Phase-and-First-Arrival VLM Feedback for Sparse-Reward Surgical Manipulation

Anonymous Authors

***Abstract*— Sparse outcome feedback limits what robots can learn from unsuccessful attempts at complex manipulation. Failed multi-stage surgical attempts can contain grasps, lifts, or transfers worth reusing. Terminal rewards collapse such attempts to the same outcome, while scalar vision–language model (VLM) ratings reveal neither what progress merits credit nor when it occurred. We introduce phase-and-first-arrival feedback: one VLM query per recorded episode identifies the furthest visually verified task phase and when that phase is first reached, allowing the learner to reuse partial behavior and localize credit. We instantiate it in SurgPhaseBench, a phase-structured suite spanning rigid and deformable tasks, and evaluate it in simulation and hardware. Across five simulated tasks, our method reaches 75.2% mean success, compared with 52.1% for a reward based on Contrastive Language–Image Pre-training (CLIP) using the same visual input; the advantage persists when only the feedback representation changes. On hardware, the same record supports autonomous block picking and slip recovery. Together, these results show that trajectory-level visual supervision can preserve partial progress while providing the temporal credit needed for sparse-reward control.**

Project website—https://surgphase.verloge.space

Index Terms—Surgical Robotics: Laparoscopy, Reinforcement Learning, Vision–Language Models.

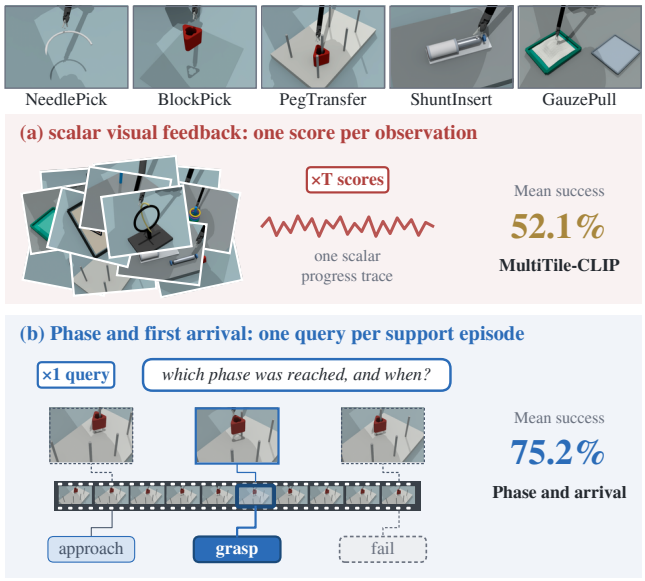

Fig. 1. One structured query bridges episode- and transition-level supervision. A confidence-and-progress gate retains partial attempts, rank scales their credit, and first arrival localizes reuse. After 120k policy updates, phase-conditioned replay reaches 75.2% mean success versus 52.1% for a CLIP scalar reward using the same views.

I. Introduction

Surgical autonomy requires reliable policies for precise, contact-rich manipulation across ordered stages [1], [2]. Reinforcement learning offers one route, but a terminal success reward gives every unsuccessful attempt the same return [3], [1]. A needle-picking attempt that grasps and lifts the needle before dropping it earns no more credit than one that never makes contact, yet contains behavior worth reusing. Dense shaping recovers that signal, but it is authored per task and reads privileged simulator state that hardware does not expose, so the effort does not carry to a new procedure or to a real manipulator. The challenge is to preserve partial progress without task-specific dense rewards.

Vision–language models supply feedback that needs no task-specific predicate: they judge progress from images and a short task description [4], [5], [6], [7]. The forms in use, however, trade economy against resolution. A scalar trajectory rating costs one query per episode but reports only how good the attempt was, not which event earned that score or when it happened. Pairwise preferences rank two attempts without locating progress inside either. Framewise scoring does localize, but only among the observations it scores, so finer temporal resolution is bought with proportionally more queries. Reinforcement learning needs the opposite of a summary: credit attached to particular transitions. Can one query per episode retain the economy of trajectory-level feedback while still locating reusable behavior in time?

We address this with phase-and-first-arrival feedback over ordered surgical phases. Peg Transfer progresses through grasps, handoffs, and placements [8], and surgical workflow models label procedural video by phase [9], [10]. In our tasks, these phases are visually distinguishable, so they can be verified without state predicates; their order gives even an unsuccessful episode measurable partial progress. In Figure 1, the VLM reads a completed-episode storyboard and returns the furthest verified phase and the interval in which it is first reached: phase identifies reusable progress, and arrival locates it in time.

A fixed compiler turns each record into a learning signal. A confidence-and-progress gate rejects uncertain or no-progress records; phase rank scales retained credit; and first arrival localizes it to an event-centered window for replay, critic targets, and behavior cloning. The VLM processes a fixed pool of recorded support episodes once before policy learning, producing a frozen record bank for mixing with online experience. Collection and evaluation continue on the sparse environment reward.

We instantiate the method in SurgPhaseBench, twelve

phase-structured ManiSkill3 environments spanning rigid and deformable interaction: five evaluate the complete observer–compiler–learner loop, and seven establish executable and semantic breadth. Across the five policy-learning tasks, phase-and-first-arrival feedback outperforms matched scalar feedback. The same frozen-record interface supports autonomous block picking and recovery from slips on physical hardware, with no VLM query during policy deployment.

Our contributions are:

1) We introduce phase-and-first-arrival feedback, a VLM-based method that converts one storyboard query per support episode into a structured progress-and-timing record, then compiles that record into localized credit for reusing partial behavior from unsuccessful attempts.
2) We provide SurgPhaseBench, a twelve-environment benchmark for phase-structured, sparse-reward surgical manipulation across rigid and deformable interactions, with a five-task track for end-to-end policy learning.
3) We show that phase-and-first-arrival feedback improves five-task mean success by 7.4 percentage points over scalar feedback from the same VLM. On a physical robot, the learned block-picking policy succeeds in 18 of 20 held-out trials without deployment-time VLM queries.

II. Related Work

Surgical platforms and benchmarks: Open surgical learning platforms include dVRL, SurRoL, LapGym, and Orbit-Surgical [11], [12], [13], [14]. They provide subtasks and support demonstration-constrained RL, sim-to-real manipulation, autonomous suction, and multi-task automation [3], [15], [16], [2]. SurgPhaseBench addresses a complementary evaluation need: each environment couples sparse success with ordered visual milestones and first-arrival semantics, making partial progress a common observable across tasks.

Phases and task structure: Bendikas et al. [1] define exploratory bottlenecks and train nested NeedlePickAndPlace subtasks. Their decomposition changes the training problem into a sequence of subtasks. Our pipeline preserves the original episode and stores its furthest phase and first-arrival interval as replay metadata. Workflow recognition predicts framewise phase sequences for video understanding [9], [10]. Our observer instead returns one maximal phase and arrival interval for replay and policy learning.

Visual supervision for control: Visual models can score observations or trajectory videos [4], [17], [18]. SuccessVQA locates the first frame after which success persists and clones accepted behavior [5]; RL-VLM-F and ERL-VLM collect online preferences or ratings, while PLARE learns from fixed VLM preferences without a reward model [6], [19], [7]. These produce success, scalar, or preference supervision. VICtoR, REDS, and SARM instead learn dense or stage-aware rewards from demonstrations or labels [20], [21], [22]. Our pipeline uses one query on each fixed support episode to jointly identify partial progress and its first arrival, then compiles the records before policy learning.

Structured credit and replay: Reward machines require event propositions evaluable along a trajectory [23]. HER relabels transitions with state-accessible achieved goals [24]; SurRoL's HER+DEMO applies Q-filtered cloning to demonstrations [12], [25]. Potential shaping redistributes a specified reward [26]. Our post-hoc visual record instead turns one verified partial phase and its arrival into a transition distribution, localized critic targets, and Q-filtered reuse while retaining the original task goal and sparse environment reward.

III. Methodology

A. Problem Formulation

We consider finite-horizon continuous-control episodes with sparse environment reward r_t and terminal bit b_t. A Soft Actor-Critic (SAC) policy $\pi_\theta(a \mid o)$ [27] acts from state observation o_t, and synchronized RGB v_t provides the post-hoc visual record. We target settings where unsuccessful episodes share one return, so grasping and lifting before a drop is indistinguishable from no contact.

The input is a support pool frozen before learning, with each transition sequence paired to synchronized RGB. From episode-level feedback, we recover *which* unsuccessful episodes contain reusable behavior and *where* it occurs. Figure 2 separates the resulting stages: one frozen-VLM query and deterministic compilation per support episode, then mixed-replay SAC. Online collection and evaluation use r_t; only bank samples receive auxiliary credit.

B. Episode-Level Phase Observation

For task $\mathcal{T}$, the author supplies description $d_{\mathcal{T}}$, a cue-annotated ordered vocabulary $\mathcal{Q}_{\mathcal{T}} = (q_0 \prec \cdots \prec q_{K_{\mathcal{T}}})$, and a nondecreasing map $\rho_{\mathcal{T}} : \mathcal{Q}_{\mathcal{T}} \to \{0, \ldots, K\}$ with $\rho_{\mathcal{T}}(q_0) = 0$. Task-specific labels map to shared progress ranks; authors specify their observable semantics and order.

The five simulated task orders are BlockPick/NeedlePick: *clear failure, approach, contact or grasp, lift, stable success*; PegTransfer: *clear failure, approach, contact or grasp, lift, transfer, stable success*; ShuntInsert: *insert failure, approach, grasp, align, seat*; and GauzePull: *clear failure, approach, attach, pull, target align, target reach*. Their semantics map monotonically to the shared ranks *none, approach, grasp, lift, at-goal, success*; inapplicable ranks are skipped.

The fixed support pool $\mathcal{D}_{\text{sup}} = \{\tau_i\}_{i=1}^{N}$ contains episodes τ_i of length T_i, each pairing a transition sequence with synchronized RGB, and may include failed attempts with reusable partial behavior. A fixed adapter $A_{\mathcal{T}}$ turns the episode video into a timestamped storyboard with full views and task-specified crops. The phase vocabulary emphasizes visually verifiable milestone states, such as an object being lifted, transferred, aligned, or placed. By combining chronological context with full views and task-relevant crops, the storyboard lets the observer identify the furthest state evidenced at any time in the episode, while adjacent timestamps delimit a first-arrival bracket sufficient for localized credit assignment. When evidence supports two adjacent labels, the observer conservatively selects the lower phase. The frozen observer

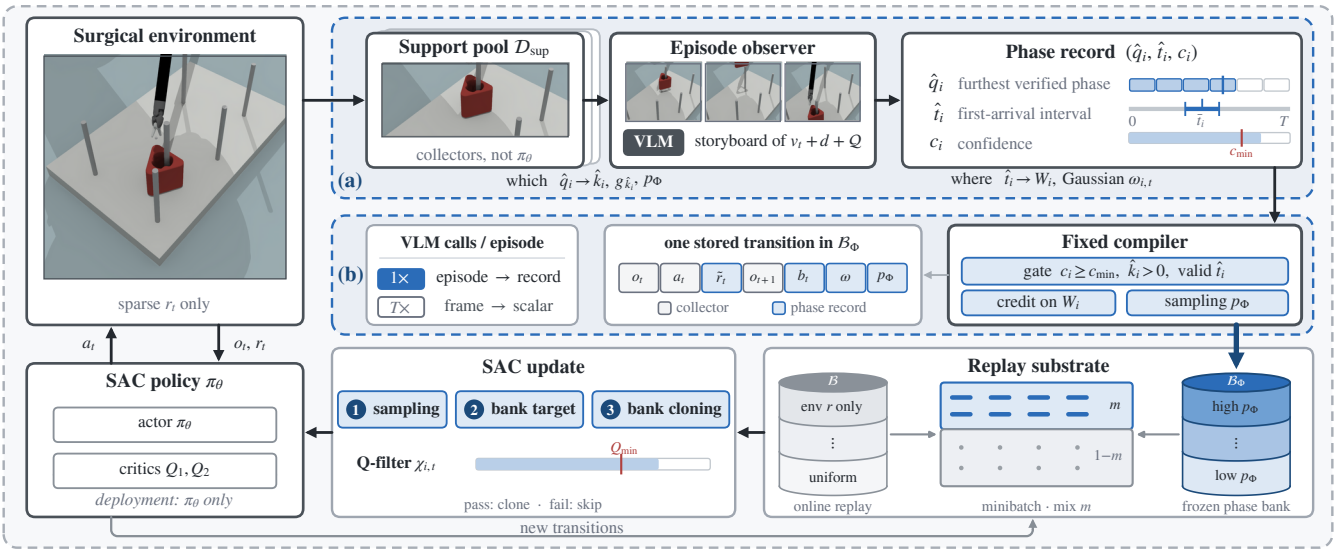

Fig. 2. From a completed attempt to reusable replay. The surrounding loop collects and evaluates on the sparse environment reward, while compiled support episodes form a frozen bank. (a) One query per support episode returns a phase record. (b) A fixed compiler produces bank rewards and sampling/BC weights; the online learner applies the Q-filter. The bank is mixed into each minibatch at rate m.

is queried once:

$$(\hat{q}_i, \hat{t}_i, c_i) = f_\phi(A_\mathcal{T}(\tau_i), d_\mathcal{T}, \mathcal{Q}_\mathcal{T}), \qquad \hat{k}_i = \rho_\mathcal{T}(\hat{q}_i),$$

The observer reads the storyboard and task specification. Here $\hat{q}_i$ is the furthest verified phase, $\hat{t}_i \in \{[\hat{t}_{i,\ell}, \hat{t}_{i,u}], \varnothing\}$ its first-arrival bracket, and $c_i \in [0,1]$ its self-reported confidence for gating. If arrival falls between displayed times, the observer returns their step interval; it returns $\varnothing$ when timing is not identifiable, and q_0 always carries $\varnothing$. Maximal phase identifies partial progress; first arrival excludes repeats.

C. Compiling Phase Records into Replay

As illustrated in Figure 3, the gate accepts a record if $c_i \geq c_{\min}$, $\hat{k}_i > 0$, and $0 \leq \hat{t}_{i,\ell} \leq \hat{t}_{i,u} < T_i$. The compiler uses midpoint $\bar{t}_i = (\hat{t}_{i,\ell} + \hat{t}_{i,u})/2$ and clipped window $W_i = \{t : 0 \leq t < T_i, |t - \bar{t}_i| \leq h\}$, where $h \geq \frac{1}{2}$ ensures W_i is nonempty. Let $0 = g_0 \leq \cdots \leq g_K$ be the shared total-credit schedule. Then

$$\tilde{r}_{i,t} = r_{i,t} + \frac{g_{\hat{k}_i}}{|W_i|} \mathbb{1}[t \in W_i]. \tag{1}$$

Thus $\sum_t (\tilde{r}_{i,t} - r_{i,t}) = g_{\hat{k}_i}$: rank sets total credit and the box window conserves it while distributing it near estimated arrival. The auxiliary reward enters bank Bellman targets, while collection and evaluation retain the environment reward. Task-specific labels enter the shared compiler through $\rho_\mathcal{T}$.

The arrival midpoint also defines

$$\omega_{i,t} = \exp\left[-\frac{(t - \bar{t}_i)^2}{2\sigma^2}\right], \qquad \sigma > 0, \tag{2}$$

which provides a smooth arrival-proximity weight for sampling and behavior cloning. The box kernel fixes total credit; the Gaussian emphasizes actions near the midpoint.

Each accepted transition enters $\mathcal{B}_\Phi$ with training reward $\tilde{r}_{i,t}$, arrival weight $\omega_{i,t}$, and fixed sampling distribution

$$p_\Phi(i,t) \propto 1 + \eta_r(\tilde{r}_{i,t} - r_{i,t}) + \eta_a \omega_{i,t}. \tag{3}$$

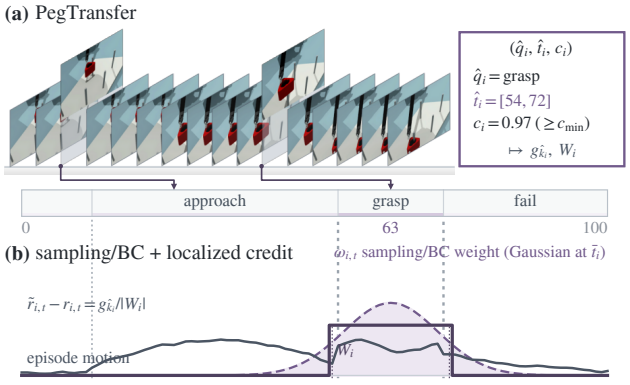

Fig. 3. Phase compilation on one recorded attempt. (a) The observer maps a completed simulated peg-transfer episode to furthest phase $\hat{q}$, first-arrival interval $\hat{t}$, and confidence c; $\rho_\mathcal{T}$ maps the label to shared rank $\hat{k}$. (b) The gate retains verified progress, rank sets total bank credit, and the arrival midpoint defines a credit window and Gaussian sampling/BC weights (Equations (1) and (2)).

The distribution is normalized over all transitions in accepted records. Here $\eta_r, \eta_a \geq 0$; the unit floor fixes the common priority scale. The resulting distribution retains episode context, emphasizes auxiliary credit and arrival proximity, and remains fixed throughout training.

D. Online Learning with the Frozen Phase Bank

Sampling and critic update: Let p_{on} be uniform over $\mathcal{B}_{\text{on}}$. A minibatch draws fraction $m \in [0,1]$ from the frozen bank, equivalently $p_{\text{mix}} = m p_\Phi + (1-m) p_{\text{on}}$. We use twin-critic SAC [27], with $\tilde{r}$ on bank transitions and r online.

Actor update: Let $a_\theta^{\text{eval}}(o)$ be the deterministic action used at evaluation and $Q_{\min,\psi} = \min_j Q_{\psi_j}$. Following demonstration-augmented RL [25], the Q-filter sets $\chi_{i,t} = 1$ when bank action a_t has no lower $Q_{\min,\psi}$ at o_t than $a_\theta^{\text{eval}}(o_t)$,

and zero otherwise. Let $\mathcal{I}_\Phi$ be the bank-origin samples in the current mixed minibatch and $\ell_{i,t} = d_a^{-1}\|a_\theta^{\text{eval}}(o_t) - a_t\|_2^2$, the mean squared action deviation. Their weighted cloning loss is

$$\mathcal{L}_{\text{BC}} = \frac{\sum_{(i,t)\in\mathcal{I}_\Phi} \chi_{i,t}\omega_{i,t}\ell_{i,t}}{\sum_{(i,t)\in\mathcal{I}_\Phi} \chi_{i,t}\omega_{i,t}}, \tag{4}$$

We skip the BC term when the denominator is zero. Here d_a is the action dimension. Because ω enters p_Φ, arrival proximity controls both bank sampling and cloning strength.

Before online learning, the actor minimizes Equation (4) alone with $\chi \equiv 1$. Online, let $\mathcal{L}_{\text{SAC}}^{\text{actor}}$ denote the standard entropy-regularized SAC actor loss evaluated on the mixed minibatch. We optimize

$$\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{SAC}}^{\text{actor}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}}, \tag{5}$$

where $\lambda_{\text{BC}} \geq 0$ weights bank cloning. Frozen sampling, bank critic targets, and Q-filtered cloning integrate the phase record into SAC. Algorithm 1 summarizes the procedure.

Algorithm 1 Fixed phase-bank construction and online SAC

Require: $\mathcal{D}_{\text{sup}}$, $(d_\mathcal{T}, \mathcal{Q}_\mathcal{T}, \rho_\mathcal{T})$, adapter $A_\mathcal{T}$, observer f_ϕ, method settings
Initialize $\mathcal{B}_\Phi$, $\mathcal{B}_{\text{on}}$, and the SAC actor/critics
for each support episode $\tau_i \in \mathcal{D}_{\text{sup}}$ **do**
  $(\hat{q}_i, \hat{t}_i, c_i) \leftarrow f_\phi(A_\mathcal{T}(\tau_i), d_\mathcal{T}, \mathcal{Q}_\mathcal{T})$ ▷ one query
  $\hat{k}_i \leftarrow \rho_\mathcal{T}(\hat{q}_i)$
  if $c_i \geq c_{\min}$, $\hat{k}_i > 0$, and $\hat{t}_i$ is valid **then**
    Compile $\tilde{r}$, ω, and p_Φ by Equations (1) to (3)
    Insert $(o, a, \tilde{r}, o', b, \omega, p_\Phi)$ into $\mathcal{B}_\Phi$
  end if
end for
Freeze $\mathcal{B}_\Phi$; warm-start by weighted BC with $\chi \equiv 1$
for each online iteration **do**
  Collect with π_θ into $\mathcal{B}_{\text{on}}$
  Draw a mixed minibatch with bank fraction m and distribution p_Φ
  Update standard SAC critics using $\tilde{r}$ on bank and r online
  Update the actor by Equation (5)
end for

IV. SurgPhaseBench

SurgPhaseBench makes partial progress an explicit evaluation object for sparse-reward surgical manipulation. Each environment couples an executable terminal objective with task-authored visual milestones, so an episode can be evaluated not only by whether it succeeds but also by which milestone it reaches and when that evidence first appears. Built on ManiSkill3 [28], its twelve environments span rigid and Warp-based deformable interaction [29].

A. Phase-Structured Task Interface

The benchmark separates task semantics from evaluation structure. Each task supplies a short description, an ordered phase vocabulary, and one-line visual decision rules. Task-specific labels preserve distinctions such as grasp–lift–transfer, align–seat, and attach–pull, while their ordinal mapping lets the same observer and compiler operate across tasks. All environments use the MSR patient-side arm, delta end-effector position control, state observations, and terminal sparse success. The five policy tasks use horizons of 75, 75, 90, 90, and 220 steps in task order. BlockPick, NeedlePick, NeedleReach, and ShuntInsert include the same proximity grasp assist for every compared method; GauzePull instead uses a Warp soft-body attachment model.

B. Task Suite and Benchmark Scope

The twelve tasks span reaching, grasp-and-lift, transfer, insertion, threading, reorientation, stacking, and deformable pulling (Figure 4). BlockPick, NeedlePick, PegTransfer, ShuntInsert, and GauzePull form the contact-rich policy-learning track: four cover rigid-object interaction, while GauzePull places deformable attachment and pulling under the same phase contract. The remaining seven environments establish executable and semantic breadth rather than entering the policy aggregate. Accordingly, the task atlas documents suite coverage, the five-task track supplies policy-learning evidence, and physical studies test external validity separately.

V. Experiments

We test whether the visual interface recovers phase and arrival, whether its records improve finite-budget policy learning, and how support quality and temporal localization affect that gain. Simulation controls are followed by observer transfer and closed-loop learning on hardware.

A. Simulation Setup

Evaluation protocol: All simulation studies run on the five-task policy-learning track. Support is collected before the downstream learner and then frozen. We study rollouts from a checkpoint trained with privileged dense shaping (*shaped*), a checkpoint trained only with terminal sparse reward (*sparse*), and untrained random exploration (*random*). Privileged shaping is therefore confined to acquisition of the shaped support pool; the observer sees only its storyboards, and the downstream learner and evaluation retain the common observation/action interface and sparse environment reward. Within each comparison, methods receive identical support rollouts, learner settings, and evaluation budgets.

The evidence is organized into three independently trained protocols: the primary feedback benchmark, 120k-update diagnostics for compiler components and support source, and two 500k-transition matched controls. The primary and diagnostic studies share the 120k SAC configuration, but their absolute values are compared only within the corresponding figure or table. Because horizons and collection parallelism differ, all methods in the primary comparison are read at pre-specified task-local points—250k, 375k, 275k, 300k, and 175k environment transitions, respectively—and averaged without weighting. Every condition uses eight seeds; task values are seed means and curves show 95% CIs.

Baselines and controls: We compare three learned-feedback families that use only image observations and a task description. MultiTile-CLIP applies CLIP-RM [30], [4] to our seven-tile views; BT-Pref [31], [6] compares pairs; and Likert-RM [7] rates an attempt with a scalar. We also report no-

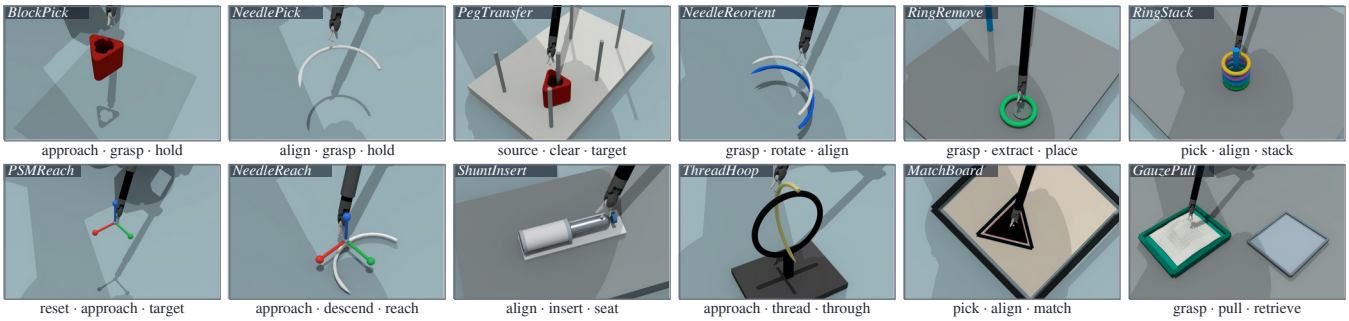

Fig. 4. One phase-structured contract spans twelve executable surgical environments. Five contact-rich tasks test the complete observer–compiler–learner loop; seven additional tasks establish execution and interface breadth.

feedback SAC, single-view CLIP-RM, and privileged Oracle-Support with ground-truth phase labels for the same pool.

Table I reports two independent 500k-transition controls. In panel (a), GPT-5.4 Scalar and Phase+Arrival share the model, seven-image input, shaped support, learner, evaluation, and annotation budget (960 requests; 134k visual tokens); only the output schema changes. Panel (b) gives MultiTile-CLIP the same phase-filtered BC, $\pm$10-step credit window, and phase-aware SAC integration used by Phase+Arrival, testing whether a matched learning stack closes the gap. This second study applies the same task-local BC/gate refinement to both ShuntInsert rows. Compare only within panels.

Implementation details: The fixed adapter forms seven panels per episode: six chronological composites on an even temporal grid and one start/final motion-difference panel. Each timed composite contains the full view, fixed task-specified crops, and its environment-step label. The observer is GPT-5.4 at temperature zero with a 400-token cap and no task-specific fine-tuning. All tasks use gate $c_{\min}$=0.70, total-credit schedule $g = (0, 0, 0.20, 0.55, 0.55, 0.90)$, Gaussian width σ=8, a $\pm$10-step arrival window, and priority coefficients $(\eta_r, \eta_a) = (4, 1)$.

The actor is warm-started for 5k updates; online learning uses bank mix m=0.6 and BC weight λ_{BC}=0.6. SAC uses entropy autotuning, γ=0.80 (0.96 for GauzePull), a 200k replay buffer, batch size 256, AdamW at 3×10^{-4}, and target rate 0.01. Each seed uses an RTX 5080 Laptop GPU and 8–32 parallel environments, set by task horizon and cost.

B. Simulation Results

Figure 5 compares seven feedback conditions on the five-task track. At the synchronized reporting points, Phase+Arrival averages 75.2% success against 52.1% for matched-view MultiTile-CLIP, 35.0% for BT-Pref, and 27.9% for Likert-RM, leading on four tasks. NeedlePick is the exception (72.5% versus 75.6%), although the phase-conditioned method later attains the highest learned-feedback peak on every task; we report synchronized points throughout. The same figure adds the wider family: sparse reward alone reaches 12.3%, standard CLIP-RM 44.0%, and privileged Oracle-Support 88.6%, leaving a 13.4-point support-observation gap.

In Table I(a), changing only GPT-5.4's output from a scalar to phase and first arrival raises average success from 76.8% to 84.2%. In panel (b), Phase+Arrival remains ahead after MultiTile-CLIP receives the matched learning stack (84.8% versus 76.0%); the panels are independent.

For the five-task shaped pool, 960 seven-image requests cost about $7.2. Using the per-task episode counts in Figure 6 and the stated horizons, full-horizon scoring would process 88,800 images instead of 6,720 (13.2$\times$ more).

Observer reliability: We compare observer outputs with frozen simulator-derived maximal-phase and first-arrival references. *Valid* denotes a parseable structured record, while

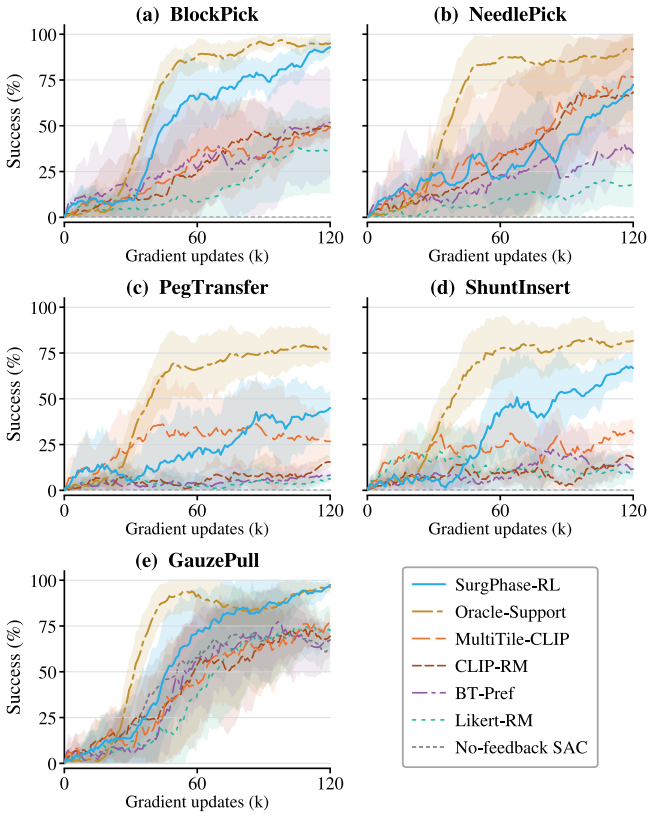

Fig. 5. Primary five-task feedback comparison after 120k SAC updates (eight seeds; shading, 95% CIs). At synchronized task-specific reporting points, Phase+Arrival averages 75.2% versus 52.1% for MultiTile-CLIP using the same views and leads on four tasks. Sparse reward reaches 12.3%; Oracle-Support reaches 88.6% using ground-truth phases on the same identically compiled pool, providing the pool's performance ceiling.

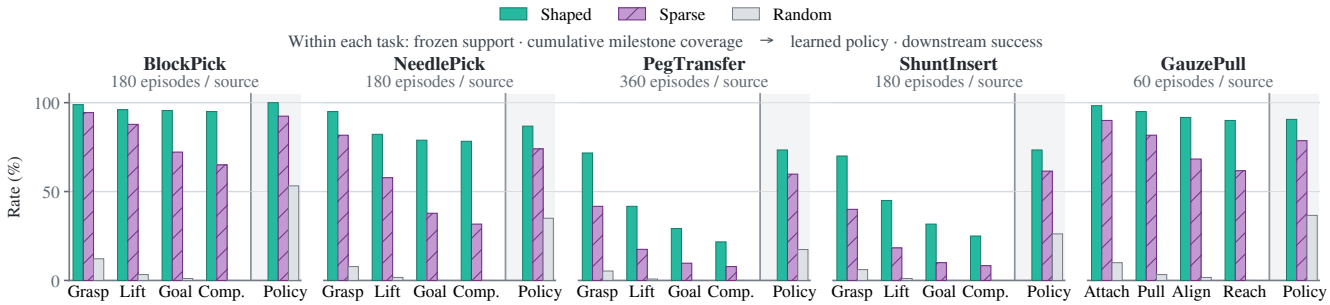

Fig. 6. Independent support-source diagnostic. For each task, the first four groups report cumulative milestone coverage in shaped, sparse, and random support, while the separated Policy group reports success after 120k updates (eight seeds per condition). Each source contains 960 episodes across tasks, giving 2,880 observer queries and 120 policy runs. PegTransfer exposes the partial-progress regime: only 21.7% of shaped episodes complete, yet the learned policy reaches 73.4% success.

gated denotes a valid record retained by the confidence, progress, and timing gate in Section III-C. Across 960 episodes, valid/gated rates are 98.6–100.0%/87.8–96.7% and within-one agreement is 73.3–91.1%; exact agreement is 33.3–58.3%. BlockPick is weakest within one phase, while ShuntInsert is hardest exactly because seating is largely hidden. First-arrival MAE against the reference step is 8.4–24.8 steps; the compiler distributes credit within ± 10 steps of the predicted arrival rather than requiring an exact event frame.

TABLE II

NEEDLEPICK MECHANISM STUDY AT 120K UPDATES (EIGHT-SEED MEAN SUCCESS, IN %). (A) CUMULATIVE CONSTRUCTION; (B) INDEPENDENTLY TRAINED ALTERNATIVES. ALL ROWS SHARE SUPPORT POOLS AND EVALUATION; THE FULL COMPILER IS REPEATED AS A REFERENCE.

(a) Cumulative build-up			*(b) Alternative mechanisms*		
Method	Shaped	Sparse	Method	Shaped	Sparse
Uniform-pool BC	30.2	26.4	Random SAC	5.0	5.0
Phase-weighted BC	50.4	42.0	Reward only	12.6	9.8
Online SAC + BC	62.8	54.4	Direct phase reward	56.2	51.8
			HER	56.8	52.6
Full compiler	**86.8**	**74.0**	**Full compiler** (ref.)	**86.8**	**74.0**

TABLE I

MATCHED CONTROLS FOR FEEDBACK REPRESENTATION AND LEARNING-STACK INTEGRATION (500K TRANSITIONS PER TASK; MEAN SUCCESS OVER EIGHT SEEDS, IN %). P+A: PHASE+FIRST ARRIVAL; Δ: GAIN OVER SCALAR.

	(a) Output format isolated			*(b) Learning stack matched*		
Task	Scalar	P+A	Δ	Scalar	P+A	Δ
BlockPick	90.0	**93.8**	+3.8	88.0	**93.8**	+5.8
NeedlePick	82.0	**85.4**	+3.4	78.0	**85.4**	+7.4
PegTransfer	68.0	**73.4**	+5.4	66.0	**73.4**	+7.4
ShuntInsert	58.0	**78.0**	+20.0	61.0	**81.0**	+20.0
GauzePull	86.0	**90.6**	+4.6	87.0	**90.6**	+3.6
Average	**76.8**	**84.2**	**+7.4**	**76.0**	**84.8**	**+8.8**

C. Ablation Studies

Each source uses the same 960-episode allocation (180/180/360/180/60 in task order) and 960 queries. Shaped and sparse collectors each use about 500k aggregate transitions per task over eight seeds; random support uses no trained collector. In this independent 120k-update diagnostic, the pools average 84.8%, 73.3%, and 33.7% success (Figure 6); these are not the primary results in Figure 5. PegTransfer illustrates the partial-progress regime: only 21.7% of shaped rollouts finish, while 71.7%/41.7%/29.2% reach grasp/lift/at-goal and support 73.4% downstream success. Random support has no successes and $\leq 1.7\%$ at-goal coverage.

ShuntInsert provides a complementary case: only 25.0% of shaped episodes complete, while 70.0%/45.0%/31.7% reach grasp/lift/at-goal. Its 20.0-point improvement in both matched-control panels is also the largest in Table I. Together, these tasks show that terminal collector success is not a sufficient support statistic: a pool may be weak at completion yet rich in transferable early- and mid-stage behavior.

Table II separates behavior selection from temporal credit. Uniform-pool BC clones all support; Phase-weighted BC gates and weights it; Online SAC+BC adds sparse-reward SAC; and Full compiler additionally supplies record-conditioned credit, weights, and priorities. Phase weighting contributes 20.2/15.6 points on shaped/sparse support, and full compilation adds 24.0/19.6 over Online SAC+BC. Reward only starts without BC, Direct phase reward omits localization and record-derived weights, and HER relabels target positions after Uniform-pool BC without a phase bank. Direct phase reward loses 30.6/22.2 points; HER is similar, but only phase records expose event time.

Together, source and mechanism studies expose complementary limits. Coverage determines which milestones enter the bank; compilation selects and credits them at first arrival. Phase magnitude alone cannot recover the full gain, and localization cannot recover a phase absent from support.

D. Physical Robot Experiments

Physical experiments use a custom patient-side manipulator: a 6-DoF UR5 carries a da Vinci Xi EndoWrist Mega Needle Driver through a 5-DoF adapter (Figure 7). A fixed oblique

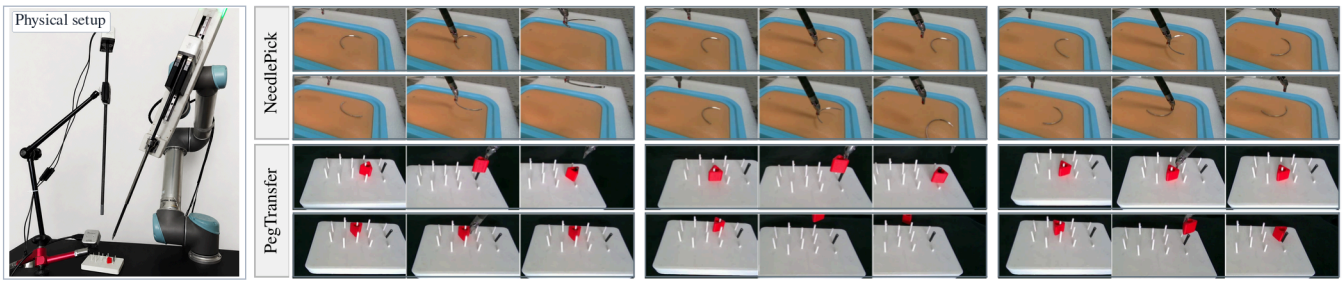

Fig. 7. Physical experimental setup and observer examples. Left: a UR5 carries a da Vinci Xi EndoWrist instrument through a custom adapter and is viewed by a fixed oblique RealSense D405. Right: success, partial, and failure examples from teleoperated physical needle-picking and peg-transfer videos; observer agreement across all three physical tasks is reported in the text.

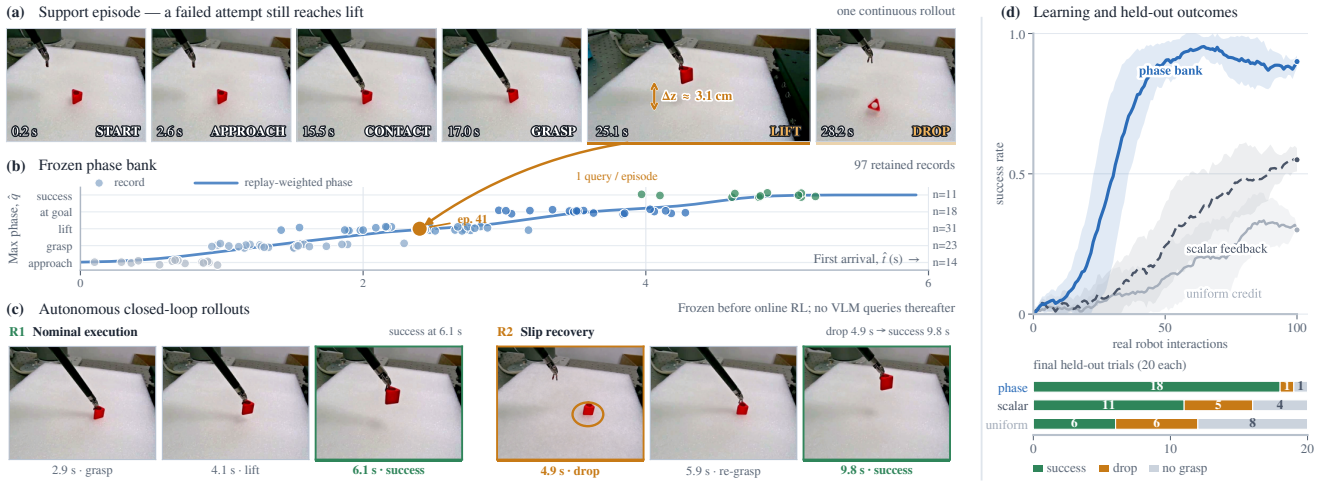

Fig. 8. Physical robot block-picking closed-loop experiment. (a) A failed support episode reaches lift before dropping the block. (b) One offline query per episode produces a 97-record phase bank, frozen before online learning. (c) Representative autonomous rollouts, including slip recovery. (d) After 100 online transitions, phase-conditioned replay achieves 18/20 held-out successes, versus 11/20 for framewise scalar feedback and 6/20 for uniform credit; deployment uses no VLM queries.

RealSense D405 provides the online RGB-D target pose and synchronized diagnostic video (848×480, ≈ 22 fps). The observer returns per-phase arrival times in seconds; the maximal phase and its timestamp form a degenerate bracket for the same downstream record used in simulation.

We test observer transfer from simulation to teleoperated physical videos of block picking, needle picking, and peg transfer. Block picking evaluates autonomous closed-loop learning in a 20-trial study. A human labels maximal phase and outcome before the VLM scores 15 videos per task (five success, five partial, five failure). All 45 outputs are valid; exact and within-one agreement reach 71.1% and 91.1%, and 41/45 success labels match human labels. Peg transfer is hardest because its phases look similar from one view.

In the physical block-picking study, the policy maps target pose and joint state to $a_t = (\Delta x, \Delta y, \Delta z, g)$ (Figure 8). One query per episode yields a frozen bank of 97 records. Each condition receives the same support and 100 online transitions, followed by 20 held-out trials. Framewise feedback scores each frame without phase or arrival; uniform credit preserves rank but spreads credit and replay weight across the episode.

Phase replay succeeds in 18/20 trials, versus 11/20 for scalar feedback and 6/20 for uniform credit. A retained failure lifts 3.1 cm, and an online rollout recovers from a slip. Under the same 100-transition budget, phase replay leaves one drop and one no-grasp, versus five/four for scalar feedback and six/eight for uniform credit. The block-picking study provides end-to-end evidence; the needle-picking and peg-transfer video audits test observer transfer across tasks.

VI. Discussion

The matched controls and ablations clarify what the structured record contributes. Holding the VLM, storyboard, support, and learning stack fixed, phase-and-arrival feedback remains more effective than a scalar summary, so the gain is not explained by additional visual evidence or an unmatched learner. The support-source and mechanism studies then separate two roles: milestone coverage determines which partial behaviors are available for reuse, while first arrival specifies the transition neighborhood where their credit should concentrate. A more precise observer cannot supply behavior missing from support, and broad support alone does not determine where its credit should land. This separation localizes failures: missing phases call for new support, whereas uncertain timing calls for calibration.

Freezing records before learning makes the interface in-

spectable: each episode can be traced to its phase, confidence, and arrival bracket, and the same bank can be recompiled without querying the VLM during deployment. The physical studies separate observer transfer across tasks from end-to-end closed-loop learning in physical block picking. More broadly, the approach is most natural for procedures with ordered, visually distinguishable milestones. Support-aware acquisition and calibrated uncertainty offer routes to ambiguous phases, longer horizons, and deformable physical tasks.

VII. Conclusion and Future Work

This work presents phase-and-first-arrival feedback as a reusable interface between trajectory-level visual supervision and sparse-reward control. Phase identifies useful partial attempts, while first arrival localizes their replay, critic credit, and behavior reuse. Across simulation and hardware, the results show that a frozen structured record can preserve partial progress and support closed-loop learning without deployment-time VLM queries. Future work will extend support-aware acquisition and uncertainty calibration to raw-image policies, deformable interaction, and longer tasks.

References

[1] R. Bendikas, V. Modugno, D. Kanoulas, F. Vasconcelos, and D. Stoyanov, "Learning needle pick-and-place without expert demonstrations," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3326–3333, 2023.

[2] Y.-J. Ho, Z.-Y. Chiu, Y. Zhi, and M. C. Yip, "SurgIRL: Toward life-long learning for surgical automation by incremental reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 10, no. 12, pp. 13 145–13 152, 2025.

[3] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg, "Safety augmented value estimation from demonstrations (SAVED): Safe deep model-based RL for sparse cost robotic tasks," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3612–3619, 2020.

[4] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner, "Vision-language models are zero-shot reward models for reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[5] Y. Du, K. Konyushkova, M. Denil, A. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi, "Vision-language models as success detectors," *arXiv preprint arXiv:2303.07280*, 2023.

[6] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Bıyık, D. Held, and Z. Erickson, "RL-VLM-F: Reinforcement learning from vision language foundation model feedback," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024.

[7] T. M. Luu, Y. Lee, D. Lee, S. Kim, M. J. Kim, and C. D. Yoo, "Enhancing rating-based reinforcement learning to effectively leverage feedback from large vision-language models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2025.

[8] G. M. Fried, L. S. Feldman, M. C. Vassiliou, S. A. Fraser, D. Stanbridge, G. Ghitulescu, and C. G. Andrew, "Proving the value of simulation in laparoscopic surgery," *Ann. Surg.*, vol. 240, no. 3, pp. 518–528, 2004.

[9] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "EndoNet: A deep architecture for recognition tasks on laparoscopic videos," *IEEE Trans. Med. Imag.*, vol. 36, no. 1, pp. 86–97, 2017.

[10] T. Rueckert, R. Maerkl, D. Rauber, L. Klausmann, M. Gutbrod, D. Rueckert, H. Feussner, D. Wilhelm, and C. Palm, "Video dataset for surgical phase, keypoint, and instrument recognition in laparoscopic surgery (PhaKIR)," *arXiv preprint arXiv:2511.06549*, 2025.

[11] F. Richter, R. K. Orosco, and M. C. Yip, "Open-sourced reinforcement learning environments for surgical robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019.

[12] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou, and P.-A. Heng, "SurRoL: An open-source reinforcement learning centered and dVRK compatible platform for surgical robot learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021.

[13] P. M. Scheikl, B. Gyenes, R. Younis, C. Haas, M. Wagner, G. Neumann, and F. Mathis-Ullrich, "LapGym: An open source framework for reinforcement learning in robot-assisted laparoscopic surgery," *J. Mach. Learn. Res.*, vol. 24, 2023.

[14] Q. Yu, M. Moghani, K. Dharmarajan, V. Schorp, W. C.-H. Panitch, J. Liu, K. Hari, H. Huang, M. Mittal, K. Goldberg, and A. Garg, "ORBIT-Surgical: An open-simulation framework for learning surgical augmented dexterity," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[15] P. M. Scheikl, E. Tagliabue, B. Gyenes, M. Wagner, D. Dall'Alba, P. Fiorini, and F. Mathis-Ullrich, "Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 560–567, 2023.

[16] Y. Ou, A. Soleymani, X. Li, and M. Tavakoli, "Autonomous blood suction for robot-assisted surgery: A sim-to-real reinforcement learning approach," *IEEE Robot. Autom. Lett.*, vol. 9, no. 8, pp. 7246–7253, 2024.

[17] K. Baumli, S. Baveja, F. Behbahani, H. Chan, G. Comanici, S. Flennerhag, M. Gazeau, K. Holsheimer, D. Horgan, M. Laskin, C. Lyle, H. Masoom, K. McKinney, V. Mnih, A. Neitz, D. Nikulin, F. Pardo, J. Parker-Holder, J. Quan, T. Rocktäschel, H. Sahni, T. Schaul, Y. Schroecker, S. Spencer, R. Steigerwald, L. Wang, and L. Zhang, "Vision-language models as a source of rewards," *arXiv preprint arXiv:2312.09187*, 2024.

[18] S. A. Sontakke, J. Zhang, S. Arnold, K. Pertsch, E. Bıyık, D. Sadigh, C. Finn, and L. Itti, "RoboCLIP: One demonstration is enough to learn robot policies," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[19] T. M. Luu, D. Lee, Y. Lee, and C. D. Yoo, "Policy learning from large vision-language model feedback without reward modeling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2025.

[20] K.-H. Hung, P.-C. Lo, J.-F. Yeh, H.-Y. Hsu, Y.-T. Chen, and W. H. Hsu, "VICtoR: Learning hierarchical vision-instruction correlation rewards for long-horizon manipulation," *arXiv preprint arXiv:2405.16545*, 2024.

[21] C. Kim, M. Heo, D. Lee, J. Shin, H. Lee, J. J. Lim, and K. Lee, "Subtask-aware visual reward learning from segmented demonstrations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

[22] Q. Chen, J. Yu, M. Schwager, P. Abbeel, Y. Shentu, and P. Wu, "SARM: Stage-aware reward modeling for long horizon robot manipulation," *arXiv preprint arXiv:2509.25358*, 2025.

[23] R. Toro Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, "Using reward machines for high-level task specification and decomposition in reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018.

[24] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[25] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 6292–6299.

[26] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 1999.

[27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018.

[28] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan, Y. Gao, X. Li, T. Mu, N. Xiao, A. Gurha, V. N. Rajesh, Y. W. Choi, Y.-R. Chen, Z. Huang, R. Calandra, R. Chen, S. Luo, and H. Su, "ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI," in *Robotics: Science and Systems (RSS)*, 2025.

[29] M. Macklin, "Warp: A high-performance python framework for GPU simulation and graphics," NVIDIA GPU Technology Conference (GTC), 2022.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021.

[31] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.