

Learning Feasibility from Failure Data in Vision–Language–Action Models – Appendix –

Anonymous Author(s)

Affiliation

Address

email

1 Formulation

From low-level MDP to thought-level SMDP. Let the robot’s low-level interaction be an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, \rho)$ under an instruction ℓ , with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition kernel $T(\cdot \mid s, a)$, and initial distribution ρ . A *thought node* is a compact abstraction of the robot/world state: $n = \phi(s) \in \mathcal{N}$, where $\phi : \mathcal{S} \rightarrow \mathcal{N}$ extracts scene/plan tokens, spatial relations, etc. A *thought edge* $e \in \mathcal{E}$ is a temporally extended subgoal executed by a closed-loop controller (System 1) until a termination condition is met (e.g., $\langle \text{eoe} \rangle$). At *decision epochs* (the sequence of nodes (n_k)), the induced process over \mathcal{N} is a semi-Markov decision process (SMDP).

Definition of Options (Thought edges) We model each edge $e \in \mathcal{E}$ as an *option* with four components

$$e \equiv (I_e^N, \mu_e, \beta_e, \Pi_e).$$

(i) *Initiation set*) $I_e^N \subseteq \mathcal{N}$ specifies the nodes at which e is admissible. Execution at node n is feasible only if $n \in I_e^N$. (ii) *Intra-option policy*) $\mu_e(a \mid s, \ell)$ is a low-level, closed-loop controller over primitive actions, run at the control rate while e is active. (iii) *Termination rule*) $\beta_e : \mathcal{S} \rightarrow [0, 1]$ gives the probability to terminate upon arrival at state s' . We assume $\beta_e(s') = 1$ for $s' \in \mathcal{G} \cup \mathcal{F} \cup \mathcal{E}_e$, where $\mathcal{G} \subseteq \mathcal{S}$ and $\mathcal{F} \subseteq \mathcal{S}$ are disjoint *success/failure* sets ($\mathcal{G} \cap \mathcal{F} = \emptyset$), and $\mathcal{E}_e \subseteq \mathcal{S}$ is the edge-specific *end-of-edge* set (e.g., detected by a done head); otherwise $\beta_e(s') \in [0, 1)$. (iv) *Projection to nodes*) $\Pi_e : \mathcal{S} \rightarrow \mathcal{N}$ maps the terminal low-level state to the next decision node; we take $\Pi_e = \phi$ by default.

Given a start state s with $n = \phi(s) \in I_e^N$, option e induces a joint distribution over its *duration* $\tau \in \mathbb{N}$ (number of low-level steps while e is active) and terminal state $s_{t+\tau}$:

$$\kappa_e(\tau, s' \mid s) = \Pr(\tau_e = \tau, s_{t+\tau} = s' \mid s_t = s, e),$$

which is determined by T , μ_e , and β_e . This yields three mutually exclusive outcomes for executing e from node n (conditioning on the local deliberation context z , defined below):

$$P_{\mathcal{G}}(n, e, z) = \mathbb{E}_{s \sim \mu(\cdot \mid n, z)} \left[\sum_{\tau \geq 1} \sum_{s' \in \mathcal{G}} \kappa_e(\tau, s' \mid s) \right],$$

$$P_{\mathcal{F}}(n, e, z) = \mathbb{E}_{s \sim \mu(\cdot \mid n, z)} \left[\sum_{\tau \geq 1} \sum_{s' \in \mathcal{F}} \kappa_e(\tau, s' \mid s) \right],$$

$$P_N(n' \mid n, e, z) = \mathbb{E}_{s \sim \mu(\cdot \mid n, z)} \left[\sum_{\tau \geq 1} \sum_{\substack{s' \in \mathcal{E}_e \\ \phi(s') = n'}} \kappa_e(\tau, s' \mid s) \right],$$

with normalization $P_{\mathcal{G}}(n, e, z) + P_{\mathcal{F}}(n, e, z) + \sum_{n' \in \mathcal{U}_N} P_N(n' \mid n, e, z) = 1$. Here $\mu(\cdot \mid n, z)$ denotes the distribution over low-level start states consistent with the node $n = \phi(s)$ and the local context z . **The context z is the partial thought path $z_{0:k-1}$ that disambiguates history when ϕ is not lumpable.**

27 **Option Planning (Thought paths) and induced option policies.** A *thought path* $z_{0:K} =$
 28 (n_0, e_0, \dots, n_K) encodes a hypothetical reasoning rollout $n_0 \xrightarrow{e_0} n_1 \xrightarrow{e_1} \dots \xrightarrow{e_{K-1}} n_K$. Interpret-
 29 ing the sequence (e_0, e_1, \dots) as a non-stationary choice of options induces a policy π_z that *commits*
 30 to executing these edges in order at decision epochs. We define the *reasoning-level feasibility* of a
 31 path via the success-before-failure probability

$$\text{Fail}(z_{0:K}) = \Pr(\text{terminate in } \mathcal{F} \mid z_{0:K}, \ell), \quad \text{Succ}(z_{0:K}) = 1 - \text{Fail}(z_{0:K}).$$

32 With the entrance-reward $r(s, a, s') = \mathbb{I}\{s' \in \mathcal{G}\}$, the undiscounted return equals the success indi-
 33 cator, hence

$$\text{Succ}(z_{0:K}) = \Pr(\text{terminate in } \mathcal{G} \mid \pi_z) = \mathbb{E}_{\pi_z} \left[\sum_{t \geq 0} r_t \right] = V^{\pi_z}(n_0, z_{-1}),$$

34 i.e., Succ is the SMDP value of the policy that follows $z_{0:K}$.

35 **Why $\Pr(\text{terminate in } \mathcal{G} \mid \pi_z) = \mathbb{E}_{\pi_z} [\sum_{t \geq 0} r_t]$ holds.** Let $\mathcal{G}, \mathcal{F} \subset \mathcal{S}$ be disjoint absorbing sets
 36 and define the hitting times

$$\tau_{\mathcal{G}} := \inf\{t \geq 0 : s_t \in \mathcal{G}\}, \quad \tau_{\mathcal{F}} := \inf\{t \geq 0 : s_t \in \mathcal{F}\}, \quad \tau := \tau_{\mathcal{G}} \wedge \tau_{\mathcal{F}}.$$

37 Episodes *terminate at* τ . Use the entrance reward $r_t := \mathbf{1}\{s_{t+1} \in \mathcal{G}\}$ and the *undiscounted episodic*
 38 *return*

$$R := \sum_{t=0}^{\tau-1} r_t.$$

39 Then pathwise,

- 40 • If $\tau = \tau_{\mathcal{G}} < \tau_{\mathcal{F}}$: for all $t < \tau - 1$, $s_{t+1} \notin \mathcal{G}$, while at $t = \tau - 1$, $s_{t+1} \in \mathcal{G}$. Hence $R = 1$.
- 41 • If $\tau = \tau_{\mathcal{F}} \leq \tau_{\mathcal{G}}$: for all $t < \tau$, $s_{t+1} \notin \mathcal{G}$. Hence $R = 0$.

42 Therefore

$$R = \mathbf{1}\{\tau_{\mathcal{G}} < \tau_{\mathcal{F}}\} \implies \mathbb{E}_{\pi_z} \left[\sum_{t \geq 0} r_t \right] = \mathbb{E}_{\pi_z} [R] = \Pr_{\pi_z}(\tau_{\mathcal{G}} < \tau_{\mathcal{F}}) = \Pr(\text{terminate in } \mathcal{G} \mid \pi_z).$$

43 *Note.* Counting only up to the termination time τ (or, equivalently, stopping the process at τ) avoids
 44 double-counting in \mathcal{G} . This is exactly the “entrance-reward” convention.

45 **Why $\mathbb{E}_{\pi_z} [\sum_{t \geq 0} r_t] = V^{\pi_z}(n_0, z_{-1})$ holds.** Let the decision epochs be $t_0 = 0$, $t_{k+1} = t_k + \tau_k$,
 46 where τ_k is the (random) number of low-level steps taken while executing the k -th edge. Define the
 47 *decision-epoch reward* as the sum of step rewards accrued during that chunk:

$$\bar{R}_k := \sum_{t=t_k}^{t_{k+1}-1} r_t, \quad r_t = \mathbf{1}\{s_{t+1} \in \mathcal{G}\}.$$

48 These chunks $[t_k, t_{k+1})$ partition the primitive time axis up to termination, hence pathwise

$$\sum_{k \geq 0} \bar{R}_k = \sum_{t \geq 0} r_t. \tag{A}$$

49 By definition of the SMDP value at decision epochs under policy π_z ,

$$V^{\pi_z}(n_0, z_{-1}) := \mathbb{E}_{\pi_z} \left[\sum_{k \geq 0} \bar{R}_k \right]. \tag{B}$$

50 Taking expectations in (A) and using (B) yields

$$\mathbb{E}_{\pi_z} \left[\sum_{t \geq 0} r_t \right] = \mathbb{E}_{\pi_z} \left[\sum_{k \geq 0} \bar{R}_k \right] = V^{\pi_z}(n_0, z_{-1}).$$

51 **Expressing \bar{R}_k via (n, e, z) -level transition probabilities.** Recall

$$\bar{R}_k = \sum_{t=t_k}^{t_{k+1}-1} r_t, \quad r_t = \mathbf{1}\{s_{t+1} \in \mathcal{G}\}.$$

52 A chunk (edge e_k) halts on $\mathcal{G} \cup \mathcal{F} \cup \mathcal{E}$, so within chunk k the reward can occur at most once and
 53 only if the chunk *terminates in \mathcal{G}* . Equivalently, at the thought level

$$\boxed{\bar{R}_k = \mathbf{1}\{x_{k+1} = \mathbf{G}\}} \quad (x_k = (n_k, z_{0:k-1}), x_{k+1} \in \{\mathbf{G}, \mathbf{F}\} \cup (\mathcal{N} \times \mathcal{Z})).$$

54 Hence, conditionally on the decision state and chosen edge,

$$\boxed{\mathbb{E}[\bar{R}_k \mid n_k, e_k, z_{0:k-1}] = \Pr(x_{k+1} = \mathbf{G} \mid (n_k, z_{0:k-1}), e_k) = P_{\mathcal{G}}(n_k, e_k, z_{0:k-1})}.$$

55 **SMDP Bellman equations at decision epochs.** Let $V(n, z)$ denote the success-before-failure
 56 value at a decision epoch (n, z) . Two equivalent forms are useful.

57 (*Undiscounted, exact success probability*). Since termination in \mathcal{F} contributes zero value,

$$V(n, z) = \max_{e \in \mathcal{E}} \left[P_{\mathcal{G}}(n, e, z) + \sum_{n' \in \mathcal{U}_N} P_N(n' \mid n, e, z) V(n', z') \right], \quad z' = (z, n, e).$$

58 (*Discounted, per-step discount*). For numerical stability, a per-step discount $\gamma \in (0, 1)$ can be used;
 59 the semi-Markov backup weights continuation by the random duration τ :

$$V_{\gamma}(n, z) = \max_{e \in \mathcal{E}} \mathbb{E}[\gamma^{\tau} (\mathbf{1}\{\text{terminate in } \mathcal{G}\} + \mathbf{1}\{\text{terminate in } \mathcal{U}_N\} V_{\gamma}(n', z')) \mid n, e, z].$$

60 Setting $\gamma = 1$ recovers the exact success probability above. In practice we approximate γ^{τ} by a
 61 per-decision factor γ (see training target below).

62 **Action-value and greedy selection.** Define the option-value

$$Q(n, e, z) = \begin{cases} P_{\mathcal{G}}(n, e, z) + \sum_{n' \in \mathcal{U}_N} P_N(n' \mid n, e, z) V(n', z'), & \text{(undiscounted),} \\ \mathbb{E}[\gamma^{\tau} (\mathbf{1}\{\mathcal{G}\} + \mathbf{1}\{\mathcal{U}_N\} V_{\gamma}(n', z'))], & \text{(discounted).} \end{cases}$$

63 Then $V(n, z) = \max_e Q(n, e, z)$. This provides a principled target for the planner's leaf evaluation
 64 and edge selection.

65 **Node-level training target (sample backup).** Let the model estimate $V(n_k, z_{0:k-1}) \approx$
 66 $\Pr(\text{eventually } \mathcal{G} \mid n_k, z_{0:k-1}, \ell)$. Executing e_k yields one of three labeled outcomes in data: (i)
 67 \mathcal{G} , (ii) \mathcal{F} , or (iii) continuation at $n_{k+1} \in \mathcal{U}_N$. We use the one-step backup

$$y_k = \begin{cases} 1, & \text{if } e_k \text{ terminates in } \mathcal{G}, \\ 0, & \text{if } e_k \text{ terminates in } \mathcal{F}, \\ \gamma V(n_{k+1}, z_{0:k}), & \text{otherwise,} \end{cases} \quad \mathcal{L}_{\text{val}} = \mathbb{E}[(V(n_k, z_{0:k-1}) - y_k)^2],$$

68 which coincides with the discounted SMDP backup when γ approximates γ^{τ} , and with the exact
 69 success-probability backup when $\gamma = 1$.

70 **Admissible search set and objective.** At each planning cycle, System 2 conducts a bounded
 71 search over the thought graph $\mathcal{T} = (\mathcal{N}, \mathcal{E})$. Given a budget $\beta = (B, M)$ (expand at most B
 72 edges per step, for at most M expansion steps), let $T \leq M$ be the expansions actually used and \mathcal{B}_k
 73 the set expanded at step k . The admissible set of paths is

$$\mathcal{Z}(\beta) = \left\{ z_{0:K} \mid T \leq M, \forall k \in \{1, \dots, T\} : |\mathcal{B}_k| \leq B \right\}.$$

74 The planner selects the path with maximal reasoning-level feasibility:

$$\max_{z_{0:K} \in \mathcal{Z}(\beta)} \text{Succ}(z_{0:K}) = \max_{z_{0:K} \in \mathcal{Z}(\beta)} V^{\pi_z}(n_0, z_{-1}),$$

75 equivalently minimizing $\text{Fail}(z_{0:K})$.

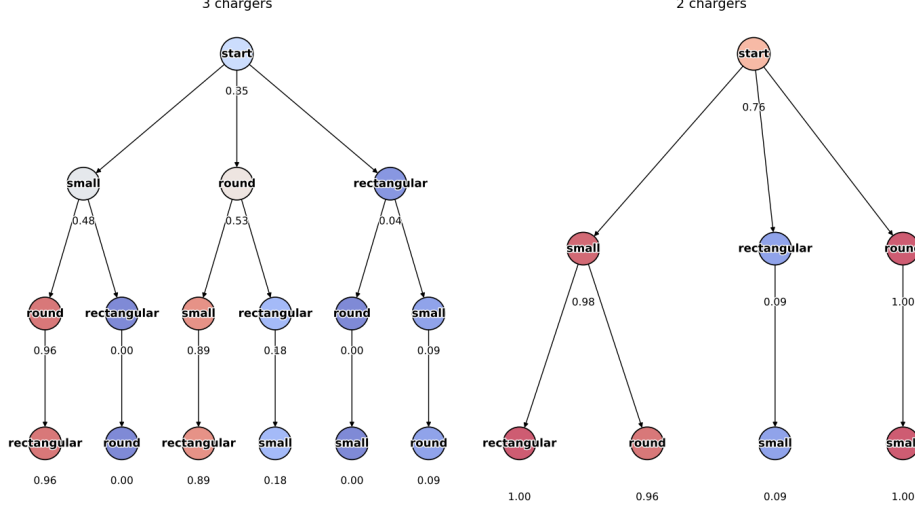


Figure 1: Data Statistics from human demonstrations

76 **Connection to the underlying MDP.** All SMDP quantities above are induced by the low-level
 77 MDP dynamics and the option controllers:

$$P_G(n, e, z) = \Pr(s_{t+\tau} \in \mathcal{G} \mid \phi(s_t) = n, e, z), \quad P_N(n' \mid n, e, z) = \Pr(\phi(s_{t+\tau}) = n' \in \mathcal{U}_N \mid \phi(s_t) = n, e, z),$$

78 with τ the (random) option duration. When ϕ is lumpable, the dependence on z vanishes and the
 79 process is Markov on \mathcal{N} ; otherwise, conditioning on z yields a consistent node-level formulation.

80 2 Dataset Details

81 **Collection.** We used teleoperation with four operators to cover *all admissible insertion orders*
 82 for each receptacle type (e.g., all permutations for 3-socket plates and for 2-socket plates). The
 83 insertion order for strips is fixed from right to left due to limited viewpoints. Operators received
 84 order guidelines beforehand so each trajectory explicitly followed a target sequence (e.g., small \rightarrow
 85 round \rightarrow rectangular). For every run we logged time-stamped robot state (arm pose, gripper
 86 command), RGB images, and camera calibration.

87 **Subgoal labels.** From the logs we derive a minimal set of discrete subgoals tied to gripper state
 88 transitions and the target order:

- 89 • **Grasp- x :** open \rightarrow close near plug x ,
- 90 • **Insert- x in y :** close \rightarrow open at socket y .

91 A trajectory thus becomes a chain of subgoals aligned to the instructed order (e.g., grasp-small
 92 charger \rightarrow insert-small charger in rightmost plug \rightarrow grasp-round charger $\rightarrow \dots$).

93 **Scene snapshots.** Around each transition time, we extract a short image window and select the
 94 central frame as the *scene* for that subgoal.

95 **Scene graph construction.** For every scene, we build a 2D scene graph containing:

- 96 • **Gripper node:** 2D image location obtained by projecting the measured 3D gripper pose
 97 into the camera frame using known intrinsics/extrinsics.
- 98 • **Object nodes:** bounding-box proposals from *GroundingDINO* [1]; a VLM (*Gemini* [2])
 99 assigns semantic names (e.g., small, round, rectangular) to crops via text prompts
 100 seeded by the task vocabulary.

101 • **Relations (edges):** coarse spatial relations between nodes (e.g., charger inside leftmost
102 plug), inferred from box geometry and, when ambiguous, VLM judgments.

103 **Output triplets.** Each subgoal yields a {image, scene-graph, label} triple: the raw image, a graph
104 with the gripper 2D position, object boxes, relations, and the symbolic subgoal induced by the
105 gripper transition and the instructed order.

106 **Data Statistics** Figure 1 visualizes the empirical *per-path success rate* for the plug-insertion
107 task from human demonstrations. Each tree enumerates all candidate insertion paths (from the
108 root start to a leaf), where a leaf corresponds to a *complete* order of socket types (small,
109 round, rectangular). The number under each leaf is the estimated success probability $\hat{p}_{\text{succ}} =$
110 $\# \text{successes} / \# \text{trials}$ for that path, and node color encodes this value (red \rightarrow high, blue \rightarrow low). The
111 left panel reports the 3-charger setting and the right panel the 2-charger setting. The distribution
112 is skewed: in 3–3-charger, sequences such as small \rightarrow round \rightarrow rectangular are highly reli-
113 able (0.96), while any path involving rectangular early tends to fail (0.00–0.18). In a 2-charger,
114 several paths are near-perfect (e.g., round \rightarrow small and small \rightarrow rectangular both at 1.00),
115 whereas rectangular \rightarrow small is weak (0.09).

116 References

- 117 [1] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding
118 dino: Marrying dino with grounded pre-training for open-set object detection. In *Proc. of the*
119 *European conference on computer vision (ECCV)*, pages 38–55. Springer, 2024. 4
- 120 [2] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein,
121 O. Ram, D. Zhang, E. Rosen, et al. Gemini 2.5: Pushing the frontier with advanced rea-
122 soning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint*
123 *arXiv:2507.06261*, 2025. 4