: progress is visually meaningful, intermediate states recur across garments and episodes, and local regressions frequently occur due to slips, failed grasps, wrinkles, or partial undoing. Unless otherwise stated, intrinsic analyses use a primary evaluation set of 150 real robot episodes sampled from a flatten-and-fold dataset released by [27]. We additionally report scaling trends by increasing the retrieval pool up to the 10^4-episode scale.

UR-VC implementation. For each frame, we compute an L_2-normalized SigLIP-2 visual embedding and retrieve semantically similar frames from other episodes. Unless otherwise stated, we use a temporal band $\tau = 0.3$ and a cosine-similarity threshold $\rho = 0.90$, which provide a conservative locality constraint and filter weak cross-episode matches.

Metrics. We evaluate four aspects of the correction. First, *coverage* measures the fraction of query frames that have at least one valid match from a different episode. Second, *semantic consistency* checks whether retrieved frames preserve task state across appearance variation. Third, *non-monotone recovery* measures whether the corrected estimate produces negative horizon advantages in states where progress locally regresses, while preserving the overall task trend. Finally, we test a downstream use case by using UR-VC-derived advantage labels to train an advantage-conditioned VLA and evaluate it on a dual 7-DoF AgileX robot with absolute joint control

B. Cross-Episode Coverage and Scaling

UR-VC is useful only if query states have high-quality matches in other episodes. This condition already holds at the scale of the primary evaluation set. With 150 episodes, 98% of frames have a nearest neighbor in a different episode with cosine similarity at least 0.90; 69.9% remain covered under the stricter threshold 0.955.

Coverage further improves with retrieval-set size. As the retrieval pool increases from the primary set to the 10^4-episode scale corpus, coverage rises to 99.9% at threshold 0.90 and 90.4% at threshold 0.955 (Fig. 2). This suggests that cross-episode redundancy is not only present in large-scale robot datasets but is already available at moderate collection scales, and it becomes denser as more demonstrations are added.

C. Semantic Consistency of Retrieved States

High coverage alone is insufficient: visually similar frames must also correspond to similar task states. Otherwise, averaging their time-derived labels would blur together unrelated configurations rather than correct temporal distortion. We therefore inspect nearest-neighbor retrievals across different episodes and garments (Fig. 3). The retrieved frames preserve the folding state across garment colors and appearances, indicating that the embedding primarily captures cloth configuration rather than superficial texture. These results support the main assumption behind UR-VC: semantically similar task states recur across independently collected episodes, and retrieval can identify them despite appearance variation.

D. Recovering Non-Monotone Progress

We next examine whether UR-VC expresses progress patterns that normalized time cannot represent. A normalized-time label is monotone by construction, so its fixed-horizon advantage is non-negative and nearly constant within an episode. It therefore cannot distinguish an action that improves the cloth state from one that stalls or temporarily undoes progress.

UR-VC preserves the global task trend while allowing local regressions. In the real episode shown in Fig. 1, the corrected estimate decreases when a slipping grasp reintroduces wrinkles and the cloth state visibly worsens, whereas the raw time-derived label continues to increase. Across the primary evaluation set, 13.4% of frames receive a negative horizon advantage under UR-VC, using ($r_i < -0.02$ with a horizon of 5% of the episode length, approximately ≈1.7 s). At the same time, the corrected estimate remains highly correlated with normalized time overall (0.98), which is expected because the temporal band constrains the correction to remain local. Thus, UR-VC does not discard the coarse temporal ordering of the demonstration; it selectively corrects local regions where the visual state suggests that progress has regressed.

The correction also becomes more stable as the retrieval pool grows. From 50 episodes to the 10^4-episode scale, the roughness of the corrected estimate, measured by mean $|\Delta^2 \hat{g}_t|$, decreases by roughly one third, while coverage increases from 96% to 99.9 % (Fig. 4). Larger retrieval sets provide both denser support and smoother estimates.

TABLE I: **Downstream real-robot evaluation.** Conditions correspond to different tablecloth backgrounds, with 30 trials per condition. UR-VC improves the average success rate from 72.8% to 78.9% and performs better in 5 of 6 conditions.

Condition	Baseline	UR-VC
Bare table	0.90	**0.97**
Beige cloth	0.70	**0.73**
Blue-gray cloth	**0.50**	0.43
Light-yellow cloth	0.63	**0.90**
Light-gray cloth	0.77	**0.80**
Khaki cloth	0.87	**0.90**
Average	0.728	**0.789**

E. Downstream Real-Robot Evaluation

UR-VC produces a corrected progress signal independently of how that signal is consumed. We evaluate one downstream use case: advantage-conditioned VLA training in the style of $\pi^*_{0.6}$, using a $\pi_{0.5}$ backbone.

Setup. All methods share the same human-collected training set, model architecture, training schedule, and hyperparameters; one policy is trained per labelling scheme. The training mix combines 5700 flatten-and-fold demonstrations with 1795 dedicated *recovery* demonstrations that pass through degraded cloth states—precisely the regime where time-derived labels are least reliable. The no-advantage baseline trains without advantage labels. The UR-VC variant computes corrected progress with Eq. 6, marks its top-20% advantage frames as positive (Sec. III-C), and conditions the policy on the resulting positive/negative labels.

Evaluation protocol. We evaluate on an AgileX bimanual robot performing flatten-then-fold on short-sleeve garments. The test set contains six table conditions, including a bare-table condition and five tablecloth backgrounds. For each condition, we evaluate 6 garments with 5 repetitions each, yielding 30 trials per condition and 180 trials per method. We report per-condition and the average success rates in Tab. I.

Results. UR-VC advantage conditioning improves average success from 0.728 (131/180) to 0.789 (142/180), with higher success in 5 of 6 table conditions. These results provide application-level evidence that UR-VC-derived advantage labels can improve policy performance when inserted into a standard advantage-conditioned VLA pipeline. Importantly, the improvement is obtained without changing the backbone, training data, optimization schedule, or policy objective, suggesting that the gains are attributable to the supervision signal rather than additional model capacity or training changes.

V. Limitations

UR-VC assumes that visually similar observations indicate similar task progress. This assumption is shared with progress estimator and value models: when visually similar states occur at different task stages or after different action histories, their latent progress can be ambiguous. Episode-balanced averaging reduces trajectory-specific timing noise, but it cannot correct systematic retrieval errors.

Our downstream evaluation focuses on one representative use case: using UR-VC-derived supervision for advantage-conditioned VLA training in real bimanual cloth manipulation. Under matched experimental settings, the corrected labels lead to higher average success, suggesting their practical utility as a drop-in supervision signal. Future work may further examine the method across broader task families, training scales, and hyperparameter settings.

VI. Conclusion

UR-VC corrects time-derived progress proxies by averaging semantic state recurrences across independent robot episodes. It uses dataset redundancy to move scalable proxy scores toward latent physical progress without training an additional value model. On real bimanual cloth data, cross-episode matches are abundant and semantic, corrected estimates recover regressions that a monotone time proxy cannot express, and the same signal can be used for advantage-conditioned VLA training. Time is useful scalable supervision, but it should not be conflated with physical progress. Offline correction of time-derived labels offers a lightweight, model-agnostic complement to learned progress or value models.

References

[1] 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.org*, 2511.14759.

[2] Chethan Bhateja, Derek Guo, Dibya Ghosh, Anikait Singh, Manan Tomar, Quan Vuong, Yevgen Chebotar, Sergey Levine, and Aviral Kumar. Robotic offline rl from internet videos via value-function pre-training. *arXiv.org*, 2309.13041.

[3] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *CoRL*, 2025.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esber, Michael Equi, Chelsea Finn, et al. π_0: A vision-language-action flow model for general robot control. In *RSS*, 2025.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

[6] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *NeurIPS*, 2021.

[7] 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.org*, 2602.19313.

[8] Yuhong Deng, Chao Tang, Cunjun Yu, Linfeng Li, and David Hsu. Clasp: General-purpose clothes manipulation with semantic keypoints. *arXiv.org*, 2408.08160.

[9] Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline rl via supervised learning? In *ICLR*, 2022.

[10] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2013.

[11] Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Robot policy learning with temporal optimal transport reward. In *NeurIPS*, 2024.

[12] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *CoRL*, 2024.

[13] Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. In *ECCV*, 2024.

[14] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, 2019.

[15] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *CoRL*, 2024.

[16] Xingyu Lin, Yufei Wang, Jake Olkin, and David Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *CoRL*, 2021.

[17] 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.org*, 2604.08168.

[18] Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *ICML*, 2023.

[19] 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. In *ICLR*, 2023.

[20] 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 *ICLR*, 2025.

[21] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, 2013.

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

[23] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018.

[24] Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. In *NeurIPS*, 2023.

[25] Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv.org*, 2510.03342.

[26] 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.org*, 2502.14786.

[27] Checheng Yu, Chonghao Sima, Gangcheng Jiang, Hai Zhang, Haoguang Mai, Hongyang Li, Huijie Wang, Jin Chen, Kaiyang Wu, Li Chen, et al. χ_0: Resource-aware robust manipulation via taming distributional inconsistencies. *arXiv.org*, 2602.09021.

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

[29] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.