

RoboJev: An Open Vision-Language Judge for Robot Success Detection

Josef Chen

Independent Researcher

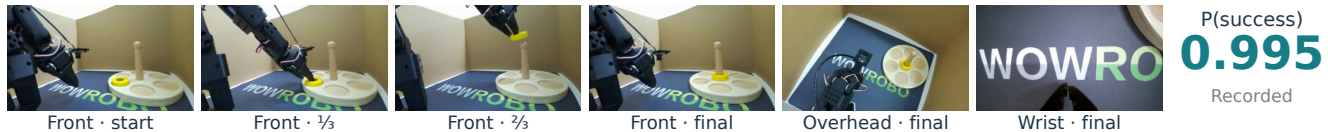
October 2026

Abstract. Many robot manipulation videos do not indicate whether the task was successfully completed. RoboJev-27B is an open-weight vision-language model that takes a task instruction, 10 frames of a manipulation attempt, and (if available) final frames from auxiliary cameras and predicts whether the robot accomplished the task. RoboJev-27B is JEV-27B-VL fine-tuned using a rank-32 LoRA applied only to the language model (vision encoder frozen) on 52,598 episode records, with a source-specific KL penalty applied only to research data to keep predictions anchored to the original model. It uses a calibrated two-token readout, outputting success/failure probabilities in a single forward pass. On 268 held-out episodes from the same robot model (SO-101), RoboJev-27B achieves 95.2% balanced accuracy and 0.993 AUROC; it generalizes to episodes from unseen labs/robots (AUROC = 0.890 over 459 episodes) and to clearly labelled community-collected episodes (accuracy = 93.3% over 135 episodes), where the underlying base model obtains AUROC = 0.900 on the unseen-lab/robot subset given the same inputs. Its inference time is 0.26 seconds per episode (median, serial) on a single H100, corresponding to an estimated \$0.19 per 1,000 success predictions at \$2.69 per GPU-hour (excluding media-processing and serving costs). We release RoboJev-27B along with inference code and a transparent evaluation containing 20,628 predictions across 29 model/protocol entries for browsing and reproducible success-detection benchmarking. We additionally compare against strong API-based frontier VLMs with fewer input frames (4 images from the primary camera only), noting that this comparison is not strictly like-for-like since RoboJev-27B receives more frames at inference.

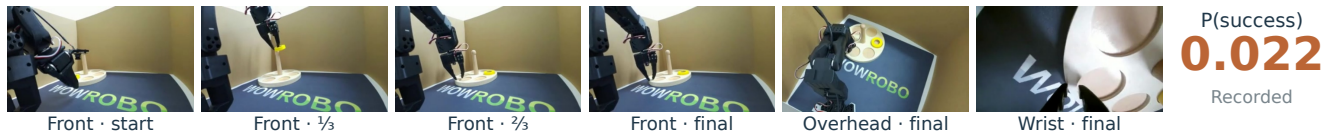
a Task: “Insert the colourful ring into the central wooden peg”

Ten front frames + final overhead and wrist views; four front frames shown below.

Successful attempt · episode 42



Failed attempt · episode 37



b

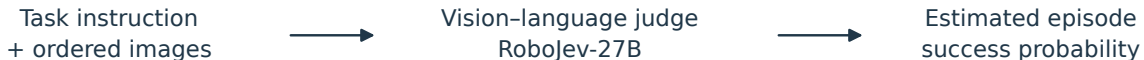


Figure 1. A task instruction and evidence of an attempt become a success probability. Real SO-101 ring-placement episodes illustrate the released inference interface. The judge reads a temporal sample and optional final auxiliary views, then scores the two answer tokens. Frames are from ArmnetBench; scores identify the exact camera protocol in the figure. These selected examples illustrate the interface and do not estimate benchmark performance.

1 Introduction

Robot interaction videos often lack machine-readable success labels. We present RoboJev, an open-weight episode judge fine-tuned from JEV-27B-VL [1, 2] to score whether a robot manipulation attempt ultimately succeeded, given the task instruction and a short uniform sample of camera frames. The

judge takes ten evenly spaced primary-camera frames plus the final frame from any auxiliary cameras, and emits calibrated success probabilities by scoring true / false tokens at a designated position without decoding a rationale.

Because success cannot be inferred from low-level features such as trajectory length, smoothness, or origin, a reusable

judge can support review queues, aggregate evaluation, and experiments with learned terminal rewards without a separate classifier for each task, provided it exposes probabilities, a documented input protocol, and auditable error records.

We evaluate RoboJev on held-out SO-101 rollouts, on lab/robot domains absent from robot-specific fine-tuning, and on clearly labelled community videos. Ambiguous community videos remain available for inspection but are excluded from binary metrics. The SO-101 split shares a physical setup with some training data, but holds out recording sessions and two complete tasks. RoboJev-27B reaches 95.2% balanced accuracy on this split, while on unseen-domain data it attains 0.890 AUROC, slightly below the 0.900 AUROC of the unadapted JEV-27B-VL run with the same input protocol. Together with released predictions, protocols, and a pinned checkpoint, these results position RoboJev as a practical, inspectable component for policy evaluation rather than evidence of universal gains from fine-tuning.

2 Related work

SuccessVQA treats robot outcomes as visual questions and evaluates generalization across different instructions and environments [3]. RoboCLIP generates trajectory rewards based on a textual instruction or video demonstration using frozen representations [4]. Video-Language Critic learns a transferable reward signal from a dataset of observations with contrastive and temporal-ranking losses [5]. RoboJev uses supervised binary success/failure labels and reports a detector that was publicly released; it does not report performance in a downstream policy learning setting.

RoboReward uses diverse robot data, counterfactual instructions, and temporal clips to train general reward models [6]. Robometer combines a frame-progress loss with trajectory preference supervision and publishes the RBM-1M dataset [7]. These targets and procedures differ from a terminal two-token prediction. Robometer-4B is evaluated here in its full-video form. RoboReward’s converted data is used during our training, but its model is not evaluated as a direct baseline. ViFailback provides failure trajectories that were initially recorded for diagnosis and subsequent correction [8].

Open X-Embodiment [9] and DROID [10] provide a wide variety of demonstration data. LeRobot [11] provides a shared format and set of tools for interacting with community datasets. All of these sources enable us to access different tasks and embodiments, yet their viewpoints, success/failure labels, and origins are diverse. We identify the processed subsets separately from their originating corpora in Appendix C.

Our implementation uses existing pieces: low-rank adaptation of the language model, a frozen vision encoder, a KL penalty towards the initial model on selected sources, and temperature scaling. We do not introduce a novel vision-language architecture. Our contribution is the released open-weight judge, the provided training and inference recipe, and the evaluation artifact describing the trade-offs of specialization, visual evidence, and deployment cost.

3 A task-conditioned episode judge

3.1 Input and decision readout

Let u be a task instruction, and $V = \{v_0, \dots, v_{N-1}\}$ a video captured by the primary camera. Our release samples $K = 10$ evenly spaced frames from the beginning to the end of the attempt, and includes the last image from each available auxiliary camera. The prompt specifies the task instruction, chronological order, and all other available views. Success requires completing the instruction by the end of the attempt; a dropped object, incorrect placement, unfinished attempt, or wrong task is negative.

The vision-language model reads through this information once and produces the logits z_0, z_1 at the answer location for the tokens `false` and `true`. We use the calibrated readout

$$p_{\theta}(y = 1 \mid u, V) = \frac{\exp((z_1 + b_1)/T)}{\sum_{c \in \{0,1\}} \exp((z_c + b_c)/T)}, \quad (1)$$

where the release saves the token IDs, additive biases b_c , and the temperature T in `decision.json`. The released temperature is $T = 0.9$. This output is an estimate of the terminal success of the trajectory; it is not a continuous metric of how far along in the task the agent is, nor is it an explanation of the outcome in natural language.

3.2 Training data

Our current model has been fit to 52,598 episode records consisting of 28,170 selected from the RBM-1M sources, 1,388 from ArmnetBench, 10,216 teacher labelled community records, 3,718 community demonstrations, 4,962 from ViFailback, and 4,144 successful DROID records. These are the post-split fit counts, not the number of records in the collection prior to the split. The data we fit on contains a mix of real policy failures, demonstrations, and simulation interactions. We hold out entire validation sources and exclude the community evaluation owners from training. The source mix is detailed in Appendix C, along with a minor unresolved inconsistency in the record count from the community loader. We do not claim that the fit record count is independently deduplicated.

The community loader reports 11,415 accepted records before validation exclusions. The executed annotation pipeline requires GPT-6.1 Sol and GPT-6 Astra to agree with high confidence from four primary-camera frames: each probability is at least 0.9 or at most 0.1. An available Claude Opus 5.5 label must agree on the binary side and can veto a candidate. The soft target averages all available teacher scores; the training loader applies a final 0.1 confidence margin to that average, retaining only means that meet the same extreme thresholds. The saved 11,387 unique target keys match this final consensus file. After removing validation records and held-out robots, 10,216 records enter model fitting. All 106 owners represented in the human-labelled community check are excluded from fitting. This separates test labels from training while retaining dependence between the student and teacher-model baselines.

We use inverse-frequency class balancing to weight success and failure examples within each source, with weights clipped to $[0.25, 4]$; sources with only one outcome retain weight 1. Data augmentation uses truncated positive episodes

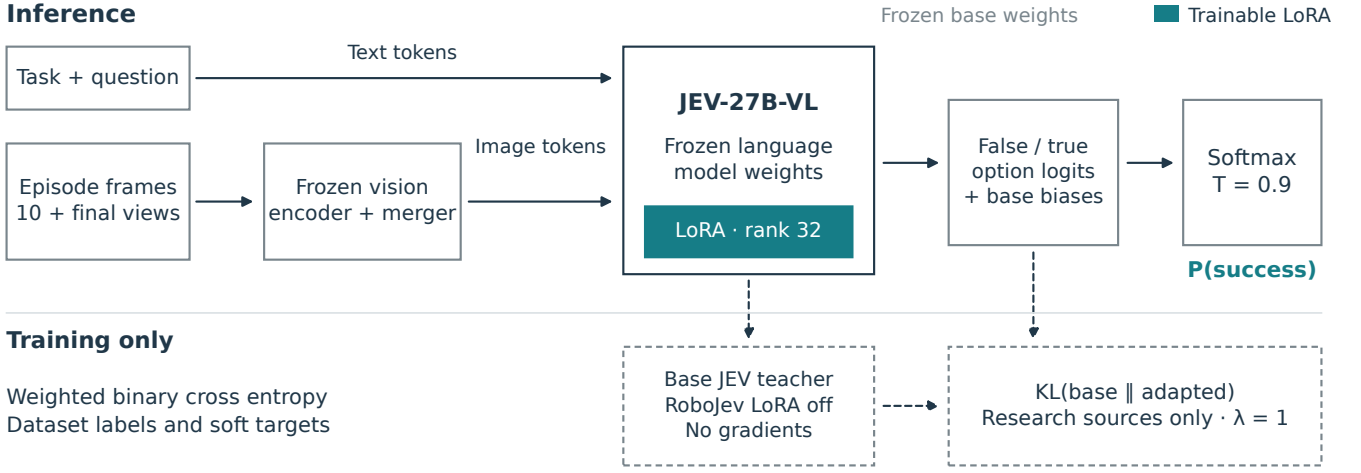


Figure 2. An existing vision-language backbone with a task-specific decision adapter. The vision encoder and merger are frozen. Rank-32 LoRA updates the language component. Training uses binary or soft success targets and a base-model KL anchor on selected research sources. At inference, the merged model scores two answer tokens and applies the released calibration. The training-only teacher branch is absent from serving.

and episodes with mismatched instructions as heuristic negatives. These steps can introduce label noise, since a truncated episode may already satisfy a brief instruction. We also resize, JPEG recompress, and perturb the colour of images. Eligible episodes with ten-frame evidence use that protocol with probability 0.6. The remaining augmentation branches preserve four-frame and final-view examples. Together, these augmentation and repetition strategies produce 102,172 training examples in the epoch.

3.3 Adaptation and calibration

We begin from JEV-27B-VL, merging in the decision adapter weights before performing any robot-specific adaptation. We use a rank-32 LoRA [12] on the language model with learning rate 5×10^{-5} for one epoch, keeping the vision encoder and visual merger frozen. Let q_θ and q_0 denote the two-way softmax outputs of our adapted model and the starting model, respectively, before fitted temperature scaling. Our per-example objective is

$$\mathcal{L}_i = w_i \text{CE}(y_i, q_\theta(x_i)) + \lambda a(x_i) D_{\text{KL}}(q_0(x_i) \| q_\theta(x_i)), \quad (2)$$

where w_i is the example weight (source-balanced for original records, and 0.5 for synthetic negatives), $a(x_i)$ indicates whether the example comes from an anchored source, and $\lambda = 1$. We use anchoring for RBM/research sources, including DROID and ViFailback, but not for ArmnetBench or community records. The target of the KL term is the starting model’s output distribution; the source-specific balancing factor scales only the cross-entropy term.

Adapter scaling and calibration use held-out validation sources. Following a weight-interpolation-style motivation [13], we evaluate scales $\{0, 0.25, 0.5, 0.75, 1\}$ on validation data and choose the nonzero scale with the highest held-out community AUROC, subject to remaining no more than 0.005 AUROC below the base on every specified validation source. These sources are AutoEval, FINO-Net, the RoboArena 0825 validation subset, and community Koch robots. All evaluated

Table 1. Evaluation composition. “Held out” has a different meaning in each setting; the SO-101 task-held-out subset contains 80 of the 268 runs.

Setting	Success	Failure	Scored
SO-101	111	157	268
Unseen domains	263	196	459
Community	60	75	135
Total	434	428	862

nonzero DA scales satisfy this constraint, and scale 1 achieves the highest community validation AUROC. We use AutoEval and FINO-Net purely for validation in this study, whereas a different RoboArena subset contributes training data. For temperature scaling [14], we minimise negative log-likelihood on validation data for the final-view, four-frame, and ten-frame protocols. Multiple protocols can share an episode. The published test sets were also examined during development of the released models, so they do not provide a completely one-shot blind test.

4 Evaluation protocol

4.1 Three complementary test settings

There are 927 labelled episodes in the evaluation artifact [15], but only 862 have a binary success/failure outcome and contribute to the metrics we report. The other 65 are community videos with unclear outcomes. Table 1 summarizes the scoring sets.

SO-101. ArmnetBench [16] includes policy rollouts from eight tasks and seven policies. Recording sessions are partitioned between training and testing. The tasks `cable_unclip` and `ring_insert` do not appear at all in the robot-specific training set and contribute 40 test episodes each. We report both the result on all 268 test runs and the result on this smaller

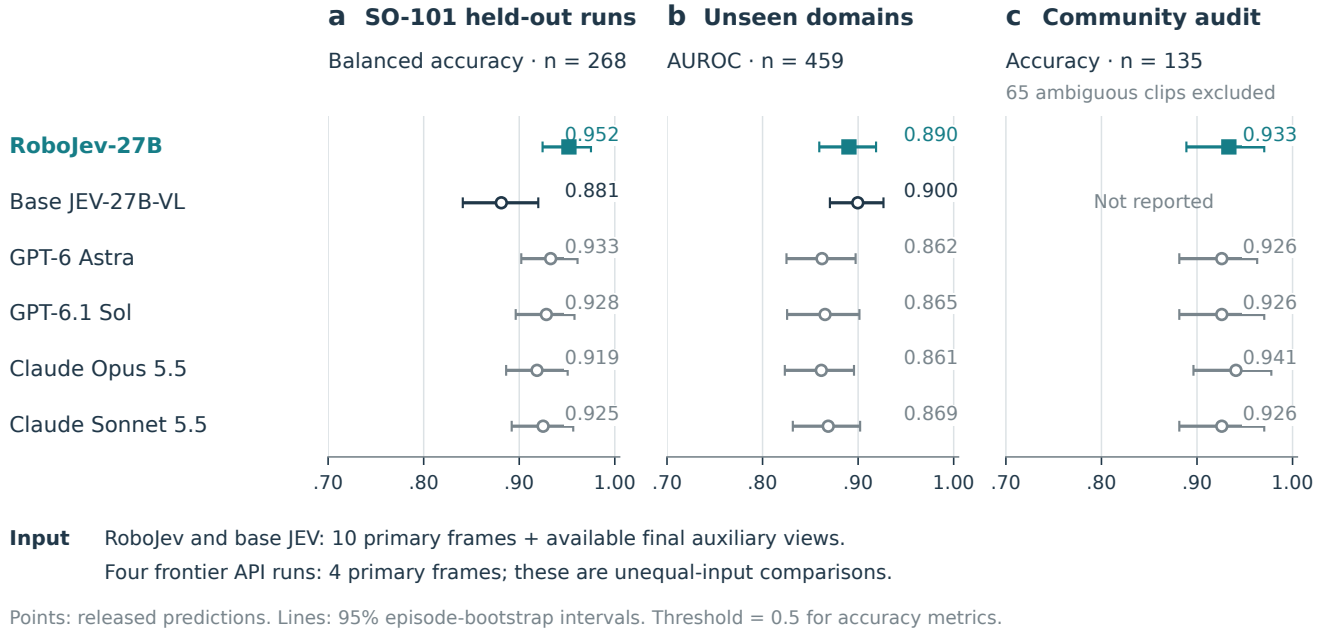


Figure 3. Performance depends on the evaluation domain and input protocol. The selected comparisons cover held-out SO-101 runs, unseen lab/robot domains, and the clearly labelled community subset. Error bars are episode bootstrap intervals. API baselines receive four primary-camera frames; RoboJev and the dense base receive ten plus available final auxiliary views. Sol has 458 recorded answers in the unseen-domain panel. The same-input base has higher unseen-domain AUROC, and Opus has higher community accuracy.

subset of held-out tasks. The overall result is not an evaluation on entirely unseen tasks.

Unseen lab/robot domains. The RBM-1M OOD set [17] consists of 459 rollouts from eight domains defined by the benchmark manifest, including Franka, Koch, Trossen, xArm, and SO-101 robots. Domain identifiers combine institution, robot, camera viewpoint, and/or scene condition. These domains do not appear in the robot-specific training set. Their sizes are uneven: the largest contains 218 runs and the smallest contains 16.

Community videos. These episodes come from publicly available LeRobot evaluation datasets. One person annotated all 200 episodes, one per dataset. We report metrics for the 135 episodes with a clear success/failure outcome and retain the other 65 as unclear in the artifact. The evaluation concerns outcomes observable to humans in community footage; it does not measure agreement among multiple human annotators.

4.2 Model and protocol comparisons

RoboJev-27B and the ten-frame JEV-27B-VL baseline are evaluated with ten primary-camera frames, as well as final frames from other cameras when available (dense-v1 protocol). Recorded frontier API runs use four primary-camera frames corresponding to the beginning, one third, two thirds, and end of the rollout (clip-v1 protocol). A Robometer result uses its native video-based interface with the full video. Older RoboJev checkpoints and different adapter scales appear as separate entries.

The released artifact contains 29 model/protocol entries, not

29 distinct architectures, for a total of 20,628 recorded predictions. Not all models provide predictions for all evaluation splits, and we treat missing responses as null. We exclude null predictions only when computing metrics for the corresponding model and use the intersection of answered episodes when comparing model pairs. Comparisons with the four-frame API models describe the recorded systems under their respective protocols; they do not isolate differences in model capability with matched visual input.

4.3 Metrics and uncertainty

We report AUROC and balanced accuracy, and ordinary accuracy for the community videos split. Balanced accuracy is the average of sensitivity and specificity. Binary predictions use a fixed threshold of 0.5. We also report the expected calibration error (ECE) with ten bins. Accuracy and AUROC were recomputed from every available prediction record, reproducing all 118 published values to their reported precision.

For the new figures, we use 4,000 bootstrap replicates of episodes stratified by label with random seed 20261024 and report percentile 95% confidence intervals. When comparing paired models, we use the same episode indices for both models in each replicate. We also retain the release’s previously published paired API intervals, which used 4,000 ordinary episode bootstrap replicates with random seed 0. These intervals quantify variability over episodes in these evaluation sets. They do not establish generalization to a new lab, account for within-domain correlation, or correct for multiple comparisons. We show the matched-input base comparison alongside the frontier API results.

5 Results

5.1 Strong SO-101 discrimination

On the 268 held-out SO-101 runs RoboJev-27B obtains 0.993 AUROC and 95.2% balanced accuracy, with ECE 0.032, versus 0.979 AUROC, 88.1% balanced accuracy and ECE 0.114 for the same-input ten-frame base model. RoboJev’s paired balanced-accuracy gain over that base is 7.08 percentage points (95% interval [3.04, 11.45]). On the 80 task-held-out SO-101 episodes, RoboJev attains 92.5% balanced accuracy (AUROC 0.986) compared with 91.3% (AUROC 0.978) for the same-input base. Astra scores 96.3% balanced accuracy (AUROC 0.987) with four frames on this subset; Sol attains 96.3% (AUROC 0.998, four frames) there. Thus the strong overall SO-101 result does not imply the best result on the withheld tasks.

GPT-6 Astra, which also supplies community training labels, scores 93.3% balanced accuracy with four frames on the full held-out SO-101 set. Against this run, RoboJev’s paired balanced-accuracy difference is +1.91 percentage points (95% interval [−1.56, 5.21]), which includes zero, so we make no resolved superiority claim there; note also the differing visual inputs.

5.2 Transfer reveals the limits of specialization

On the 459 unseen-domain runs RoboJev attains 78.2% balanced accuracy (0.890 AUROC), slightly below the same-input base’s 79.4% (0.900 AUROC); the paired AUROC difference is −0.0092 (95% interval [−0.0218, 0.0026]), which includes zero, indicating no evidence of transfer benefit from adaptation. Input-image hashes and task text match exactly between these adapted and base runs on all 727 SO-101 and unseen-domain episodes.

Against the recorded four-frame Astra run, the paired AUROC difference is +0.0282 (95% interval [0.0049, 0.0514]); this is a recorded-protocol comparison, not a controlled same-input architecture comparison. ECE remains 0.148 on unseen domains, substantially higher than on SO-101, which limits a literal probability interpretation after transfer.

5.3 Community accuracy and ambiguous outcomes

RoboJev attains 93.3% accuracy (0.984 AUROC) on the 135 clearly labelled community episodes (126 correct); Opus attains 94.1% and Astra and Sol 92.6%. These differences correspond to only one or two episodes. Against Astra the paired accuracy difference is 0.74 percentage points (95% interval [−4.44, 5.93]).

The 65 excluded ambiguous clips are 32.5% of the community sample, so we retain them in the released data artifact and do not extrapolate clear-subset accuracy to all community uploads. A detector should not turn an unobservable outcome into a confident ground-truth label.

5.4 Temporal evidence and anchoring

Relative to the prior four-frame RoboJev release (SO-101 AUROC 0.987; unseen-domain AUROC 0.872), this release obtains 0.993 and 0.890, respectively (training data and visual input also change, so this is an aggregate release comparison,

not a controlled effect estimate). Likewise the ten-frame base obtains unseen-domain AUROC 0.900 versus 0.880 under its four-frame protocol, representing an input-protocol (including prompt-wording) comparison rather than a controlled frame-count effect.

The training record also includes an unanchored run, DB, with 0.847 unseen-domain AUROC, compared with 0.890 for the anchored release, DA. This is consistent with the anchor helping preserve transfer, but it is a single run comparison without repeated seeds. Final-frame-only evaluation is weaker still on unseen domains: the released model reaches 0.776 AUROC. The evidence supports retaining temporal context for tasks whose result is not recoverable from one image.

5.5 Calibration changes across domains

Figure 5 reports reliability diagrams with ten equal-width probability bins. Calibration degrades across domains: ECE rises from 0.032 on SO-101 to 0.148 on unseen domains, and the corresponding Brier scores are 0.033 and 0.162. On the clear community subset, ECE is 0.027 and Brier score is 0.052. Bins can be sparse, so binomial confidence intervals are wide. The unseen-domain curve shows that the model frequently assigns probabilities below the observed bin success rates. These plots motivate local labelled validation when choosing thresholds for a new deployment.

6 Serving and intended uses

The merged checkpoint can be served through a standard vision-language inference endpoint. The client extracts the two answer-token log probabilities and applies Eq. 1. The released grader resolves video offsets from LeRobot episode metadata and writes per-episode probabilities. The overlay displays how the probability changes by scoring prefixes of the attempt as evidence arrives. A LeRobot plugin provides episode rewards and a processor for a thresholded terminal reward.

Serial inference on a single H100 has a measured median response time of 0.257 seconds. At a rental rate of \$2.69 per GPU-hour, the corresponding compute estimate is

$$C_{1000} = 1000 (0.257/3600) \times 2.69 \approx \$0.192. \quad (3)$$

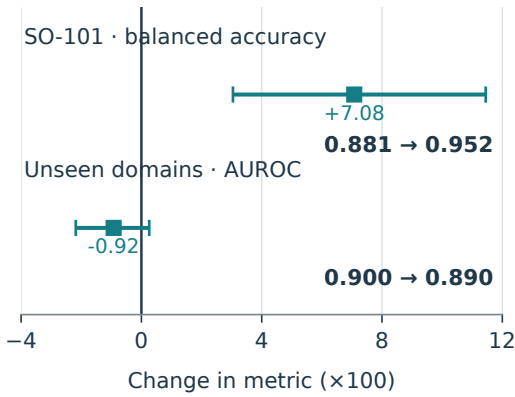
This estimate covers only GPU compute and excludes video download, decoding, queueing, GPU allocation, and idle capacity. API response times use different serving infrastructure and include their own overheads, so these are operational measurements rather than a controlled hardware comparison. The full bf16 checkpoint requires approximately 55 GB of storage and is intended for an 80 GB GPU.

The released catalogue samples 68,555 episodes from 7,641 public datasets, totalling 501.17 hours. Scaling the measured serial latency, one score per episode would take roughly 4.9 GPU-hours. This is an inference-cost scenario computed from the measured latency, not a report that the current checkpoint has already regraded that catalogue, and it excludes the same media and service costs as Eq. 3.

The results support reproducible offline evaluation and post-hoc episode review. The terminal-reward integration is implemented and testable, but we have not measured its effect on

a Matched-input adaptation

RoboJev – base JEV; same images and task text



SO-101: 10 front + 2 final views (n = 268).
Unseen domains: 10 primary frames (n = 459).
95% paired, label-stratified episode bootstrap.

b Latency and scoring cost

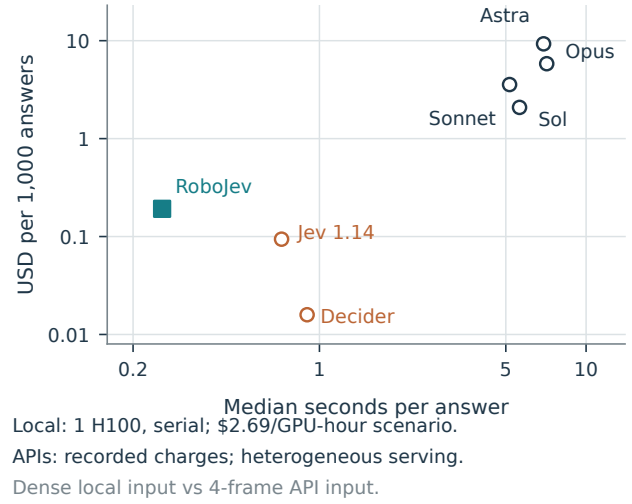
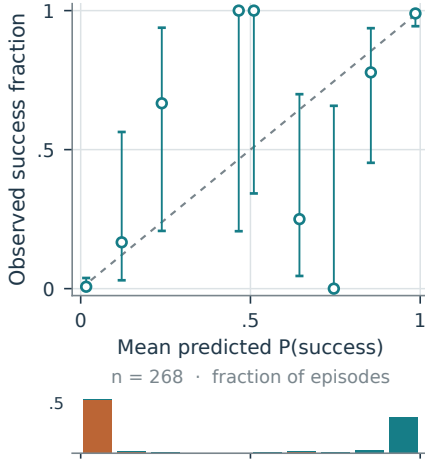


Figure 4. Matched-input adaptation and deployment cost. Left: the paired adapted-minus-base difference under verified identical inputs. The unseen-domain AUROC interval spans zero. Right: median response time and cost per 1,000 answers. The H100 point uses a GPU-hour cost estimate; API points use recorded charges. Infrastructure and input budgets differ, so these are operational comparisons rather than controlled hardware speedups.

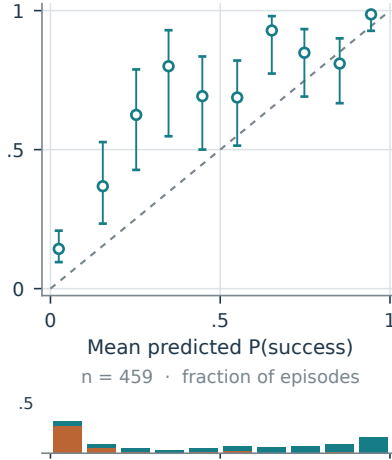
a SO-101 held-out runs

ECE₁₀ = 0.032 · Brier = 0.033



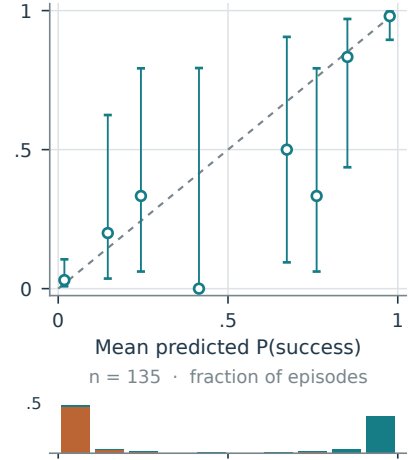
b Unseen domains

ECE₁₀ = 0.148 · Brier = 0.162



c Community audit

ECE₁₀ = 0.027 · Brier = 0.052



Nonempty 10-bin reliability estimates; 95% Wilson intervals. Outcomes:

■ Success ■ Failure

Figure 5. Good calibration in one setting does not transfer automatically. Reliability diagrams for the released model use ten equal-width bins; vertical intervals are 95% Wilson intervals for the observed success fraction. Histograms show the fraction of episodes in each bin, split by outcome. ECE and Brier score are computed from all scored episodes in each panel. The diagonal denotes perfect calibration. These are post-hoc diagnostics of the evaluation sets, not an additional calibration fit.

policy learning. Using this signal in an online controller requires separate closed-loop evaluation and an appropriate operating threshold: a controller can exploit reward errors or end an attempt prematurely.

7 Limitations and data responsibility

Observability. Success probabilities are conditional on what is visible in the provided video. If the relevant object or target is not visible (common in wrist-camera videos) or the clip ends before the outcome is clear, the model cannot certify success or failure. Likewise, sparse frame sampling may miss

brief events such as contact, spilling, or recovery.

Labels and evaluation reuse. Training labels from community datasets are model-generated, and agreement filtering may retain correlated errors. Human labels on community test videos come from a single annotator, and we do not have an estimate of inter-annotator agreement. Although public source overlap can be excluded at the level of the named split, we cannot rule out overlap with the pretraining data of the underlying foundation model. Related benchmarks were also examined during prior exploratory experiments, so the reported intervals should not be interpreted as a fully prospective, preregistered evaluation.

Domain coverage and calibration. The reported numbers cover eight test domains, providing limited evidence about the range of real deployments. Wrist-camera tasks and long-horizon actions such as stirring and folding remain challenging for success prediction. Temperature calibration was fit on validation data; this does not guarantee calibrated probabilities in new domains. We have not studied sensitivity to instruction paraphrases, a known failure mode of vision-language reward models [18]. Neither a low average error nor high confidence makes the detector a safety monitor.

Licensing and provenance. The released RoboJev checkpoint is licensed under Apache-2.0. The underlying training collection combines data with heterogeneous licences and provenance constraints: some research datasets accessed via RBM-1M do not provide dataset-specific licences, and a small number of community sources are no longer publicly available. We surface these gaps instead of treating public accessibility as blanket permission to redistribute every underlying asset. For real robot figures, the paper identifies the source, episode identifiers, and licence.

8 Conclusion

We presented RoboJev, an open-weight, task-conditioned success detector that outputs a success probability with a single forward pass, together with inference code and evaluation artifacts for transparent reuse. Its strongest measured result is 95.2% balanced accuracy on held-out SO-101 data. The unseen-domain comparison retains a meaningful counterexample in which the starting model ranks outcomes slightly better. The released checkpoint, prediction records, and inference tools support episode review and inspection of where the method helps and where it fails. Next steps include equal-input comparisons, additional human-labelled domains, and studies of how using RoboJev as a learned reward affects downstream policy training.

Artifacts. [Weights and inference tools](#); [evaluation records](#); [data explorer](#); [interactive demo](#).

Writing and verification tools. Language-model assistants helped with manuscript drafting, source inspection, code used

to produce figures, and consistency checks. Candidate edits proposed by StealthGPT were incorporated after being checked against the recorded methods and results. No robot images were generated. Deterministic analysis scripts computed the reported metrics from recorded predictions. The author takes responsibility for the content of the paper.

References

- [1] Autotrust. autotrust/JEV-27B-VL, 2026. URL <https://huggingface.co/autotrust/JEV-27B-VL>. Accessed 2026-10-24.
- [2] Josef Chen. RoboJev-27B model release, 2026. URL <https://huggingface.co/josefchen/RoboJev-27B/tree/e089a9b69a365744b80765057d86583647af95a9>. Pinned release (full revision in Appendix A). Accessed 2026-10-24.
- [3] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-Language Models as Success Detectors, 2023. URL <https://arxiv.org/abs/2303.07280>.
- [4] Sumedh A Sontakke, Jesse Zhang, Sébastien M. R. Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. RoboCLIP: One Demonstration is Enough to Learn Robot Policies, 2023. URL <https://arxiv.org/abs/2310.07899>.
- [5] Minttu Alakuijala, Reginald McLean, Isaac Woungang, Nariman Farsad, Samuel Kaski, Pekka Marttinen, and Kai Yuan. Video-Language Critic: Transferable Reward Functions for Language-Conditioned Robotics, 2024. URL <https://arxiv.org/abs/2405.19988>.
- [6] Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. RoboReward: General-Purpose Vision-Language Reward Models for Robotics, 2026. URL <https://arxiv.org/abs/2601.00675>.
- [7] Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, et al. Robometer: Scaling General-Purpose Robotic Reward Models via Trajectory Comparisons, 2026. URL <https://arxiv.org/abs/2603.02115>.
- [8] Xianchao Zeng, Xinyu Zhou, Youcheng Li, Jiayou Shi, Tianle Li, Liangming Chen, Lei Ren, and Yong-Lu Li. Diagnose, Correct, and Learn from Manipulation Failures via Visual Symbols, 2025. URL <https://arxiv.org/abs/2512.02787>.
- [9] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models, 2023. URL <https://arxiv.org/abs/2310.08864>.

- [10] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, et al. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset, 2024. URL <https://arxiv.org/abs/2403.12945>.
- [11] Remi Cadene, Simon Aliberts, Francesco Capuano, Michel Aractingi, Adil Zouitine, et al. LeRobot: An Open-Source Library for End-to-End Robot Learning, 2026. URL <https://arxiv.org/abs/2602.22818>.
- [12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, 2021. URL <https://arxiv.org/abs/2106.09685>.
- [13] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, et al. Robust fine-tuning of zero-shot models, 2021. URL <https://arxiv.org/abs/2109.01903>.
- [14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks, 2017. URL <https://arxiv.org/abs/1706.04599>.
- [15] Josef Chen. RoboJev evaluation records, 2026. URL <https://huggingface.co/datasets/josefchen/RoboJev-evals>. Benchmark manifests, predictions and protocol summaries. Accessed 2026-10-24.
- [16] Armnet. armnet/armnetbench_v01_robometer, 2026. URL https://huggingface.co/datasets/armnet/armnetbench_v01_robometer. Accessed 2026-10-24.
- [17] Robometer contributors. robometer/rbm-1m-ood, 2026. URL <https://huggingface.co/datasets/robometer/rbm-1m-ood>. Accessed 2026-10-24.
- [18] Wonje Jeung, Sangyeon Yoon, Hyesoo Hong, Yoonjun Cho, Dongjae Jeon, et al. Same Trajectory, Contradictory Rewards (ROBORMBENCH): Paraphrase Fragility in Vision Language Reward Models, 2026. URL <https://arxiv.org/abs/2609.05401>.
- [19] Robometer contributors. robometer/processed_datasets, 2026. URL https://huggingface.co/datasets/robometer/processed_datasets. Accessed 2026-10-24.
- [20] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, et al. MolmoAct: Action Reasoning Models that can Reason in Space, 2025. URL <https://arxiv.org/abs/2508.07917>.
- [21] Pranav Atreya, Karl Pertsch, Tony Lee, Moo Jin Kim, Arhan Jain, et al. RoboArena: Distributed Real-World Evaluation of Generalist Robot Policies, 2025. URL <https://arxiv.org/abs/2506.18123>.
- [22] Zhiyuan Zhou, Pranav Atreya, Abraham Lee, Homer Walke, Oier Mees, and Sergey Levine. Autonomous Improvement of Instruction Following Skills via Foundation Models, 2024. URL <https://arxiv.org/abs/2407.20635>.
- [23] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning, 2023. URL <https://arxiv.org/abs/2306.03310>.
- [24] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, et al. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning, 2019. URL <https://arxiv.org/abs/1910.10897>.
- [25] Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. AutoEval: Autonomous Evaluation of Generalist Robot Manipulation Policies in the Real World, 2025. URL <https://arxiv.org/abs/2503.24278>.
- [26] Arda Inceoglu, Eren Erdal Aksoy, Abdullah Cihan Ak, and Sanem Sariel. FINO-Net: A Deep Multimodal Sensor Fusion Framework for Manipulation Failure Detection, 2020. URL <https://arxiv.org/abs/2011.05817>.

A Reproducibility and protocol ledger

This manuscript describes the model released on 23 October 2026, with checkpoint pinned to e089a9b69a365744b80765057d86583647af95a9 in [josefchen/RoboJev-27B](#). The release ships the inference code and two-token calibration file with the weights. The paper artifact bundle includes snapshots of the released benchmark records, metric-generation code, figure source, and BibTeX entries, but no weights or private training videos.

Table 2. Inference and adaptation settings recorded for the released model.

Setting	Value and scope
Starting model	autotrust/JEV-27B-VL; its decision adapter is merged before robot adaptation.
Architecture class	Qwen3_5ForConditionalGeneration; model_type=qwen3_5 in the pinned checkpoint. The model card’s Qwen3.8 naming is not used to infer an independently verified foundation checkpoint lineage.
Adapter	Rank 32, language component; frozen vision encoder; full-strength exported merge.
Optimization	One epoch, learning rate 5×10^{-5} ; binary log loss plus source-specific KL anchor of weight 1.0.
Input protocol	Ten uniformly spaced primary-camera frames and the last frame of available auxiliary cameras.
Answer tokens	false: 3721; true: 1802.
Calibration	$T = 0.9$; $b_0 = +0.0001196011$, $b_1 = -0.0001198180$; fixed threshold 0.5 for reported binary metrics.
Image preprocessing	288-pixel short-side target/minimum with JPEG preprocessing; exact client behaviour is part of the release.
Serving measurement	One H100, serial requests; median 0.257 seconds per answer; media and allocation overhead excluded.

A.1 Named protocols

dense-v1

Ten evenly spaced frames from the primary-camera video, augmented with the final frames of the other cameras when present; used for the current release and the matched-input base-model row.

clip-v1

Four primary-camera frames at relative positions 0, 1/3, 2/3, 1; used for the recorded decision-API and frontier baselines and for earlier model variants.

done-v1

Final frame from every available camera; used for the final-frame diagnostic, not the headline release results.

video

The full clip, ingested through a baseline’s own video interface; a distinct input budget.

B Complete metric inventory

Tables are generated from released per-episode predictions and retain historical model/protocol rows for development and protocol comparisons. A dash means no answer was recorded for that setting; it is not zero performance. Each entry uses its available binary-labelled episodes. The separate release summary may contain historical aggregate scores for an entry whose individual predictions are unavailable in this snapshot.

Table 3. Complete available SO-101 and unseen-domain results. BA is balanced accuracy (%); AUC is AUROC. n is the number of available binary-labelled predictions. D: dense-v1; C: clip-v1; V: whole video.

Model / historical variant	Input	SO-101			Unseen domains		
		n	AUC	BA	n	AUC	BA
RoboJev-27B	D	268	0.993	95.2	459	0.890	78.2
GPT-6 Astra	C	268	0.987	93.3	459	0.862	77.8
GPT-6.1 Sol	C	268	0.989	92.8	458	0.865	79.1
Claude Opus 5.5	C	268	0.978	91.9	459	0.861	73.9
Claude Sonnet 5.5	C	268	0.980	92.5	459	0.869	74.4
JEV-27B-VL	C	268	0.983	87.7	459	0.880	77.4
TypeSafe Jev 1.14	C	268	0.975	82.9	459	0.841	72.1
Perplexity Decider	C	268	0.954	81.1	459	0.885	76.1
Liquid D1	C	268	0.897	82.8	459	0.828	74.7
Mercury Decide	C	268	0.938	81.6	459	0.859	75.6
Cloudflare Clef	C	268	0.907	77.5	459	0.803	72.0
Cloudflare Clef Flash	C	268	0.810	73.6	459	0.815	71.9
Cloudflare Clef Omni	C	268	0.700	56.3	459	0.729	58.1
Liquid D1 Ultra	C	268	0.918	81.4	459	0.859	73.5
Intern-Decision-4B	C	268	0.772	72.9	459	0.808	68.8
RoboJev-27B v1, scale 0.5	C	268	0.987	92.4	459	0.872	76.1
RoboJev-27B historical aggregate	C	—	—	—	—	—	—
RoboJev-4B v2	C	268	0.804	66.2	459	0.635	57.3
RoboJev-4B v3, scale 0.25	C	268	0.950	84.0	459	0.820	72.6
RoboJev-4B v3, scale 0.5	C	268	0.978	86.8	459	0.792	69.0
RoboJev-4B v3, scale 0.75	C	268	0.979	89.3	459	0.773	64.1
RoboJev-4B v3, scale 1.0	C	268	0.978	89.1	459	0.750	61.4
GPT-6 Luna Decisions	C	268	0.841	74.8	459	0.736	65.1
Perplexity Decider Flash v1.2	C	268	0.915	74.9	459	0.848	73.3
Respan Span-02	C	268	0.755	65.4	459	0.774	67.4
TypeSafe Jev 1.13	C	268	0.936	79.4	459	0.843	72.1
JEV-27B-VL (10 frames)	D	268	0.979	88.1	459	0.900	79.4
RoboJev-27B historical dense	D	—	—	—	—	—	—
Robometer-4B	V	268	0.777	67.7	459	0.818	59.8

Table 4. Available results on the 135 clearly labelled community episodes. Accuracy is ordinary accuracy (%), not balanced accuracy. The other 65 community clips are marked unclear.

Model / historical variant	Input	n	AUROC	Accuracy
RoboJev-27B	D	135	0.984	93.3
GPT-6 Astra	C	135	0.991	92.6
GPT-6.1 Sol	C	135	0.990	92.6
Claude Opus 5.5	C	135	0.988	94.1
Claude Sonnet 5.5	C	135	0.972	92.6

B.1 The two withheld tasks

The 80 episodes from `cable_unclip` and `ring_insert` contain 40 successes and 40 failures. This subset differs from the full 268-run SO-101 evaluation. The frontier rows still use fewer images.

Model and input	AUROC	Balanced accuracy (%)
RoboJev-27B	0.9862	92.50
JEV-27B-VL (10 frames)	0.9775	91.25
GPT-6 Astra	0.9872	96.25
GPT-6.1 Sol	0.9978	96.25
Claude Opus 5.5	0.9769	92.50
Claude Sonnet 5.5	0.9703	88.75

B.2 Model identities and primary links

The links below identify the evaluated service endpoints or upstream checkpoints. They do not verify historical provider routing, billing, or model internals. The recorded Astra request specified medium reasoning effort. The four frontier chat baselines returned a requested numerical probability; RoboJev and decision models use their documented decision readout. Historical RoboJev variants are linked to the release documentation rather than represented as independently downloadable checkpoints.

Table 5. Primary model sources. Multiple recorded variants share a source.

Model family / recorded variant	Exact model or endpoint identifier (linked)
RoboJev-27B and historical 27B/4B variants	josefchen/RoboJev-27B (release documentation)
GPT-6 Astra	openai/gpt-6-astra
GPT-6.1 Sol	openai/gpt-6.1-sol
Claude Opus 5.5	anthropic/claude-opus-5.5
Claude Sonnet 5.5	anthropic/claude-sonnet-5.5
JEV-27B-VL	autotrust/JEV-27B-VL
TypeSafe Jev 1.14	typesafe/jev-1.14
Perplexity Decider	perplexity/pplx-decider-v1.2
Liquid D1	liquid/d1
Mercury Decide	inception/mercury-decide
Cloudflare Clef	cloudflare/clef
cloudflare/clef-flash	cloudflare/clef-flash
cloudflare/clef-omni	cloudflare/clef-omni
liquid/d1-ultra	liquid/d1-ultra
open/intern-decision-4b	internlm/Intern-Decision-4B
openai/gpt-6-luna-decisions	openai/gpt-6-luna-decisions
perplexity/pplx-decider-flash-v1.2	perplexity/pplx-decider-flash-v1.2
respan/span-02	respan/span-02
typesafe/jev-1.13	typesafe/jev-1.13
open/robometer-4b	robometer/Robometer-4B

C Data sources and visual provenance

The source ledger labels each dataset as judge training, held-out validation, or evaluation-only, and records availability plus upstream licence as observed for this release. Community episode identifiers and source dataset links remain in the public benchmark and catalogue.

C.1 Training records and validation

Source family	Fit records	Local validation / exclusions
Selected RBM-1M sources	28,170	2,703 validation across all RBM sources
ArmnetBench	1,388	139 validation
Teacher-labelled community	10,216	917 validation + 282 Koch records
Community demonstrations	3,718	No local holdout in this loader
ViFailback	4,962	224 validation
DROID successful conversion	4,144	158 validation
Total fit records	52,598	102,172 examples after repetition/augmentation

The native validation outputs contain 4,413 unique episode IDs across protocols. The community loader reports 11,415 accepted records before exclusions, while the saved target map contains 11,387 unique dataset/episode keys. Its 282 held-out Koch records map to 272 unique validation IDs. Within-part duplicate metadata is a plausible mechanism, but the exact cause of the 28-record discrepancy has not been established. The deterministic owner/robot split places repeated keys on the same side; the discrepancy alone does not demonstrate train/test leakage.

Training-label provenance. The executed run reads the final consensus file. Its 11,387 saved target values match that file. The recorded pipeline requires high-confidence Sol and Astra labels on four primary-camera frames; any available Opus label must agree on the binary side and can veto a candidate. The target averages the available model probabilities and the loader applies its final confidence margin. Early public release text instead described Sol/Opus agreement on ten frames. This manuscript follows the executed queue, consensus script, and saved targets. The training teachers therefore overlap the API baseline panel.

Optimization details. Rank 32, LoRA alpha 64, dropout 0.05, and 233,455,616 trainable parameters; AdamW with zero weight decay, learning rate 5×10^{-5} , 5% warmup followed by cosine decay, and gradient norm clipping at 1.0. The per-GPU batch is 1 with accumulation 2. The documented seven-GPU run therefore has global optimizer batch 14. The random seed is 7. Temperature is selected from 0.25, 0.30, . . . , 5.20 by validation negative log-likelihood. The source records report approximately 5.5 hours on seven H100s; a 41.6-minute resumed-process log is not the total training duration.

C.2 Originating corpora and processed subsets

The processed RBM-1M repository [7, 19] supplies the exact subset identifiers below. Originating corpora include Open X-Embodiment [9], MolmoAct [20], RoboArena [21], RoboReward [6], SOAR [22], LIBERO [23], and Meta-World [24]. AutoEval [25] and FINO-Net [26] are whole-source validation sets. Failsafe and RACER are attributed through their exact processed-source identifiers; no unverified original-paper attribution is added. The harvest was not pinned to a complete historical repository revision, which limits byte-for-byte reconstruction of all training inputs. The bundled source ledger records the remote archive paths observed on 24 October 2026, separately from the historic training counts. There are 41 RBM source keys, of which 38 contribute fit records; these are not 41 independent originating datasets.

Table 6. Exact processed RBM source identifiers, linked to the verified repository. Fit counts are after local and whole-source holdouts.

Processed subset identifier	Fit	Validation
aliangdw_metaworld_metaworld_eval	107	11
aliangdw_metaworld_metaworld_train	97	3
jesbu1_auto_eval_rfm_auto_eval_rfm	0	800
jesbu1_failsafe_rfm_failsafe	600	40
jesbu1_fino_net_rfm_fino_net	0	229
jesbu1_molmoact_rfm_molmoact_dataset_household	300	40
jesbu1_molmoact_rfm_molmoact_dataset_tabletop	300	40
jesbu1_oxe_rfm_eval_oxe_bc_z_eval	300	40
jesbu1_oxe_rfm_eval_oxe_berkeley_cable_routing_eval	163	2
jesbu1_oxe_rfm_eval_oxe_bridge_v2_eval	300	40
jesbu1_oxe_rfm_eval_oxe_jaco_play_eval	104	5
jesbu1_oxe_rfm_eval_oxe_toto_eval	98	3
jesbu1_oxe_rfm_eval_oxe_viola_eval	14	1
jesbu1_oxe_rfm_oxe_aloha_mobile	256	16
jesbu1_oxe_rfm_oxe_austin_buds_dataset_converted_externally_to_rlds	45	5

Processed subset identifier	Fit	Validation
jesbu1_oxe_rfm_oxe_berkeley_cable_routing	300	40
jesbu1_oxe_rfm_oxe_berkeley_fanuc_manipulation	300	7
jesbu1_oxe_rfm_oxe_berkeley_mvp_converted_externally_to_rlds	300	20
jesbu1_oxe_rfm_oxe_berkeley_rpt_converted_externally_to_rlds	300	40
jesbu1_oxe_rfm_oxe_dlr_edan_shared_control_converted_externally_to_rlds	45	3
jesbu1_oxe_rfm_oxe_furniture_bench_dataset_converted_externally_to_rlds	300	40
jesbu1_oxe_rfm_oxe_iamlab_cmu_pickup_insert_converted_externally_to_rlds	300	21
jesbu1_oxe_rfm_oxe_imperialcollege_sawyer_wrist_cam	164	4
jesbu1_oxe_rfm_oxe_jaco_play	300	32
jesbu1_oxe_rfm_oxe_nyu_rot_dataset_converted_externally_to_rlds	14	0
jesbu1_oxe_rfm_oxe_stanford_hydra_dataset_converted_externally_to_rlds	300	30
jesbu1_oxe_rfm_oxe_tokyo_u_lsmo_converted_externally_to_rlds	47	1
jesbu1_oxe_rfm_oxe_toto	300	38
jesbu1_oxe_rfm_oxe_ucsd_kitchen_dataset_converted_externally_to_rlds	141	9
jesbu1_oxe_rfm_oxe_utaustin_mutex	300	40
jesbu1_racer_rfm_racer_train	600	40
jesbu1_roboarena_0825_rfm_roboarena	0	800
jesbu1_roboarena_eval_debug_nowrist_roboarena_eval_debug_nowrist	4,004	40
jesbu1_roboarena_rfm_roboarena_train	6,000	40
jesbu1_roboarena_rfm_roboarena_val	3,971	40
jesbu1_soar_rfm_soar_rfm	6,000	40
ykorkmaz_libero_failure_rfm_libero_10_failure	300	15
ykorkmaz_libero_failure_rfm_libero_90_failure	300	40
ykorkmaz_libero_failure_rfm_libero_goal_failure	300	13
ykorkmaz_libero_failure_rfm_libero_object_failure	300	23
ykorkmaz_libero_failure_rfm_libero_spatial_failure	300	12

Other source repositories. [ArmnetBench](#); [Teacher-labelled LeRobot community evaluations](#); [Community demonstrations](#); [ViFailback](#); [DROID successful episodes](#). The community evaluation and demonstration training source lists are not fully exposed by the released evaluation dataset. The supplied ledger therefore identifies the available provenance without representing it as a complete public training manifest. The public benchmark itself retains source dataset links for its evaluation episodes.

C.3 Licences and figure credits

The release declares Apache-2.0 for model weights and [ArmnetBench](#); MIT for RoboArena, LIBERO, Meta-World, and ViFailback; and CC BY 4.0 for RoboReward, MolmoAct, and DROID. Open X-Embodiment contains source-specific terms. The release audit found no explicit dataset licence for the SOAR, AutoEval, and FINO-Net copies accessed through RBM-1M. Community source availability has changed, and the complete training licence chain remains a provenance limitation. The source bundle re-distributes the paper figures and their documented [Armnet](#) frames, not the training corpus. Figure 1 uses [ArmnetBench SO-101](#) ring-placement episodes 42 (success) and 37 (failure), credited to [Armnet](#) under Apache-2.0. The local source snapshot records revision 1db69c82b589c7172ee31a6b8f7ef58d1fc2c663; the historical shard was not re-downloaded for this paper. The extracted input JPEGs are the exact recorded model inputs. The three-camera probabilities are 0.995478816 and 0.022072101 at the pinned RoboJev revision. The public front-only demo receipts for the same clips are different, 0.988252938 and 0.026133848. The figure uses the three-camera scores consistently. Every displayed frame has a SHA-256 hash in the bundled figure source manifest, together with the public dataset URL, episode index, video offsets, model revision, and query identifier. The figures contain no generated robot imagery or synthetic experimental screenshots.

D Verification and scope of the release

The source bundle’s metric-generation code recomputes the reported metrics from the released per-episode records on CPU only. This verifies arithmetic and coverage over those records, not fresh 27B inference or the provenance of every historical API call. The interactive demo records its pinned weights revision and hashes of the supplied example clips; its recorded examples are explicitly separate from new inference on an uploaded video.

We do not claim frontier superiority under equal inputs, general probability calibration, safety certification, a fully regraded community catalogue, or improved policy learning from the judge; each would require separate experiments.