FAITHFUL RULES ARE NOT ENOUGH: BOOLEAN SURROGATES FOR DEEP RL AND AN AUDIT OF THEIR CONCEPTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Rule surrogates are among the most readable explanations of deep reinforcement learning (DRL) policies, and their quality is almost always judged by a single number: how often the rules reproduce the policy's actions. We ask whether that number says anything about the concepts the rules are written in. We first build a strong surrogate for vision-based policies. LUCID jointly trains a sparse autoencoder on the frozen policy's features, a learnable binarization bottleneck and a product t-norm logic layer, and extracts Disjunctive Normal Form (DNF) rules; once the selectors polarize, the extracted rules are invariant over a wide range of extraction thresholds, and a state-level certificate bounds the disagreement between the rules and the continuous surrogate. Across five vision-based tasks, from MiniGrid to Pong and Boxing, the rules agree with the PPO teacher on 96.8–100% of states and need about $2\times$ fewer feature tests than a decision tree of matched fidelity on gridworlds and 28–$46\times$ fewer on Atari; controls show that joint training is what keeps them faithful after extraction. We then audit the concepts along two axes: alignment with simulator ground-truth factors, and causal effect under intervention on the frozen teacher. Under our criteria the two never hold together in any of the five environments, and on Dynamic-Obstacles rules with near-perfect fidelity rest on concepts whose manipulation moves the teacher away from the predicted action. VLM-generated concept labels, tested blind, predict their concepts above chance but well below certainty. Behavioural fidelity, factor alignment and causal efficacy are distinct properties, and rule-extraction work should report all three.

1 INTRODUCTION

Post-hoc surrogates explain a trained deep RL policy without changing it: a decision tree (Bastani et al., 2018), a concept bottleneck (Koh et al., 2020; Ye et al., 2025) or a logic program is fitted to the frozen agent, and the explanation is read off the surrogate. The standard evidence that such an explanation is good is *fidelity*, the fraction of states on which the surrogate takes the agent's action, sometimes complemented by the return it obtains when deployed. For policies over structured state vectors this is a reasonable proxy, because the atoms of the rules (positions, velocities, predicates) are defined by a human and mean what they say. For vision-based policies they are not: the atoms must be learned from convolutional features that are *polysemantic*, with single units encoding several entangled factors. The same fidelity number is then asked to vouch for two things at once, that the rules reproduce the policy and that the concepts they are written in are meaningful.

Existing surrogates avoid the second question rather than answer it. Decision trees on polysemantic features either stay shallow and lose accuracy or grow deep and lose readability (Frosst & Hinton, 2017). Concept bottlenecks fix the concept set before training; LICORICE (Ye et al., 2025) reduces the number of labels and can obtain them from a VLM, but behaviour that depends on an unlisted feature cannot be expressed. Sparse autoencoders (SAEs) decompose dense activations into sparse, often more interpretable features (Cunningham et al., 2023; Lim et al., 2025), which suggests learning the atoms instead of specifying them, but then nothing but fidelity is left to certify them.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We address both questions. We first build a strong surrogate, so that weaknesses of its concepts cannot be attributed to a weak surrogate. LUCID trains an SAE, a learnable binarization bottleneck and a product t-norm DNF layer jointly on the frozen teacher's features, and extracts Boolean rules from them. We then audit the concepts of those rules with two evaluations that the literature usually conflates: whether a concept *names* a factor a human would specify, measured against simulator ground truth, and whether it *drives* the decision, measured by intervening on the frozen teacher. Our contributions:

- **A faithful, compact rule surrogate for vision policies** (Section 3). Once selectors polarize the extracted rule set is invariant over a central threshold interval (Proposition 1), and a state-level certificate, linked to the teacher by a triangle bound, bounds soft–hard disagreement (Theorem 1).
- **Evidence that the rules are faithful for the right reason** (Section 4). On five tasks the rules agree with the teacher on 96.8–100% of states. Controlled ablations show that joint training is what makes extraction nearly lossless, and a fidelity–complexity sweep shows the advantage over trees is small on gridworlds and large on Atari.
- **A concept audit with a negative main finding** (Section 5). Factor alignment and causal efficacy never hold together in any environment under our criteria, and fidelity of 1.000 coexists with causally counterproductive concepts.
- **A blind evaluation protocol for VLM concept labels** (Section 5), with the namer and a scorer from different developers, distractor labels and a positive control.

2 Related Work

Interpretable and neuro-symbolic RL. Symbolic policies (Landajuela et al., 2021), differentiable ILP (Evans & Grefenstette, 2018), S-REINFORCE (Dutta et al., 2023) and SINDy-RL (Zolman et al., 2025) replace the policy by an interpretable one; Neural DNF-MT (Baugh et al., 2025) learns DNF policies end to end with actor–critic methods, and NUDGE (Delfosse et al., 2023b) searches logic policies guided by a neural one but needs object-centric symbolic input. LUCID instead explains a frozen pixel-based policy post hoc. **Post-hoc surrogates and concept bottlenecks.** VIPER (Bastani et al., 2018) and CART (Breiman et al., 1984) clone the policy into trees; soft trees (Frosst & Hinton, 2017) trade readability for accuracy; CBMs (Koh et al., 2020; Ye et al., 2025) route decisions through predefined concepts. **Discovered concepts.** SAEs recover sparse features in language (Bricken et al., 2023; Cunningham et al., 2023) and vision models (Lim et al., 2025); Discover-then-Name (Rao et al., 2024) names SAE features of CLIP by text alignment; Bekkemoen & Langseth (2025) distil CNN policies into sparse linear models over NMF concepts; and RAMi (Liu et al., 2021) learns a disentangled object representation from pixels and then fits compact regression trees that mimic the agent's Q-values. These works evaluate discovered concepts by fidelity or by inspection; we add factor alignment, intervention and blind label tests on the same dictionaries. An extended discussion is in Appendix B.

3 LUCID

Surrogate. Let $x \in \mathbb{R}^d$ be the teacher's standardized CNN features of an observation and a^{base} its action (Fig. 1). An overcomplete TopK SAE ($D \gg d$) produces sparse concepts, and a learnable per-concept sigmoid pushes each toward a binary state:

$$h = \mathrm{ReLU}(W_e(x - b_d) + b_e), \quad z = \mathrm{TopK}_k(h), \quad \hat{x} = W_d z + b_d, \qquad c_j = \sigma\big(\alpha_j(z_j - \beta_j)\big), \tag{1}$$

with $\alpha_j = \mathrm{softplus}(\rho_j) + \frac{1}{2}$. Each action a has n_a clauses. A 3-way softmax gives, for clause i and concept j, the probabilities $p_{a,i,j}$, $n_{a,i,j}$ and $u_{a,i,j}$ that c_j must be present, absent, or is ignored, and

$$\ell_{a,i,j} = p_{a,i,j}c_j + n_{a,i,j}(1 - c_j) + u_{a,i,j}, \quad C_{a,i} = \sigma\Big(\tilde{b}_{a,i} + \textstyle\sum_j \log(\ell_{a,i,j} + \epsilon)\Big), \quad s_a = \textstyle\sum_i C_{a,i}, \tag{2}$$

where $\tilde{b}_{a,i} = \min(b_{a,i}, 5)$ is a clipped clause bias; without the cap a clause could fire regardless of its literals. All parts are trained jointly on

$$\begin{aligned}\mathcal{L} = {} & \lambda_{\text{rec}} \tfrac{1}{d}\|x - \hat{x}\|_2^2 + \lambda_1 \tfrac{1}{D}\|h\|_1 + \lambda_{\text{CE}}\,\mathrm{CE}\big(\mathrm{softmax}(s), a^{\text{base}}\big) \\ & + \omega_t \lambda_{\text{bim}} \tfrac{1}{D} \textstyle\sum_j c_j(1 - c_j) + \lambda_{\text{L0}}\mathcal{L}_{\text{L0}} + \omega_t \lambda_{\text{ent}}\mathcal{L}_{\text{ent}},\end{aligned} \tag{3}$$

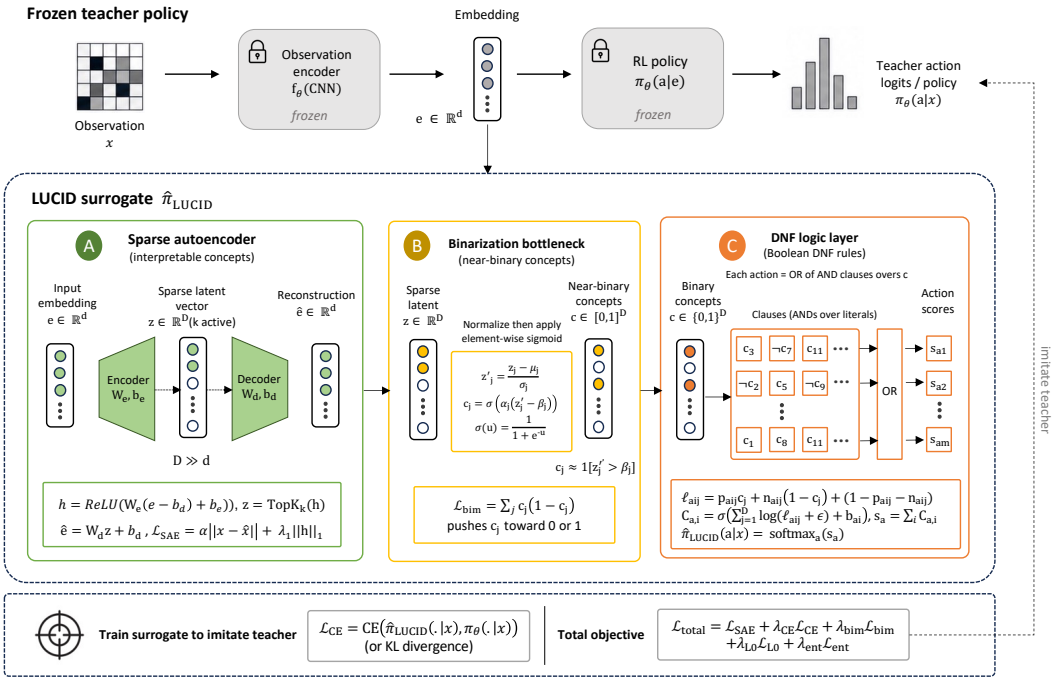

Figure 1: LUCID. A sparse autoencoder, a binarization bottleneck and a product t-norm DNF layer are trained jointly on the frozen teacher's features; Boolean rules are then extracted by thresholding the literal selectors.

where $\mathcal{L}_{\text{L0}}$ is the mean selector mass $p + n$, $\mathcal{L}_{\text{ent}}$ the entropy of $[p, n, u]$, and ω_t ramps linearly from 0 to 1 after a warmup, so that concepts and rules fit the teacher before discreteness is enforced (Appendix I). We attribute the stability of joint training to TopK sparsity (not ablated here): the imitation gradient reaches only the k active concepts of each input, while the remaining atoms are shaped by reconstruction. The discreteness terms then contract the bottleneck outputs and the selectors toward $\{0, 1\}$, which freezes the combinatorial structure of the DNF while its continuous parameters are still moving, so that dictionary and rules stabilize together rather than one collapsing the other. The same terms tighten the certificate below: near-binary concepts make a satisfied literal contribute almost no drag to a true clause and a violated literal almost no mass to a false one.

Rule extraction. A selector threshold τ defines literal sets $P_{a,i} = \{j : p_{a,i,j} > \tau\}$ and $N_{a,i} = \{j : n_{a,i,j} > \tau\}$, and a concept threshold θ binarizes concepts, $c_j^{\text{hard}} = \mathbf{1}[c_j > \theta]$ ($\tau = \theta = 0.66$ throughout). Three conventions are fixed: a clause with $P_{a,i} \cap N_{a,i} \neq \emptyset$ (possible only for $\tau < 1/2$) is `False`; a clause with no selected literal is *universal*, always `True`, acts as a constant prior for its action and is not counted in complexity; and the clipped bias is used everywhere. With $T_{a,i}(x)$ the truth value of the conjunction,

$$s_a^{\text{hard}}(x) = \sum_i \sigma(\tilde{b}_{a,i})\, \mathbf{1}[T_{a,i}(x)], \qquad \pi^{\text{hard}}(x) = \begin{cases} \arg\max_a s_a^{\text{hard}}(x) & \text{if some clause fires,} \\ a_{\text{fallback}} & \text{otherwise,} \end{cases} \tag{4}$$

with a_{fallback} the most frequent teacher action in the training data.

Proposition 1 (Central-interval invariance; proof in Appendix A.5)**.** *For $\tau \geq 1/2$ the literal sets are disjoint and every selected literal holds a strict majority of its selector mass; for $\tau \geq 2/3$ its discarded mass is below the uniform prior $1/3$. If no $p_{a,i,j}$ or $n_{a,i,j}$ lies in $(\delta, 1 - \delta)$ with $\delta < 1/3$, the extracted rule set is identical for every $\tau \in [1/2, 2/3]$.*

In all five environments no selector falls in $(0.5, 0.9]$, so the rules are identical for every $\tau \in [0.5, 0.9]$ and τ is not a tuning knob. Appendix C casts the choice of τ as minimizing the soft–hard disagreement plus λ times the number of literals; both terms are piecewise constant in τ (Lemma 4), so on this

Table 1: Return, action fidelity to the teacher (%) and rule size (feature tests; Soft-DT: non-zero split weights). Return: mean over evaluation seeds 42–46. Fidelity is measured on the teacher rollouts used for training (LUCID) or on a 20% split held out from them (trees); on independent rollouts LUCID's hard fidelity is within 0.13 points (Table 5). LUCID: training seed 42; three seeds in Table 15. Bold: best post-hoc surrogate.

		PPO	LUCID soft	LUCID hard	VIPER	SA-DT	Soft-DT	LICORICE
DoorKey	Return	0.968	0.968	**0.968**	0.955	**0.968**	0.937	0.968
	Fidelity	–	100.00	**99.92**	99.16	99.86	84.55	–
	Size	–	–	**16**	30	44	396	–
Dyn.-Obs.	Return	0.891	0.891	0.891	0.891	0.891	0.891	0.891
	Fidelity	–	100.00	**100.00**	98.37	**100.00**	**100.00**	–
	Size	–	–	**4**	5	5	48	–
PixelCartPole	Return	500.0	475.5	442.5	402.5	370.5	**494.0**	500.0
	Fidelity	–	99.99	97.13	95.67	96.25	**99.10**	–
	Size	–	–	45	**34**	64	204	–
Pong	Return	21.0	21.0	**21.0**	20.9	0.3	**21.0**	21.0
	Fidelity	–	99.90	**99.85**	63.02	62.34	88.85	–
	Size	–	–	**19**	190	240	5766	–
Boxing	Return	92.2	94.1	**92.5**	90.1	74.7	92.0	92.6
	Fidelity	–	99.78	**96.80**	69.20	69.96	86.43	–
	Size	–	–	**93**	214	377	8496	–

interval the objective is flat for every λ and no trade-off weight has to be chosen. We use the literal count only as a sparsity proxy, not as a code length in the MDL sense (Grünwald, 2007).

Soft–hard certificate. Binarization can flip the arg-max only where the hard margin is small. Let $\gamma(x) = s^{\mathrm{hard}}_{a^*}(x) - \max_{a \neq a^*} s^{\mathrm{hard}}_a(x)$ with $a^* = \pi^{\mathrm{hard}}(x)$, and let $\Delta_{a,i}(x)$ be the per-clause bound of Eq. (24) (Appendix D): the exact log-drag of a true clause, and for a false clause a bound from its most violated literal alone.

Theorem 1 (Soft–hard fidelity; proof in Appendix A.3)**.** *Let $\Delta_a(x) = \sum_i \Delta_{a,i}(x)$. If some clause fires at x, $\gamma(x) > 0$ and $\Delta_{a^*}(x) + \Delta_a(x) < s^{\mathrm{hard}}_{a^*}(x) - s^{\mathrm{hard}}_a(x)$ for all $a \neq a^*$, then $\pi^{\mathrm{soft}}(x) = \pi^{\mathrm{hard}}(x)$. Hence, with $\mathcal{X}_{\mathrm{cert}}$ the set of such states, $\Pr[\pi^{\mathrm{soft}} \neq \pi^{\mathrm{hard}}] \leq 1 - \Pr[x \in \mathcal{X}_{\mathrm{cert}}]$.*

The theorem relates the rules to the soft surrogate, not to the teacher. The triangle inequality links the two: $\Pr[\pi^{\mathrm{hard}} \neq \pi^{\mathrm{teacher}}] \leq (1 - \mathrm{AF}_{\mathrm{soft}}) + (1 - \Pr[x \in \mathcal{X}_{\mathrm{cert}}])$, where $\mathrm{AF}_{\mathrm{soft}}$ is the soft surrogate's action fidelity. Every term is measurable, so the bound can be checked on any trajectory.

4 Faithful and Compact Rules

Setup. We use *MiniGrid-Dynamic-Obstacles-5x5* and *MiniGrid-DoorKey-6x6* (Chevalier-Boisvert et al., 2018), *PixelCartPole* (`CartPole-v1` rendered as 84×84 grayscale, 4-frame stack, episodes capped at 500 steps) and the Atari games (Bellemare et al., 2013) *Pong* and *Boxing* (6 and 18 actions). These are similar to the five environments of LICORICE (Ye et al., 2025), the closest concept-based method for vision-based RL; we exclude continuous control because pixel-based teachers performed poorly there (Appendix G). Each teacher is a PPO agent (Schulman et al., 2017); LUCID is trained on its features from teacher rollouts with a 90/10 train/validation split. The tree baselines receive the same frozen CNN features without an SAE: SA-DT (CART by behavioural cloning), VIPER (Bastani et al., 2018) and Soft-DT (Frosst & Hinton, 2017) (tuning in Appendix E.1). LICORICE (Ye et al., 2025), retrained with the authors' code, is a concept-bottleneck policy learned from scratch and serves as a reference that uses concept annotations. Appendices I and E give all settings; the audits of Section 5 use separate held-out teacher rollouts.

Fidelity and size. The soft surrogate reproduces 99.8–100% of the teacher's actions and the hard rules 96.8–100% (Table 1). The rules match the teacher's return on both MiniGrid tasks, Pong and Boxing and reach 88% of it on PixelCartPole, where one wrong action late in an episode ends it.

Table 2: Structure of the extracted hard rules on LUCID's validation split (mean $\pm$ s.d. over three training seeds). *Coverage*: states on which at least one non-universal clause fires. *Conflict*: states on which clauses of two or more actions fire. *Fidelity*: agreement with the teacher when universal clauses are removed and uncovered states take the fallback action. Δ_{unw}: fidelity lost when clause weights are replaced by clause counts (ties $\rightarrow$ fallback). Coverage is low on DoorKey by design, not by omission: the teacher selects `Forward` on $\approx$53% of states, and the rules fire almost exactly on the complementary non-`Forward` states (turning, picking up the key, opening the door), leaving `Forward` as the default action. The other environments have no such dominant action, so their rules cover nearly all states.

Environment	Coverage (%)	Conflict (%)	Fidelity (%)	Δ_{unw} (%)
DoorKey-6x6	46.5 ± 0.5	0.0	99.85 ± 0.07	0.00
Dynamic-Obs-5x5	100.0 ± 0.0	0.0	99.99 ± 0.01	0.00
PixelCartPole	95.1 ± 1.6	0.4 ± 0.5	97.45 ± 0.96	0.00
Pong	99.8 ± 0.1	5.3 ± 0.2	99.78 ± 0.11	0.04 ± 0.03
Boxing	99.1 ± 0.4	18.4 ± 4.2	98.40 ± 0.25	0.14 ± 0.08

Table 3: Complete extracted rule set of the seed-42 DoorKey model ($\tau = \theta = 0.66$), with the labels proposed by the namer VLM (run 1) and, in brackets, the blind-test AUC of each label (Appendix H). Rules fire when all their concepts are ON; the action with the largest summed clause weight wins. Universal clauses give `Forward` a constant prior and `Drop` a prior of $\sigma(-14.6) \approx 0$. –: concept ON in fewer than 1.3% of states, not labelled.

Action	Rule (concept: VLM label [AUC])
`TurnLeft`	f_{23}: key or door in agent row [0.70] $\wedge$ f_{92}: door visible and key not on floor in front or right [0.64] $\wedge$ f_{237}: key adjacent left or front, separated from door [0.68] $\wedge$ f_{294}: clear path to closed door without key [0.46]
`TurnRight`	f_{87}: yellow object immediately right [0.84] $\wedge$ f_{170}: yellow object immediately right [0.68]
`TurnRight`	$f_{87} \wedge f_{170} \wedge f_{258}$: yellow object right of agent in same row [0.90] (subsumed)
`Pickup`	f_{54}: key directly in front of agent [0.68] $\wedge$ f_{131}: key directly in front [0.88]
`Toggle`	$f_{129} \wedge f_{182} \wedge f_{262}$ (–); $f_{129} \wedge f_{182}$ (–)
`Forward`	universal prior
`Drop`	universal prior ≈ 0 (never selected)

LUCID needs fewer feature tests than the two decision-tree extractors at equal or higher fidelity in four of five environments, most clearly on Atari, where VIPER wins Pong and scores 90.1 on Boxing while reproducing only 63–69% of the teacher's actions, i.e. it reaches a high return with a policy that differs from the teacher's. On PixelCartPole VIPER is smaller (34 vs. 45 tests) but less faithful. The rules are also structurally simple: conflicts between actions are absent on MiniGrid, and replacing clause weights by clause counts changes fidelity by at most 0.26 points in any environment and seed, so the DNF behaves as an unweighted rule set with a default action (Table 2). Coverage is low on DoorKey by design: the teacher moves forward in about half of all states, and the rules fire almost exactly on the complementary states, leaving `Forward` to the universal-clause prior. Table 3 shows the complete rule set of the seed-42 DoorKey model. The `Pickup` rule is a conjunction of two concepts that the namer independently labelled "key directly in front" and that detect the simulator predicate `key_in_front` with AUC 0.98 and 1.00, which is exactly MiniGrid's pickup condition; the prior of `Drop`, an action the teacher never takes, is driven to zero. The rule set also shows the limits of reading rules at face value: the second `TurnRight` clause is subsumed by the first, and the three concepts of the `Toggle` rule are active in fewer than 1.3% of states and could not be labelled.

What makes the rules faithful. The baselines of Table 1 differ from LUCID in both representation and rule learner, so they cannot say which component matters. Four controls share LUCID's teacher, data split and hyperparameters (Table 4); complexity is counted uniformly as feature tests (literals for DNF, internal nodes for trees, non-zero coefficients for linear models), the convention most favourable to trees. A reconstruction-only SAE frozen before training the same bottleneck and DNF layer (*two-stage*), and the DNF layer applied directly to CNN features (*no SAE*), both imitate the teacher almost perfectly in soft form (92.8–100%) but lose 14–47 points once their rules are extracted, and on Atari their selectors never polarize, so extraction yields thousands of literals where LUCID has 15–99. Joint training is therefore what makes extraction nearly lossless. Against trees and sparse

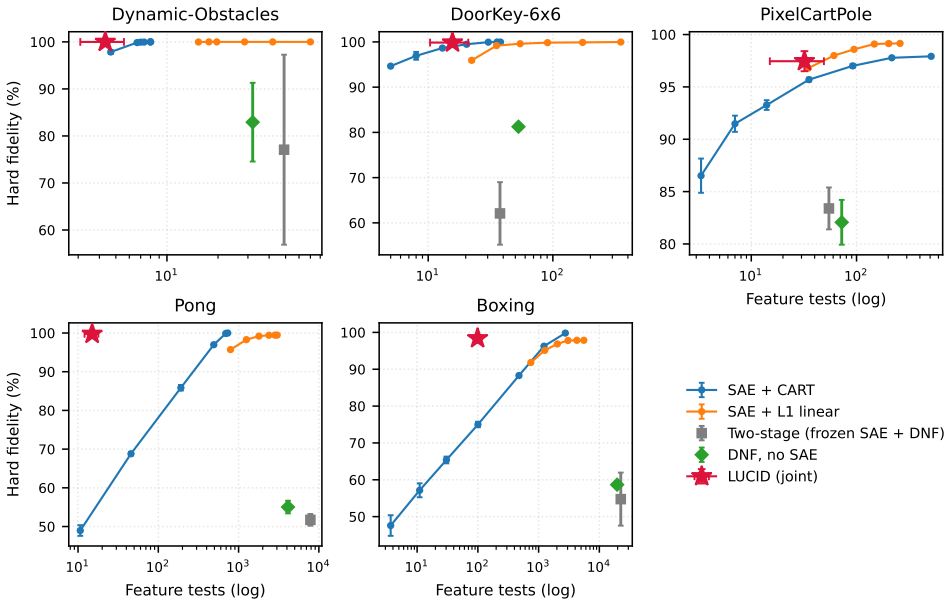

Figure 2: Fidelity–complexity trade-off on LUCID's validation split (mean ± s.d. over three seeds). Lines sweep a tree or sparse linear model on reconstruction-only SAE concepts over its complexity knob; points are DNF controls at LUCID's hyperparameters. A tree needs about 2× more tests than LUCID to match its fidelity on gridworlds and 28–46× more on Atari.

Table 4: Controlled ablations on LUCID's validation split (mean ± s.d. over three training seeds). All controls use the same teacher and LUCID's per-environment hyperparameters. Hard fidelity is agreement with the teacher using non-universal clauses plus the fallback action. *@budget*: best fidelity over the sweep among models with no more feature tests than LUCID (same seed); "–" means no model in the sweep was that small. *Tests to match*: fewest feature tests reaching LUCID's fidelity (same seed); n.r. = never reached. †One of three seeds never reached it; mean over the other two.

	Method	Dynamic-Obs	DoorKey	PixelCartPole	Pong	Boxing
Hard fid. (%)	**LUCID (joint)**	$\mathbf{99.99 \pm 0.01}$	$\mathbf{99.85 \pm 0.07}$	$\mathbf{97.45 \pm 0.96}$	$\mathbf{99.78 \pm 0.10}$	$\mathbf{98.39 \pm 0.24}$
	Two-stage	77.08 ± 20.20	62.09 ± 6.91	83.39 ± 2.00	51.73 ± 1.52	54.75 ± 7.20
	No SAE	82.93 ± 8.37	81.26 ± 0.72	82.07 ± 2.14	55.04 ± 1.62	58.67 ± 0.86
Soft fid. (%)	Two-stage	100.00 ± 0.00	99.65 ± 0.14	97.35 ± 0.25	98.36 ± 0.21	92.84 ± 0.23
	No SAE	100.00 ± 0.00	99.95 ± 0.02	99.07 ± 0.11	99.04 ± 0.04	95.52 ± 0.08
Literals	**LUCID (joint)**	$\mathbf{4.3 \pm 1.2}$	$\mathbf{15.7 \pm 5.3}$	$\mathbf{32.0 \pm 17.0}$	$\mathbf{15.0 \pm 2.9}$	$\mathbf{99.0 \pm 5.9}$
	Two-stage	49.0 ± 12.2	37.7 ± 9.3	54.7 ± 29.0	7841 ± 96	22530 ± 603
	No SAE	32.0 ± 8.5	53.0 ± 7.8	72.3 ± 4.8	4126 ± 120	19624 ± 1194
SAE+CART	Fid. @budget (%)	97.75 ± 0.49	98.67 ± 0.51	93.96 ± 2.52	48.97 ± 1.38	71.88 ± 3.68
	Tests to match	7.7 ± 1.2	30.3 ± 2.1	$297 \pm 201^{\dagger}$	692 ± 27	2754 ± 64
SAE+L1	Fid. @budget (%)	–	96.41	96.90 ± 0.06	–	–
	Tests to match	16.7 ± 4.5	127 ± 52	63.3 ± 27.0	n.r.	n.r.

linear models on reconstruction-only SAE concepts (Figure 2), the advantage is small on gridworlds and large on Atari: at LUCID's size, CART is within 1–2.5 points of LUCID on MiniGrid but reaches only 49% on Pong and 72% on Boxing, and an L1-regularized linear model never reaches LUCID's fidelity on either Atari game. LUCID's checkpoint is selected on the split used here while the controls select on their own validation split, so the two close comparisons (CART on DoorKey, linear on PixelCartPole) should be read as ties. Across three training seeds, hard fidelity varies by at most two points in every environment, but rule size varies more (9–22 literals on DoorKey and 8–45 on PixelCartPole, LUCID's least stable case), so a single extracted rule set is one faithful description of the teacher rather than the unique one (Appendix F).

Table 5: Empirical check of Theorem 1 on independent teacher rollouts (N states), disjoint from the training and test data of Table 1 ($\tau = \theta = 0.66$). Measured hard error can therefore differ slightly from 100− the hard fidelity of Table 1. *Clean*: rule-concept activations with $\min(c_j, 1 - c_j) < 0.05$. γ, Δ_{total}: medians over states. *Cert.*: $\Pr[x \in \mathcal{X}_{\text{cert}}]$ under condition (25). *Soft≠hard*: measured disagreement and its bound 1−Cert. (Eq. 26). *Hard error*: measured $1 - \text{AF}_{\text{hard}}$ and the triangle bound of Eq. (27).

Environment	N	**Clean**	γ	Δ_{total}	**Cert.**	**Soft≠hard (%)** measured / bound	**Hard error (%)** measured / bound
Dynamic-Obs-5x5	23,921	100.00%	9.93	0.68	100.00%	0.000 / 0.000	0.000 / 0.000
DoorKey-6x6	25,253	99.19%	8.31	0.30	99.92%	0.075 / 0.079	0.087 / 0.091
PixelCartPole	99,641	70.95%	11.46	4.99	87.55%	3.12 / 12.45	3.00 / 12.84
Pong	165,891	99.52%	9.93	1.24	99.15%	0.16 / 0.86	0.15 / 0.96
Boxing	119,611	99.61%	8.03	6.57	68.95%	3.17 / 31.05	3.23 / 31.27

Certificate and threshold. We evaluate Theorem 1 in double precision on independent teacher rollouts, collected with a different environment seed and never used for training or model selection (Table 5). No state violates the theorem or its lemmas. The theorem certifies 99.1–100% of states on both MiniGrid tasks and Pong, where its bound exceeds the measured soft–hard disagreement by at most 0.7 points, but only 87.6% on PixelCartPole and 69.0% on Boxing, where about 3% of states disagree (Appendix E.5). The slack comes from false clauses: on PixelCartPole only 71% of rule-concept activations are near-binary, and on Boxing small per-clause bounds accumulate over 18 actions: the false clauses of the worst rival action sum to 6.7 on average, against a mean margin of 6.2. The triangle bound inherits this slack, so it is informative only where the certificate is tight (DoorKey: 0.091% bound vs. 0.087% measured hard error). Replacing the state-dependent terms by uniform worst-case ones loosens the bound to 15–97% disagreement, which is why we report the state-dependent form. Changing θ within $[0.5, 0.9]$ moves hard fidelity by at most 1.6 points (Appendix F).

5 Auditing the Concepts

The rules above are faithful and compact. We now ask what their atoms are, with two evaluations on held-out teacher rollouts, three training seeds each. Alignment is measured on the whole dictionary, since it asks what the learned concepts encode; causal efficacy is measured on the rule concepts, since it asks what drives each decision.

Do concepts name simulator factors? We extract *factors* from the simulator: predicates for MiniGrid (`has_key`, `door_open`, `wall_ahead`, ...), the four state variables for PixelCartPole, and object coordinates from OCAtari (Delfosse et al., 2023a) for Pong and Boxing, binned by tertiles and sign. For each dictionary dimension we compute the AUC of its activation as a detector of each factor and call it *monosemantic* (G1) if its best AUC is at least 0.75 and exceeds the second best by 0.10, and *unaligned* (G3) if no AUC reaches 0.65; thresholds were fixed before the analysis. We compare against the teacher's own CNN units, a random projection and PCA, and also test the second-best AUC of aligned dimensions with a one-sided Mann–Whitney test (CLES, lower is cleaner). Shuffling the factor labels gives a mean AUC of 0.500 in every environment.

Do concepts drive the decision? Given a held-out state we edit concepts, decode the edit into the teacher's feature space, $\hat{h} = h + W_d(z' - z) \odot \sigma_{\text{feat}}$, re-run the frozen PPO head, and record how often its arg-max action changes. *Necessity* switches off the literals of the winning clause; *sufficiency* switches on the literals of a clause of another action and asks whether the teacher moves to that action. Effects are differences from a paired baseline that applies the same Δz magnitudes to decoder directions of random live concepts outside the winning clauses, with 95% cluster-bootstrap CIs over episodes. We call concepts *causally effective* when the clause-level necessity effect is significantly positive in all three seeds.

Results. Table 6 summarizes both audits. *Alignment.* On both MiniGrid tasks the dictionary is less polysemantic than the teacher's own units: it roughly triples the monosemantic fraction, and its

Table 6: The two audits on the same dictionaries (three seeds). *G1*: monosemantic dimensions (%), LUCID vs. the teacher's CNN units. *CLES*: Mann–Whitney effect size of LUCID vs. CNN on the second-best factor AUC (lower is cleaner). *Necessity* (clause level) and *Sufficiency*: change in the teacher's action relative to the paired baseline, mean ± s.d.; *CI>0*: seeds with a significantly positive necessity effect. *Routed*: decisions made by at least one concept literal rather than a universal-clause prior (seed 42). *Aligned*: higher G1 than the CNN and a significantly cleaner second-best AUC in every seed; *Causal*: CI>0 in all three seeds. Details in Appendices E.6 and E.7.

	Factor alignment		**Causal efficacy**				**Verdict**	
Environment	G1 (%)	CLES	Necessity	CI>0	Sufficiency	Routed	Aligned	Causal
Dynamic-Obs.	30.5 vs. 11.9	0.26–0.33	-0.006 ± 0.024	1/3	-0.023 ± 0.010	100%	yes	**no**
DoorKey	26.1 vs. 7.8	0.26–0.27	$+0.038 \pm 0.034$	2/3	≈ 0	47.2%	yes	no (2/3)
PixelCartPole	9.0 vs. 46.2	n.s.	$+0.422 \pm 0.177$	3/3	$+0.562 \pm 0.107$	92.2%	**no**	yes
Pong	1.1 vs. 4.8	0.14–0.24	$+0.040 \pm 0.020$	3/3	$+0.159 \pm 0.031$	99.9%	**no**	yes
Boxing	2.3 vs. 7.8	0.21–0.23	$+0.124 \pm 0.081$	3/3	$+0.049 \pm 0.043$	97.0%	**no**	yes

aligned dimensions have cleaner second-best AUCs in every seed ($p \leq 3.2 \times 10^{-4}$). The advantage does not transfer: on PixelCartPole the CNN is far more aligned, and on Atari 84–92% of LUCID dimensions align with no factor, although the few that do are cleaner than CNN units. Two further results bound the claim: LUCID does not beat random projections or PCA on second-best AUC in any environment, and its completeness (best AUC per factor) is always below the CNN's, a coverage-for-purity trade-off of the sparse dictionary. *Causal efficacy.* Editing rule concepts moves the teacher in three environments. PixelCartPole is the clearest case: switching off a winning clause changes the teacher's action 42 points more often than the baseline, although the concept edit displaces the features *less* than the baseline edit ($\|\Delta\hat{h}\|$ 2.90 vs. 4.16), and sufficiency shows a dose–response on PixelCartPole and Pong. The criterion measures consistency rather than size: the clause-level effect on Pong ($+0.040$) is as small as on DoorKey ($+0.038$) but significant in every seed. On DoorKey the effect is unstable and sufficiency is null. On Dynamic-Obstacles it tends to reverse: necessity is significantly negative at all three levels in two seeds (and significantly positive in the third), and sufficiency is significantly negative in all three, although the rules reproduce the teacher with fidelity 0.999–1.000 and every decision is routed through a concept literal.

Sanity checks and where decisions come from. The negative result on Dynamic-Obstacles is not a data artefact: on those runs the teacher's recorded actions match its recomputed actions exactly and the shuffled-factor check passes. Because decoded edits must remain plausible activations, $\hat{h}$ is clamped to the non-negative orthant, and the pre-clamp negative fraction (0.32–0.37) is reported with every result; an isotropic Gaussian baseline of matched norm is also computed. Finally, rules can decide through the universal-clause prior rather than through concepts. In four environments 92–100% of decisions are made by at least one concept literal; DoorKey is the exception (47.2%), where the prior encodes "move forward when nothing is detected", agrees with the teacher on 99.8% of the states it decides and suppresses the useless `drop` action. Elsewhere the prior is a failure mode, agreeing with the teacher on only 4.9% (Boxing), 30.5% (Pong) and 61.9% (PixelCartPole) of the states it decides.

The two axes diverge. In no environment is the dictionary aligned with our factors while its rule concepts are causally effective: where they name factors best (MiniGrid) intervention does least, and where intervention works (PixelCartPole, Atari) most concepts name no factor. We report this as an observation, not a law. Five environments cannot establish a trade-off, the factor vocabularies differ in kind and may simply describe gridworlds better than pixel control, the number of actions (2 on PixelCartPole, 18 on Boxing) mechanically changes how easily an edit flips the action, and all dictionaries come from one method whose objective shapes concepts for imitation rather than for human-specified factors, so the divergence may partly be specific to LUCID. What the divergence does establish is methodological: fidelity of 96.8–100% constrains neither axis, so neither fidelity nor either audit alone licenses the claim that a surrogate's atoms are the policy's reasons.

Can VLM labels be trusted? Readable rules need names. A namer VLM sees 8 ON and 8 OFF states of a concept from the agent's own view and proposes a label, without rules, actions

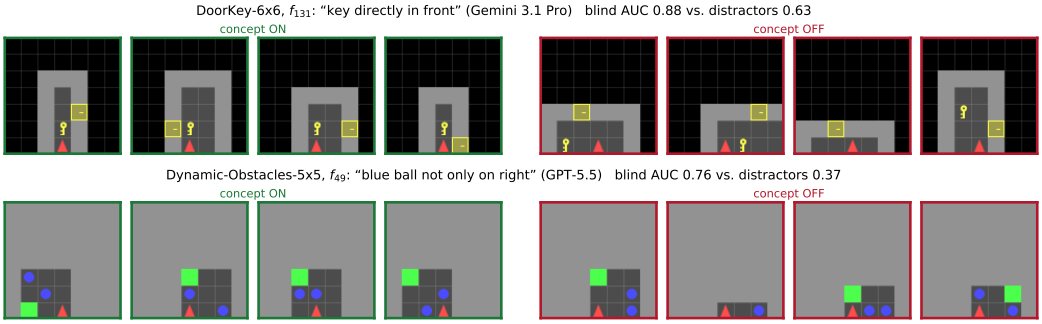

Figure 3: Input and output of the namer for two concepts: four of the eight ON and four of the eight OFF views it received (agent's egocentric view; no heatmap, rules or actions), the label it proposed, and the label's blind-test AUC on held-out states against the mean AUC of three distractor labels.

or attribution maps; a scorer from a different developer (`deepseek-flash`) then sees only the label and one held-out state and gives the probability that it holds. We compare these probabilities with the concept's activation (ROC AUC), against labels of other concepts as distractors and against a positive control of ground-truth factor statements that bounds what this scorer can reach. With `gemini-3.1-pro-preview` naming DoorKey and `gpt-5.5` naming Dynamic-Obstacles, true labels beat distractors in both: AUC 0.72 vs. 0.52 ($p = 0.002$) and 0.69 vs. 0.43 ($p = 0.017$; combined $p < 0.001$), against scorer ceilings of 0.80 and 0.79. The concepts most tightly aligned with a factor receive labels naming it (f_{131} and f_{54}, AUC ≥ 0.98 for `key_in_front`, are labelled "key directly in front"), but a label can also miss the factor its concept tracks. Labels are therefore useful hypotheses whose reliability can be measured, not verified semantics (protocol, prompts and all labels in Appendix H). The positive control also shows why the scorer matters: an open-weight scorer (Qwen3-VL-8B) reached AUC 0.36–0.54 on three of the five DoorKey factor statements, and with it the labels of an open-weight namer were no more predictive than distractors ($p = 0.35$).

6 Conclusion and Limitations

LUCID extracts compact Boolean rules from vision-based RL policies that agree with the teacher on 96.8–100% of states, with a threshold-invariant extraction and a checkable soft–hard certificate, and controls show that joint training is what keeps extraction faithful. Auditing the concepts of these rules gives a different picture: alignment with human-specified factors and causal efficacy on the teacher never coincide in our five environments, and a rule set with near-perfect fidelity can rest on counterproductive concepts. LUCID is also what makes this audit possible: its decoder turns a concept edit into a feature edit on the frozen teacher, which a tree over raw CNN features does not allow. Behavioural fidelity, factor alignment and causal efficacy are three separate properties, and we recommend that rule-extraction work report all three: fidelity on held-out states together with online return, since per-step fidelity does not guarantee long-horizon behaviour; alignment against simple baselines such as random projections and PCA, not only against the raw network; causal effects against a norm-matched perturbation of the same dictionary; and, if concepts are named automatically, a blind predictive test of the names with a positive control for the evaluator.

Limitations. LUCID requires discrete actions; our attempt at continuous control failed both at obtaining a pixel-based teacher and because discretized joint action spaces grow too fast for a per-action DNF layer (Appendix G). Five environments and three identified confounds limit the divergence claim, and the negative effect on Dynamic-Obstacles remains unexplained. Outside MiniGrid the factor vocabulary may be too coarse to separate a misaligned dictionary from a poor vocabulary. The certificate is loose where concepts are weakly binarized or many clauses compete, and is evaluated only on states visited by the teacher. VLM labels were tested on two environments with one scorer and no human study.

AI USE STATEMENT

Large language models were used in three ways. (i) As experimental instruments: `gemini-3.1-pro-preview` and `gpt-5.5` proposed concept labels and `deepseek-flash` scored them in the blind evaluation of Section 5; models, prompts and settings are given in Appendix H. (ii) As coding assistants, to help write the evaluation scripts of the VLM labelling protocol. (iii) As writing assistants, for editing grammar and wording and for help restructuring the manuscript. All research ideas, experimental design choices, results and claims were made and verified by the authors, who take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

All settings of LUCID are listed in Appendix I and those of the baselines in Appendix E.1. Complete proofs of Theorem 1, Proposition 1 and Lemma 4 are given in Appendix A. The two concept audits are specified in Appendices E.6 and E.7, and the VLM labelling protocol, including prompts, model versions and statistics, in Appendix H. Anonymized code for training, rule extraction, the certificate check and both audits, together with the raw VLM responses, is provided as supplementary material.

REFERENCES

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. Verifiable reinforcement learning via policy extraction. In *Advances in Neural Information Processing Systems*, 2018.

Kexin Gu Baugh, Luke Dickens, and Alessandra Russo. Neural DNF-MT: A neuro-symbolic approach for learning interpretable and editable policies. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2025.

Yanzhe Bekkemoen and Helge Langseth. Interpretable deep reinforcement learning via concept-based policy distillation. *Machine Learning*, 2025. doi: 10.1007/s10994-025-06928-5.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Leo Breiman, Jerome Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL `https://transformer-circuits.pub/2023/monosemantic-features`.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for OpenAI Gym. `https://github.com/Farama-Foundation/Minigrid`, 2018.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Quentin Delfosse, Jannis Blüml, Bjarne Gregori, Sebastian Sztwiertnia, and Kristian Kersting. OCAtari: Object-centric Atari 2600 reinforcement learning environments. *arXiv preprint arXiv:2306.08649*, 2023a.

Quentin Delfosse, Hikaru Shindo, Devendra Dhami, and Kristian Kersting. Interpretable and explainable logical policies via neurally guided symbolic abstraction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

R. Dutta, Q. Wang, A. Singh, D. Kumarjiguda, X. Li, and S. Jayavelu. S-REINFORCE: A neuro-symbolic policy gradient approach for interpretable reinforcement learning. *arXiv preprint arXiv:2305.07367*, 2023.

Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

Peter Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Jana Hüls, Jan-Ole Perschewski, and Sebastian Stober. INAM: Image-scale neural additive models. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 249–254, 2025.

Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117, 2017.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Mikel Landajuela, Brenden K. Petersen, Sookyung Kim, Claudio P. Santiago, Ruben Glatt, T. Nathan Mundhenk, Jacob F. Pettit, and Daniel Faissol. Discovering symbolic policies with deep reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5979–5989, 2021.

Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *International Conference on Learning Representations*, 2025.

Guiliang Liu, Xiangyu Sun, Oliver Schulte, and Pascal Poupart. Learning tree interpretation from object representation for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Ignacio D Lopez-Miguel, Ezio Bartocci, Thomas Eiter, and Martin Tappler. ORCAID: Oblique rule-based continuous-action interpretation for deep RL policies. *arXiv preprint arXiv:2607.07235*, 2026.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Hugo M Proença and Matthijs van Leeuwen. Interpretable multiclass classification by mdl-based rule lists. *Information Sciences*, 512:1372–1393, 2020.

Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In *European Conference on Computer Vision (ECCV)*, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop (ITW)*, pp. 1–5, 2015.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Zhuorui Ye, Stephanie Milani, Geoffrey J. Gordon, and Fei Fang. LICORICE: Label-efficient concept-based interpretable reinforcement learning. In *International Conference on Learning Representations*, 2025.

Nicholas Zolman, Christian Lagemann, Urban Fasel, J. Nathan Kutz, and Steven L. Brunton. SINDy-RL for interpretable and efficient model-based reinforcement learning. *Nature Communications*, 16:10714, 2025.

A Proofs and Supporting Results

In this appendix, we provide the formal definitions, supporting lemmas, and the complete proof for the Soft-Hard Fidelity guarantee (Theorem 1) stated in the main text.

A.1 Formal Definitions

Definition 1 (LUCID Policy)**.** *A* LUCID policy *over D concepts and A actions is specified, for each action $a \in [A]$, by n_a clauses. Clause $C_{a,i}$ has literal sets $P_{a,i}, N_{a,i} \subseteq [D]$ with $P_{a,i} \cap N_{a,i} = \emptyset$ (guaranteed for $\tau \geq 1/2$) and a clipped bias $\tilde{b}_{a,i} = \min(b_{a,i}, b_{\max})$. Given $c^{\mathrm{hard}}(x) = \mathbf{1}[c^{\mathrm{soft}}(x) > \theta] \in \{0,1\}^D$,*

$$C^{\mathrm{hard}}_{a,i}(x) \;=\; \sigma(\tilde{b}_{a,i}) \cdot \prod_{j \in P_{a,i}} c^{\mathrm{hard}}_j(x) \cdot \prod_{j \in N_{a,i}} \big(1 - c^{\mathrm{hard}}_j(x)\big), \tag{5}$$

with the empty product equal to 1. The discrete policy is $\pi^{\mathrm{hard}}(x) = \arg\max_a \sum_i C^{\mathrm{hard}}_{a,i}(x)$ on $\mathcal{X}_{\mathrm{trig}}$ and a_{fallback} otherwise.

Definition 2 (State-Dependent Literal Evaluation)**.** *The soft literal is $\ell^{\mathrm{soft}}_{a,i,j}(x) = p_{a,i,j} c^{\mathrm{soft}}_j(x) + n_{a,i,j}(1 - c^{\mathrm{soft}}_j(x)) + u_{a,i,j} \in [0,1]$, where $\mathbf{v}_{a,i,j} = [p, n, u]$ are the selector probabilities. The hard literal $\ell^{\mathrm{hard}}_{a,i,j}(x) \in \{0,1\}$ is defined for $j \in P_{a,i} \cup N_{a,i}$ as in Section D. The action margin is $\gamma(x) = s^{\mathrm{hard}}_{a^*}(x) - \max_{a \neq a^*} s^{\mathrm{hard}}_a(x)$.*

A.2 Supporting Lemmas

Lemma 1 (Per-Clause Error Bound)**.** *For any clause i of action a at state x, $|C^{\mathrm{soft}}_{a,i}(x) - C^{\mathrm{hard}}_{a,i}(x)| \leq \Delta_{a,i}(x)$, with $\Delta_{a,i}$ as in Eq. (24). The bound holds with equality for true clauses.*

Proof. Recall $C^{\mathrm{soft}}_{a,i} = \sigma(\tilde{b}_{a,i} + L_{a,i})$ with $L_{a,i} = \sum_{j=1}^{D} \log(\ell^{\mathrm{soft}}_{a,i,j} + \epsilon)$.

Case 1: True clause ($T_{a,i}(x) = \mathrm{True}$). Then $C^{\mathrm{hard}}_{a,i} = \sigma(\tilde{b}_{a,i})$, and the divergence is exactly $|\sigma(\tilde{b}_{a,i}) - \sigma(\tilde{b}_{a,i} + L_{a,i}(x))|$; the sum runs over all D concepts, capturing the log-drag of ignored literals.

Case 2: False clause ($T_{a,i}(x) = \mathrm{False}$). Then $C^{\mathrm{hard}}_{a,i} = 0$ and $V_{a,i}(x) \neq \emptyset$: either some literal is violated by $c^{\mathrm{hard}}(x)$, or the clause is contradictory (convention (i)) and every $j \in P_{a,i} \cap N_{a,i}$ lies in $V_{a,i}(x)$. The argument therefore covers contradictory clauses and every $\tau \in [0,1]$. The divergence equals $C^{\mathrm{soft}}_{a,i}(x)$. Since $\ell^{\mathrm{soft}} \leq 1$, every term satisfies $\log(\ell^{\mathrm{soft}}_{a,i,j} + \epsilon) \leq \log(1+\epsilon)$. Keeping only the term of $j^* \in V_{a,i}(x)$,

$$L_{a,i}(x) \leq \log\big(\ell^{\mathrm{soft}}_{a,i,j^*}(x) + \epsilon\big) + (D-1)\log(1+\epsilon).$$

Since σ is increasing, $C^{\mathrm{soft}}_{a,i}(x) \leq \Delta_{a,i}(x)$. □

Lemma 2 (Action Score Perturbation)**.** *For every action a, $|s^{\mathrm{soft}}_a(x) - s^{\mathrm{hard}}_a(x)| \leq \sum_{i=1}^{n_a} \Delta_{a,i}(x) = \Delta_a(x)$.*

Proof. Apply the triangle inequality to $s_a = \sum_{i=1}^{n_a} C_{a,i}$ and use Lemma 1. □

Lemma 3 (Structural Bound on Violated Literals)**.** *Let τ be the selector threshold and θ the concept threshold. If $j \in P_{a,i}$ is violated, then $\ell^{\text{soft}}_{a,i,j}(x) \le 1 - p_{a,i,j}(1-\theta) < 1 - \tau(1-\theta)$. If $j \in N_{a,i}$ is violated, then $\ell^{\text{soft}}_{a,i,j}(x) \le 1 - n_{a,i,j}\,\theta < 1 - \tau\theta$.*

Proof. If $j \in P_{a,i}$ is violated then $c_j^{\text{soft}} \le \theta$, so $\ell = pc + n(1-c) + u \le p\theta + (n+u) = 1 - p(1-\theta)$, and $p > \tau$. If $j \in N_{a,i}$ is violated then $c_j^{\text{soft}} > \theta$, so $\ell \le n(1-\theta) + (p+u) = 1 - n\theta$, and $n > \tau$. □

Lemma 3 shows that the false-clause bound is data-independent only up to a factor that depends on θ: with $\tau = \theta = 0.66$, a violated positive literal is only guaranteed $\ell^{\text{soft}} < 0.776$, while a violated negative literal is guaranteed $\ell^{\text{soft}} < 0.565$. The worst-case gaps are balanced at $\theta = 1/2$; for $\theta > 1/2$, suppressing false clauses with violated positive literals relies on bimodality ($c_j^{\text{soft}} \to 0$, giving $\ell^{\text{soft}} \to n + u = 1 - p$), which is why we report the state-dependent bound rather than this worst case.

A.3 Proof of Theorem 1

Proof. Let $x \in \mathcal{X}_{\text{trig}}$ with $\gamma(x) > 0$, so $a^* = \arg\max_a s_a^{\text{hard}}(x)$ is unique. For any $a \neq a^*$, Lemma 2 gives

$$s_{a^*}^{\text{soft}} - s_a^{\text{soft}} \;\ge\; \big(s_{a^*}^{\text{hard}} - s_a^{\text{hard}}\big) - \Delta_{a^*}(x) - \Delta_a(x). \tag{6}$$

Under (25) the right-hand side is strictly positive for every $a \neq a^*$, so a^* is the unique maximizer of s^{soft} and $\pi^{\text{soft}}(x) = \pi^{\text{hard}}(x)$ regardless of tie-breaking. Since $s_{a^*}^{\text{hard}} - s_a^{\text{hard}} \ge \gamma(x)$ and $\Delta_a(x) \le \max_{a' \neq a^*} \Delta_{a'}(x)$, the condition $\Delta_{\text{total}}(x) < \gamma(x)$ implies (25). Finally, $\{\pi^{\text{soft}} \neq \pi^{\text{hard}}\} \subseteq \mathcal{D} \setminus \mathcal{X}_{\text{cert}}$, which yields Eq. (26). □

A.4 Theoretical Insights and Loss Alignment

Remark 1 (The Necessity of Exact State-Dependent Bounds)**.** *Global worst-case bounds (e.g., a uniform error $\bar{\varepsilon}$ across all concepts, or the structural bound of Lemma 3) ignore the state-dependent sparsity of the SAE and the polarization of the bottleneck. By using $\ell^{\text{soft}}_{a,i,j}(x)$ directly, Lemma 1 isolates the true perturbations without artificial inflation.*

Empirically, replacing every $\varepsilon_j(x)$ by a uniform cleanliness level $\bar{\varepsilon}$ (best value over a grid) and every violated literal by Lemma 3 bounds $\Pr[\pi^{\text{soft}} \neq \pi^{\text{hard}}]$ by 75% on DoorKey, 97% on PixelCartPole, 15% on Pong and 60% on Boxing, against 0.08%, 12.5%, 0.86% and 31% for Theorem 1; only on Dynamic-Obstacles, where every rule concept is clean, are the two equally tight.

Remark 2 (Categorical Entropy Minimizes Selector Softness)**.** *For a true clause with $L_{a,i}(x) \le 0$, since $\sigma(b + L) \ge e^L \sigma(b)$,*

$$\Delta_{a,i}(x) \le \sigma(\tilde{b}_{a,i})\big(1 - e^{L_{a,i}(x)}\big) \le \sigma(\tilde{b}_{a,i})\,|L_{a,i}(x)|,$$
$$|L_{a,i}(x)| \le \sum_{j \in P_{a,i} \cup N_{a,i}} -\log\big(\ell^{\text{soft}}_{a,i,j} + \epsilon\big) + \sum_{j \notin P_{a,i} \cup N_{a,i}} -\log\big(u_{a,i,j} + \epsilon\big),$$

using $\ell^{\text{soft}}_{a,i,j} \ge u_{a,i,j}$. The second sum is the structural drag of the $D - |P_{a,i} \cup N_{a,i}|$ ignored literals; it is large unless $u \to 1$ (e.g., at initialization $u \approx 0.91$, so $-\log u \approx 0.095$ per literal, i.e., ≈ 28 over $D = 300$). The categorical-entropy penalty $\mathcal{L}_{\text{ent}}$ (together with $\mathcal{L}_{\text{L0}}$) drives $u \to 1$ for ignored literals and removes this drag.

At convergence the structural drag has vanished in all five environments (below 10^{-5} per true clause), and the drag of the selected literals averages 0.001–0.027 per true clause, so true clauses contribute almost nothing to Δ_{total}; the residual slack of Theorem 1 comes from false clauses (Section E.5).

Remark 3 (Bimodality Loss Minimizes Concept Softness)**.** $\mathcal{L}_{\text{bim}}$ *drives $c_j^{\text{soft}} \to \{0, 1\}$. For satisfied active literals this gives $\ell^{\text{soft}} \to p + u$ or $n + u$, which tends to 1 once selectors are sharp, shrinking the first sum above; for violated active literals it gives $\ell^{\text{soft}} \to 1 - p$ or $1 - n$, which tends to 0, shrinking $\Delta_{a,i}$ of false clauses. Together with $\mathcal{L}_{\text{ent}}$, it aligns training with tightening Theorem 1.*

Remark 4 (Role of Bias Clipping)**.** *Because $\tilde{b}_{a,i} \le b_{\max}$, every hard clause contributes at most $\sigma(b_{\max}) \approx 0.9933$, and the clause cannot overcome the log-drag by inflating its bias: a false clause stays below $\sigma(b_{\max} + \log(\ell_{j^*} + \epsilon))$. Theorem 1 is stated with $\tilde{b}$ because both the soft forward pass and the extracted rules use the clipped value. Likewise, a universal* `True` *clause (convention (ii)) is a true clause with $P_{a,i} \cup N_{a,i} = \emptyset$, so its error is the pure structural drag $|\sigma(\tilde{b}_{a,i}) - \sigma(\tilde{b}_{a,i} + L_{a,i}(x))|$, which is small only when all $u_{a,i,j} \to 1$.*

Remark 5 (Margin Loss to Actively Optimize Fidelity)**.** *A natural extension, not used in our experiments, is to add the hinge penalty $\mathcal{L}_{\text{margin}} = \sum_x \max\big(0,\ \Delta_{\text{total}}(x) - \gamma(x) + \xi\big)$ to Eq. (3), directly penalizing states in the uncertified zone.*

A.5 Proofs for the Optimization-Based Extraction Framework

This subsection provides the formal proofs for Lemma 4 (step-function structure) and Proposition 1 (central-interval invariance and robustness).

A.5.1 Proof of Lemma 4 (Step-Function Structure)

Proof. The description-length term is a sum of indicators

$$\mathcal{L}_{\text{MDL}}(\tau) = \sum_{a,i,j} \big(\mathbf{1}[p_{a,i,j} > \tau] + \mathbf{1}[n_{a,i,j} > \tau]\big),$$

each of which is constant on any open interval of τ that does not contain the corresponding selector probability. Hence $\mathcal{L}_{\text{MDL}}(\tau)$ is piecewise constant on the intervals between consecutive elements of the sorted set $\mathcal{S}$.

For the information-loss term, observe that the literal sets $P_{a,i}(\tau)$ and $N_{a,i}(\tau)$ are themselves fully determined by which inequalities $\{p_{a,i,j} > \tau\}$ and $\{n_{a,i,j} > \tau\}$ hold. On any open interval of τ that avoids all elements of $\mathcal{S}$, none of these inequalities flip, so the rule sets are constant and the induced hard policy π_τ^{hard} is identical for all τ in that interval. Therefore the inner disagreement indicator in Eq. (20) is constant pointwise in x, and so is its expectation $\mathcal{L}_{\text{info}}(\tau)$.

The sum of two piecewise-constant functions sharing the same partition is piecewise constant on that partition, and a piecewise-constant function on a finite partition of $[0, 1]$ attains its minimum on at least one of its constant pieces. The optimum τ^* therefore lies in some open interval between consecutive elements of $\mathcal{S}$ (with ties broken arbitrarily within that interval). □

A.5.2 Proof of Proposition 1 (Central-Interval Invariance & Robustness)

Proof. We treat a selected positive literal ($p > \tau$); the negative case is symmetric with p and n exchanged. Throughout, $p + n + u = 1$.

(a) Majority. If $\tau \ge 1/2$ and $p > \tau$, then $n < 1 - p < 1/2 \le \tau$, so $j \notin N_{a,i}(\tau)$ and the literal sets are disjoint. Moreover $n + u = 1 - p < 1/2 < p$. For any $\tau < 1/2$, the selector $p = \tau + \eta$, $n + u = 1 - \tau - \eta$ with small $\eta > 0$ is selected while $n + u > p$, so no smaller threshold guarantees a majority.

(b) Confidence. The maximum-entropy prior over the three states assigns $1/3$ to each. If $\tau \ge 2/3$ and $p > \tau$, the discarded mass is $n + u = 1 - p < 1 - \tau \le 1/3$. For any $\tau < 2/3$, a selector with $p \in (\tau, 2/3)$ violates this, so $2/3$ is the smallest threshold with this guarantee.

(c) Invariance and robustness. If $\delta < 1/3$, then $[1/2, 2/3] \subset (\delta, 1 - \delta)$. Since no selector value lies in $(\delta, 1 - \delta)$, the literal sets $P_{a,i}(\tau), N_{a,i}(\tau)$ do not change as τ moves within this interval (Lemma 4), so the rule set, and hence π_τ^{hard}, is identical for all $\tau \in [1/2, 2/3]$. In particular every literal selected at $\tau = 1/2$ satisfies $p > 1/2 > \delta$, hence $p \ge 1 - \delta > 2/3$, so (b) holds at $\tau = 1/2$; (a) holds since $\tau = 1/2$. Finally, a perturbation of magnitude η changes the rule set only if it moves some selector across τ, which requires $\eta > \min(\tau - \delta, 1 - \delta - \tau)$. This margin is a concave, piecewise-linear function of τ with a unique maximizer at $\tau = 1/2$, where it equals $1/2 - \delta$. □

B Extended Related Work

B.1 Inherently Interpretable and Neuro-Symbolic RL

One paradigm designs ad-hoc interpretable models from scratch. Neuro-symbolic methods such as Deep Symbolic Policies (Landajuela et al., 2021) and Differentiable Inductive Logic Programming (Evans & Grefenstette, 2018) search over mathematical or logical expressions to natively parameterize the policy, while S-REINFORCE (Dutta et al., 2023) concurrently trains neural and symbolic components and SINDy-RL (Zolman et al., 2025) identifies sparse non-linear dynamics for interpretable world models. These methods provide functional rather than rule-based interpretability, making them ill-suited for domains requiring explicit logical triggers (e.g., "if key is held AND door is reached"). They also restrict the hypothesis space to human-readable symbolic primitives, which often prevents peak performance on complex continuous control tasks and forces discarding state-of-the-art neural architectures. LUCID instead retains the high-performing neural policy and explains it post-hoc.

B.2 Post-Hoc Policy Extraction and Concept Bottlenecks

Post-hoc surrogates attempt to map the decisions of a frozen black-box agent into a transparent format. Decision tree extractors like VIPER (Bastani et al., 2018) and CART (Breiman et al., 1984) use imitation learning to clone the neural policy. While these methods have shown reasonable success when applied to agents operating on low-dimensional, explicitly structured state vectors (typically using Feed-Forward Networks), they scale poorly to vision-based RL. Vision-based agents utilize CNNs to process high-dimensional pixel data, resulting in latent representations that are deeply entangled. Because traditional surrogates evaluate these polysemantic, continuous features using simple binary splits, they struggle to compactly represent multivariate rules. Consequently, they face an accuracy–interpretability trade-off: shallow trees lose accuracy, while deep trees become hard to read.

To ground explanations in human terms, Concept Bottleneck Models (CBMs) (Koh et al., 2020; Ye et al., 2025) have been adapted for RL to route decisions through understandable concepts. However, traditional CBMs rely heavily on manual concept specification by domain experts. This is labor-intensive and incomplete, as it fails to capture emergent, task-critical visual features discovered by the neural agent itself. LUCID instead discovers its concepts without annotation; whether these concepts are human-readable is tested separately (Section 5).

B.3 Mechanistic Interpretability and Sparse Autoencoders

A central challenge in interpreting deep networks is *polysemanticity*, where single neurons fire for multiple, unrelated features. Recently, Sparse Autoencoders (SAEs) have been used successfully in mechanistic interpretability for Large Language Models (Bricken et al., 2023) by disentangling these activations into an overcomplete dictionary of sparse, more interpretable concepts, thereby significantly reducing polysemanticity. Despite this success, the use of SAEs to explain vision-based DRL agents remains underexplored. Dictionary learning has been extended to extract interpretable features from standard vision models (Lim et al., 2025), but not, to our knowledge, to rule extraction from RL policies. Complementing this, additive models like INAM (Hüls et al., 2025) provide feature-level importance for image classifiers. LUCID builds upon this "feature-first" philosophy but goes a step further: while SAE analyses of vision models focus on passive feature discovery, we unify this discovery with an end-to-end logical bottleneck to produce formal DNF rules, moving from feature identification to active rule-based surrogacy. Applying SAEs for explicit rule extraction introduces significant challenges, particularly regarding the continuous-discrete bottleneck required for formal logic. Unlike standard post-hoc approaches that struggle with binarization (e.g., dead gradients from hard thresholding), LUCID utilizes a learnable sigmoid bottleneck.

B.4 Closest Precedents

Several works combine parts of LUCID's pipeline. Bekkemoen & Langseth (2025) discover concepts post hoc in the activations of CNN policies with non-negative matrix factorization and distil the policy into a set of sparse linear models over those concepts; the closest analogue in our setup is the SAE+L1 control of Section E.3, which does not reach LUCID's fidelity on either Atari game at any

sparsity. Discover-then-Name (Rao et al., 2024) discovers concepts with an SAE on CLIP features and names them by their alignment with text embeddings, to build a concept bottleneck for image classification; RL policy features have no such text-aligned space, so we name concepts with a VLM from examples and test the names blind (Section 5). NUDGE (Delfosse et al., 2023b) uses a neural policy to guide the search for first-order logic policies, but operates on object-centric symbolic states produced by an object extractor, whereas LUCID starts from the teacher's pixel features. Neural DNF-MT (Baugh et al., 2025) learns DNF policies end to end with actor-critic methods and converts them into editable logic programs; it replaces the policy instead of explaining a frozen one. RAMi (Liu et al., 2021) is post hoc and pixel-based like LUCID: it learns an identifiable object-centric representation of the observations and then searches compact regression trees that mimic the agent's action values over its latent features. The representation is learned before, and independently of, the tree, which corresponds to the two-stage control of Section E.3; there, freezing the representation before fitting the rules costs 14–47 points of hard fidelity. LUCID also outputs Boolean DNF rules over binary concepts rather than trees over continuous latents, and audits those concepts against simulator factors and interventions.

Complexity-penalized rule extraction. Our formulation of threshold selection is inspired by classical information-theoretic principles: the minimum description length principle (MDL) (Grünwald, 2007), which balances model complexity against residual error, and the information bottleneck (IB) (Tishby et al., 2000; Tishby & Zaslavsky, 2015), which formalizes the tradeoff between preserving task-relevant information and compressing the representation. Prior work has applied MDL to interpretable rule learning (Proença & van Leeuwen, 2020) and IB to representation learning (Alemi et al., 2017). We borrow only the tradeoff structure: our complexity term is a literal count rather than a code length, and we make no claim to optimize either quantity in its formal sense.

B.5 Comparison to Prior Work

Table 7 situates LUCID within the broader landscape of explainable RL methods based on structural and methodological criteria.

Table 7: Taxonomy of interpretability methods in deep RL.

Method	Paradigm	Policy Structure	Variable	Concept Source
LICORICE (CBM) (Ye et al., 2025)	Ad-hoc	MLP	**Discrete**	Predefined (human/VLM)
VIPER (Bastani et al., 2018)	Post-hoc	Decision Tree	Continuous	None
SA-DT	Post-hoc	Decision Tree	Continuous	None
Soft-DT (Frosst & Hinton, 2017)	Post-hoc	Decision Tree	Continuous	None
LUCID (Ours)	Post-hoc	**DNF**	**Binary**	Discovered; VLM-labelled, blind-tested

Traditional surrogate models (VIPER, SA-DT, Soft-DT) rely heavily on hierarchical decision trees operating on continuous, polysemantic variables. On Atari, trees fitted to these features need 190–377 feature tests (Table 1). Conversely, CBMs like LICORICE provide binary semantic variables, but their concept set is fixed before training, so the explanation can only use features someone anticipated. LUCID sits between these paradigms: it discovers binary concepts without annotation and composes them with native DNF logic. The discovered concepts carry no semantics by construction; the VLM labels we attach to them are evaluated by a blind predictive test (Section 5, Appendix H), and what the concepts encode is established separately with simulator factors and interventions (Sections E.6 and E.7).

C Formal Details of LUCID

C.1 Problem Formulation

Given a trained baseline policy π_{base} operating on high-dimensional visual observations $\mathcal{O}$ (e.g., raw pixels), with a frozen feature extractor $\phi : \mathcal{O} \to \mathbb{R}^d$ (e.g., a CNN) and a discrete action space, our objective is to distill this model into an interpretable surrogate π_{LUCID} such that it maintains high fidelity to the original policy:

$$\pi_{\text{LUCID}}(o) \approx \pi_{\text{base}}(o) \quad \forall o \in \mathcal{O} \tag{7}$$

Importantly, our framework is agnostic to the specific RL algorithm (e.g., PPO (Schulman et al., 2017), DQN (Mnih et al., 2015)) used to train the base policy, requiring only access to the learned embeddings in $\mathbb{R}^d$ and the discrete action distributions. To guarantee interpretability, π_{LUCID} must take the form of Disjunctive Normal Form (DNF) Boolean rules. Each action a is scored by the weighted number of its satisfied conjunctions $C_{a,i}$ over a set of binary concepts $c(o)$:

$$\pi_{\text{LUCID}}(o) = \arg\max_a \sum_{i=1}^{n_a} \sigma(\tilde{b}_{a,i})\, \mathbf{1}\big[C_{a,i}(c(o))\big] \tag{8}$$

where $c : \mathbb{R}^d \to \{0,1\}^D$ maps the continuous CNN features into a discrete, interpretable concept space generated by the SAE.

C.2 Sparse Concept Encoder

The first module of LUCID is an overcomplete SAE ($d \to D$, where $D \gg d$) that maps each normalized embedding x to a sparse concept activation z via a TopK gating function:

$$h = \text{ReLU}(W_e(x - b_d) + b_e) \quad \text{(pre-activation)} \tag{9}$$

$$z = \text{TopK}_k(h) \quad \text{(sparse concepts, } k \ll D\text{)} \tag{10}$$

$$\hat{x} = W_d z + b_d \quad \text{(reconstruction)} \tag{11}$$

The SAE is regularized to reconstruct the input while maintaining sparsity, contributing the following terms to the joint loss:

$$\mathcal{L}_{\text{SAE}} = \lambda_{\text{rec}} \tfrac{1}{d}\|x - \hat{x}\|_2^2 + \lambda_1 \tfrac{1}{D}\|h\|_1. \tag{12}$$

Following each gradient step, decoder columns (d_j) are strictly re-normalized to unit norm ($\|d_j\|_2 = 1$). The TopK projection is what makes joint training tractable: at any input, the logic layer only sees, and only sends gradients back through, the k active concepts, while the remaining $D - k$ atoms continue to be shaped almost exclusively by the reconstruction signal.

C.3 Binarization Bottleneck

Each sparse concept activation z_j, $j \in [D]$, is passed through a learnable per-concept sigmoid bottleneck that softly forces it toward a binary state $c_j \in \{0,1\}$:

$$c_j = \sigma\big(\alpha_j(z_j - \beta_j)\big), \qquad \alpha_j = \text{softplus}(\rho_j) + \tfrac{1}{2}. \tag{13}$$

The sharpness α_j (kept above $1/2$ by construction) and the shift β_j are learned jointly with the rest of the network. Because the input embeddings are standardized before the SAE, no further normalization of z is applied. Note that a concept switched off by the TopK gate ($z_j = 0$) is mapped to $c_j = \sigma(-\alpha_j\beta_j)$ rather than exactly 0; the learned shift $\beta_j > 0$ together with the bimodality penalty below drives this value toward 0.

C.4 Product T-Norm DNF Logic Layer

With binarized concepts, we learn n_a conjunctive clauses per action. We use product t-norm fuzzy logic to keep the conjunctions differentiable during training. For a given clause i and concept j, we define a soft literal ℓ_{ij}:

$$\ell_{ij} = p_{ij} \cdot c_j + n_{ij} \cdot (1 - c_j) + (1 - p_{ij} - n_{ij}). \tag{14}$$

The weights p_{ij} and n_{ij} are derived from a learnable 3-way softmax. They dictate whether concept j must be present ($p_{ij} \approx 1$), must be absent ($n_{ij} \approx 1$), or is entirely ignored ($p_{ij}, n_{ij} \approx 0$).

The logical AND within clause C_i is implemented via the log-product formulation of the product t-norm, and the logical OR across clauses is a simple summation of logits for the final action score:

$$C_i = \sigma\left(\sum_{j=1}^{D} \log(\ell_{ij} + \epsilon) + b_i\right), \qquad \text{score}_a = \sum_{i=1}^{n_a} C_{a,i}, \tag{15}$$

where $b_i \in \mathbb{R}$ is a learned per-clause bias (written $b_{a,i}$ when the action index is needed) and $\epsilon > 0$ is a small numerical-stability constant.

C.5 JOINT OPTIMIZATION

The full LUCID model, SAE encoder/decoder, per-feature binarization parameters (α_j, β_j), and DNF selector and bias parameters, is optimized end-to-end under a single weighted loss:

$$\mathcal{L}_{\text{total}} \;=\; \mathcal{L}_{\text{SAE}} \;+\; \lambda_{\text{CE}}\,\mathcal{L}_{\text{CE}}\big(s^{\text{soft}}(x), a^{\text{base}}\big) \;+\; \omega_t\,\lambda_{\text{bim}}\,\mathcal{L}_{\text{bim}} \;+\; \lambda_{\text{L0}}\,\mathcal{L}_{\text{L0}} \;+\; \omega_t\,\lambda_{\text{ent}}\,\mathcal{L}_{\text{ent}}, \quad (16)$$

where all terms are averaged over the mini-batch. $\mathcal{L}_{\text{CE}}$ is the (optionally class-weighted) cross-entropy between the softmax of the action scores $s^{\text{soft}}(x)$ and the baseline policy's action a^{base}; $\mathcal{L}_{\text{bim}} = \frac{1}{D}\sum_j c_j(1-c_j)$ pushes the bottleneck outputs toward 0 or 1; $\mathcal{L}_{\text{L0}} = \frac{1}{|\mathcal{C}|D}\sum_{i,j}(p_{ij}+n_{ij})$, with $|\mathcal{C}| = \sum_a n_a$ the total number of clauses, encourages short, sparse rules; and $\mathcal{L}_{\text{ent}} = \sum_{i,j}\mathcal{H}(\mathbf{v}_{ij})$ is a categorical-entropy penalty on the selection probability vector $\mathbf{v}_{ij} = [p_{ij}, n_{ij}, 1-p_{ij}-n_{ij}]$ that enforces unambiguous, one-hot literal choices. The discreteness weight follows a warmup-then-linear-ramp schedule, $\omega_t = 0$ for epoch $t < T_{\text{w}}$ and $\omega_t = \min\big(1, (t-T_{\text{w}})/T_{\text{r}}\big)$ otherwise, so that concepts and rules first fit the teacher before being forced toward discrete states (values in Appendix I). Throughout training, clause biases enter the forward pass clipped at $b_{\max} = 5$ (Section C.6), gradients are clipped to a global norm of 5, and decoder columns are re-normalized after every step.

Why we expect joint training to be well-behaved. We expect two architectural choices to keep this joint objective from degenerating (we do not ablate them separately), despite the discrete structure being learned alongside the dictionary. First, the SAE's TopK sparsity ensures that the imitation signal $\nabla_{\theta_{\text{SAE}}}\mathcal{L}_{\text{CE}}$ only reaches the small subset of concepts active for each input, so the dictionary atoms relevant to the agent's decision are shaped jointly by reconstruction and imitation, while inactive atoms continue to be governed almost exclusively by reconstruction. Second, the bimodality and entropy losses progressively contract the binarization bottleneck and selector probabilities toward their discrete limits, which freezes the effective combinatorial structure of the DNF as training proceeds, even while the underlying continuous parameters keep moving. The net effect is that concept geometry and rule structure stabilize together rather than racing each other.

C.6 RULE EXTRACTION AND THRESHOLD SELECTION

We make the one free choice in rule extraction, the selector threshold τ, explicit. Given the trained parameters from Eq. (3), an extraction is fully determined by a threshold $\tau \in [0,1]$ which selects positive and negative literal sets $P_{a,i}(\tau) = \{j : p_{a,i,j} > \tau\}$ and $N_{a,i}(\tau) = \{j : n_{a,i,j} > \tau\}$.

Because the selectors come from a 3-way softmax ($p_{a,i,j} + n_{a,i,j} + u_{a,i,j} = 1$), any $\tau \geq 1/2$ guarantees $P_{a,i}(\tau) \cap N_{a,i}(\tau) = \emptyset$. For $\tau < 1/2$, a concept may satisfy both $p > \tau$ and $n > \tau$; the clause then requires $c_j \wedge \neg c_j$, which is always false, so it contributes nothing to the disjunction and is pruned.

Extraction conventions. Clause extraction follows three fixed rules, applied identically in all experiments. (i) *Contradiction* $\Rightarrow$ `False`: if some concept j has both $p_{a,i,j} > \tau$ and $n_{a,i,j} > \tau$ (possible only for $\tau < 1/2$), the clause is set to `False` and pruned, as argued above. (ii) *Universal clause* $\Rightarrow$ `True`: if no concept is selected, i.e., $p_{a,i,j} \leq \tau$ and $n_{a,i,j} \leq \tau$ for all j (in particular whenever $u_{a,i,j} > \tau \geq 1/2$ for every j), the clause is kept as `True` and fires on every state, acting as a constant prior $\sigma(\tilde{b}_{a,i})$ for action a; it is not counted in rule complexity. In practice universal clauses act as a default action, and Section E.4 shows that replacing them by the fallback changes hard fidelity by at most 1.4 points, raising it where the learned prior is poor (Boxing). (iii) *Bias clipping*: every clause uses $\tilde{b}_{a,i} = \min(b_{a,i}, b_{\max})$ with $b_{\max} = 5$, both in the soft forward pass during training and in the extracted rules. Without this cap, the network could inflate $b_{a,i}$ so that $\sigma(b_{a,i} + L_{a,i}(x)) \approx 1$ despite a strongly negative log-product, i.e., a clause would fire regardless of its literals, which bypasses the sparsity and entropy penalties and makes the extracted conjunction meaningless. The cap forces the clause output to be governed by its literals.

These literal sets induce a hard rule policy π_τ^{hard} in the standard way: each non-contradictory clause evaluates to a boolean weighted by its sigmoid bias, action scores accumulate over satisfied clauses,

and a fallback action is taken when no clause fires:

$$T_{a,i}(x) = \Big(\bigwedge_{j\in P_{a,i}(\tau)} c_j^{\text{hard}}(x)\Big) \wedge \Big(\bigwedge_{j\in N_{a,i}(\tau)} \neg c_j^{\text{hard}}(x)\Big), \tag{17}$$

$$s_a^{\text{hard}}(x) = \textstyle\sum_{i=1}^{n_a} \sigma(\tilde{b}_{a,i}) \cdot \mathbf{1}[T_{a,i}(x) = \text{True}], \tag{18}$$

$$\pi_\tau^{\text{hard}}(x) = \begin{cases} a_{\text{fallback}} & \text{if } \max_a s_a^{\text{hard}}(x) = 0 \\ \arg\max_{a\in\mathcal{A}} s_a^{\text{hard}}(x) & \text{otherwise,} \end{cases} \tag{19}$$

where $c_j^{\text{hard}}(x) = \mathbf{1}[c_j(x) > \theta]$ with concept threshold θ ($\theta = 0.66$ in all experiments; distinct from the selector threshold τ), $\tilde{b}_{a,i} = \min(b_{a,i}, 5)$ is the clipped bias of rule (iii), contradictory clauses are `False` (rule (i)), universal clauses are `True` (rule (ii)), and $a_{\text{fallback}} \in [A]$ is the most frequent teacher action in the offline dataset, returned when no clause fires. The goal is to choose τ that best preserves the soft policy's behavior while keeping the rule set compact.

Unified objective. We define the *information loss* of an extraction as the expected disagreement between the soft and hard policies (interpreted as the 0–1 information cost of discretizing the teacher's decisions), and the *description length* as the total count of active literals:

$$\mathcal{L}_{\text{info}}(\tau) = \mathbb{E}_{x\sim\mathcal{D}}\big[\mathbf{1}\big[\pi^{\text{soft}}(x) \neq \pi_\tau^{\text{hard}}(x)\big]\big], \quad \mathcal{L}_{\text{MDL}}(\tau) = \textstyle\sum_{a,i} \big(|P_{a,i}(\tau)| + |N_{a,i}(\tau)|\big), \tag{20}$$

i.e. the number of literals summed over clauses (Table 10 counts each distinct clause once).

The optimal extraction threshold is then the minimizer of the unified Lagrangian:

$$\boxed{\tau^* \;=\; \arg\min_{\tau\in[0,1]} \;\mathcal{L}_{\text{info}}(\tau) \;+\; \lambda\,\mathcal{L}_{\text{MDL}}(\tau).} \tag{21}$$

This objective has a direct operational reading: it minimizes the fraction of teacher decisions destroyed by discretization, penalized by the number of literals the rule set uses. We denote the second term $\mathcal{L}_{\text{MDL}}$ because it plays the role of the model-cost term in an MDL two-part code (Grünwald, 2007); it is, however, a literal count rather than a code length in bits, and λ is a tradeoff weight rather than a calibrated description-length price. The analogy to the information bottleneck (Tishby et al., 2000) is likewise structural only. None of our results depend on these interpretations: Lemma 4 and Proposition 1 use only that both terms are piecewise constant in τ. We do not tune λ: since the rule set is identical for every $\tau \in [0.5, 0.9]$, the objective takes the same value on that whole range for any λ.

Step-function structure. Both terms in Eq. (21) only change value when τ crosses a learned selector probability. This yields:

Lemma 4 (Step-Function Structure)**.** *Let $\mathcal{S} = \{p_{a,i,j}\} \cup \{n_{a,i,j}\} \cup \{0, 1\}$ be the sorted set of all selector probabilities and endpoints. Then $\mathcal{L}_{\text{info}}(\tau) + \lambda\,\mathcal{L}_{\text{MDL}}(\tau)$ is piecewise constant on the open intervals between consecutive elements of $\mathcal{S}$, and τ^* is attained on one such interval.*

Central-interval invariance. The bimodality and entropy terms in Eq. (3) drive the selector probabilities sharply toward $\{0, 1\}$, leaving the central region empty at convergence. This exactly carves out an invariant operational plateau:

Proposition 1 (full statement) (Central-Interval Invariance & Robustness)**.** *Let $\mathbf{v}_{a,i,j} = [p, n, u]$ denote the selector probabilities, and call the selectors δ-*polarized *if no $p_{a,i,j}$ or $n_{a,i,j}$ lies in the open interval $(\delta, 1-\delta)$. Then:*

- *(a)* Majority. *For any $\tau \geq 1/2$, $P_{a,i}(\tau) \cap N_{a,i}(\tau) = \emptyset$, and every selected literal holds a strict majority of its selector mass ($p > n+u$, resp. $n > p+u$); $\tau = 1/2$ is the smallest threshold with this property.*
- *(b)* Confidence. *For any $\tau \geq 2/3$, every selected literal leaves discarded mass below the uniform prior, $1 - p < 1/3$ (resp. $1 - n < 1/3$).*
- *(c)* Invariance and robustness. *If the selectors are δ-polarized with $\delta < 1/3$, the extracted rule set $\{P_{a,i}(\tau), N_{a,i}(\tau)\}$ is identical for all $\tau \in [1/2, 2/3]$. Consequently, extraction at $\tau = 1/2$ satisfies both (a) and (b), and $\tau = 1/2$ uniquely maximizes the distance $\min(\tau - \delta, 1 - \delta - \tau)$ to the nearest selector value, i.e., it is the threshold most robust to symmetric perturbations of the selectors.*

Parts (a) and (b) motivate the interval $[1/2, 2/3]$: $1/2$ is the minimal threshold with a majority guarantee, and $2/3$ is the minimal threshold with a uniform-prior confidence guarantee. Part (c) shows that, once training has polarized the selectors, the two coincide: every τ on the plateau inherits both guarantees, and under symmetric polarization $\tau = 1/2$ is the choice most robust to selector perturbations. The result characterizes τ through these structural guarantees; it does not claim that any particular τ minimizes Eq. (21) for every model, which is an empirical question. Empirically, no selector probability falls in $(0.5, 0.9]$ in any of the five environments (Appendix F), so every selected literal has $p_{a,i,j} > 0.9$ (resp. $n_{a,i,j} > 0.9$) and the rule set is identical for all $\tau \in [0.5, 0.9]$. We extract at $\tau = 0.66$, which therefore returns exactly the rules obtained at $\tau = 1/2$ and at $\tau = 2/3$ and satisfies both (a) and (b).

D Soft–Hard Certificate: Full Statement

A core challenge in neuro-symbolic surrogate modeling is ensuring that the extracted discrete rules accurately reflect the behavior of the continuous neural policy. In this section, we establish a formal guarantee of *Soft–Hard Fidelity*: we prove that the soft LUCID surrogate is guaranteed to agree with the extracted hard logical rules whenever the action margin exceeds the total conceptual perturbation (Theorem 1). The guarantees in this section depend only on the converged values of the LUCID parameters and are therefore agnostic to the particular optimization schedule used to produce them. Full proofs and supporting lemmas are deferred to Appendix A.

D.1 Formal Setup

We state the guarantee for the model exactly as implemented. Let σ denote the sigmoid and $\epsilon = 10^{-8}$ the stability constant of the logic layer.

Soft surrogate. The bottleneck outputs $c_j^{\text{soft}}(x) = \sigma\big(\alpha_j(z_j(x) - \beta_j)\big)$ with $\alpha_j = \text{softplus}(\rho_j) + \frac{1}{2} > 0$. Each clause uses the *clipped* bias $\tilde{b}_{a,i} = \min(b_{a,i}, b_{\max})$ with $b_{\max} = 5$, the same value used during training and extraction. With $\ell_{a,i,j}^{\text{soft}}(x) = p_{a,i,j}\, c_j^{\text{soft}}(x) + n_{a,i,j}\,(1 - c_j^{\text{soft}}(x)) + u_{a,i,j}$ and $u_{a,i,j} = 1 - p_{a,i,j} - n_{a,i,j}$,

$$L_{a,i}(x) = \sum_{j=1}^{D} \log\big(\ell_{a,i,j}^{\text{soft}}(x) + \epsilon\big), \qquad C_{a,i}^{\text{soft}}(x) = \sigma\big(\tilde{b}_{a,i} + L_{a,i}(x)\big), \qquad s_a^{\text{soft}}(x) = \sum_{i=1}^{n_a} C_{a,i}^{\text{soft}}(x), \tag{22}$$

and $\pi^{\text{soft}}(x) = \arg\max_a s_a^{\text{soft}}(x)$.

Hard rules. Two thresholds are involved and must not be confused: the *selector threshold* τ defines the literal sets $P_{a,i} = P_{a,i}(\tau)$, $N_{a,i} = N_{a,i}(\tau)$ (Section C.6), which are disjoint for $\tau \geq 1/2$, and the extraction conventions (i)–(iii) apply; the *concept threshold* $\theta \in (0, 1)$ binarizes concepts, $c_j^{\text{hard}}(x) = \mathbf{1}[c_j^{\text{soft}}(x) > \theta]$ (we use $\theta = 0.66$). A literal is satisfied, $\ell_{a,i,j}^{\text{hard}}(x) = 1$, iff $j \in P_{a,i}$ and $c_j^{\text{hard}}(x) = 1$, or $j \in N_{a,i}$ and $c_j^{\text{hard}}(x) = 0$; literals outside $P_{a,i} \cup N_{a,i}$ are ignored. By convention (ii), a clause with $P_{a,i} \cup N_{a,i} = \emptyset$ is always true; by convention (i), a clause with $P_{a,i} \cap N_{a,i} \neq \emptyset$ is always false, since for $j \in P_{a,i} \cap N_{a,i}$ exactly one of the two literals on j is violated whatever $c_j^{\text{hard}}(x)$ is. The hard clause and score are

$$C_{a,i}^{\text{hard}}(x) = \sigma(\tilde{b}_{a,i}) \cdot \mathbf{1}\big[T_{a,i}(x)\big] \in \{0, \sigma(\tilde{b}_{a,i})\}, \qquad s_a^{\text{hard}}(x) = \sum_{i=1}^{n_a} C_{a,i}^{\text{hard}}(x), \tag{23}$$

with $T_{a,i}$ as in Eq. (17). Let $\mathcal{X}_{\text{trig}} = \{x : \exists(a, i),\ T_{a,i}(x) = \text{True}\}$ be the set of states on which at least one clause fires. On $\mathcal{X}_{\text{trig}}$ the hard policy is $\pi^{\text{hard}}(x) = \arg\max_a s_a^{\text{hard}}(x)$; outside it returns a_{fallback}, the most frequent teacher action in the offline dataset.

The *action margin* is $\gamma(x) = s_{a^*}^{\text{hard}}(x) - \max_{a \neq a^*} s_a^{\text{hard}}(x)$ with $a^* = \pi^{\text{hard}}(x)$, and the *per-concept softness* is $\varepsilon_j(x) = \min(c_j^{\text{soft}}, 1 - c_j^{\text{soft}}) = \big(1 + e^{\alpha_j |z_j(x) - \beta_j|}\big)^{-1}$.

D.2 SOFT–HARD FIDELITY GUARANTEE

Theorem 1 (full statement) (Soft–Hard Fidelity)**.** *For each action a and clause i, define the per-clause error bound*

$$\Delta_{a,i}(x) = \begin{cases} \left|\sigma(\tilde{b}_{a,i}) - \sigma\big(\tilde{b}_{a,i} + L_{a,i}(x)\big)\right| & \text{if } T_{a,i}(x) = \text{True}, \\ \sigma\Big(\tilde{b}_{a,i} + \log\big(\ell^{\text{soft}}_{a,i,j^*}(x) + \epsilon\big) + (D-1)\log(1+\epsilon)\Big) & \text{if } T_{a,i}(x) = \text{False}, \end{cases} \tag{24}$$

where $j^* = \arg\min_{j \in V_{a,i}(x)} \ell^{\text{soft}}_{a,i,j}(x)$ and $V_{a,i}(x)$ is the (non-empty) set of concepts carrying a violated literal, i.e., $j \in P_{a,i}$ with $c^{\text{hard}}_j(x) = 0$ or $j \in N_{a,i}$ with $c^{\text{hard}}_j(x) = 1$ (so $V_{a,i}(x) \supseteq P_{a,i} \cap N_{a,i}$ for contradictory clauses), and let $\Delta_a(x) = \sum_{i=1}^{n_a} \Delta_{a,i}(x)$. Let $x \in \mathcal{X}_{\text{trig}}$ with $\gamma(x) > 0$. If

$$\Delta_{a^*}(x) + \Delta_a(x) < s^{\text{hard}}_{a^*}(x) - s^{\text{hard}}_a(x) \quad \forall a \neq a^*, \tag{25}$$

then $\pi^{\text{soft}}(x) = \pi^{\text{hard}}(x)$. Condition (25) is implied by the simpler sufficient condition $\Delta_{\text{total}}(x) := \Delta_{a^*}(x) + \max_{a \neq a^*} \Delta_a(x) < \gamma(x)$. Consequently, with the certified set $\mathcal{X}_{\text{cert}} = \{x \in \mathcal{X}_{\text{trig}} : \gamma(x) > 0,$ (25) holds$\}$,

$$\Pr_{x \sim \mathcal{D}}\big[\pi^{\text{soft}}(x) \neq \pi^{\text{hard}}(x)\big] \leq 1 - \Pr_{x \sim \mathcal{D}}[x \in \mathcal{X}_{\text{cert}}]\,. \tag{26}$$

Intuitively, the theorem isolates a *safe zone* on which the discrete rules and the continuous surrogate provably agree: whenever the hard margin exceeds the cumulative soft perturbation, binarization slack cannot flip the argmax. A true clause loses score only through the cumulative "log-drag" $L_{a,i}(x)$ of all D literals, while a false clause is dominated by its most-violated literal. States on which no clause fires ($x \notin \mathcal{X}_{\text{trig}}$, fallback) or on which the hard policy ties ($\gamma(x) = 0$) are never certified and are counted as uncertified in Eq. (26). All quantities depend only on converged parameters and soft activations, so the *fidelity coverage* $\Pr[x \in \mathcal{X}_{\text{cert}}]$ is directly computable on any trajectory. The $(D-1)\log(1+\epsilon)$ term (below 6×10^{-6} for $D \leq 512$) accounts for the fact that $\ell + \epsilon$ may slightly exceed 1.

Bridging the Gap to the PPO Teacher. Theorem 1 certifies agreement strictly between π^{hard} and the *soft LUCID surrogate* π^{soft}; it is not a worst-case guarantee for the PPO teacher π^{teacher}. It links back to the teacher via the triangle inequality for 0–1 disagreement:

$$\begin{aligned} \Pr_{x \sim \mathcal{D}}\big[\pi^{\text{hard}}(x) \neq \pi^{\text{teacher}}(x)\big] &\leq \underbrace{\Pr_{x \sim \mathcal{D}}\big[\pi^{\text{soft}}(x) \neq \pi^{\text{teacher}}(x)\big]}_{\text{Imitation Error}} + \underbrace{\Pr_{x \sim \mathcal{D}}\big[\pi^{\text{soft}}(x) \neq \pi^{\text{hard}}(x)\big]}_{\text{Soft–Hard Discretization Error}} \\ &\leq \big(1 - \text{AF}_{\text{soft}}\big) + \big(1 - \Pr[x \in \mathcal{X}_{\text{cert}}]\big), \end{aligned} \tag{27}$$

where AF_{soft} is the soft action fidelity to the teacher. The left-hand side equals $1 - \text{AF}_{\text{hard}}$, so the bound is fully checkable from reported quantities: the imitation error is minimized by $\mathcal{L}_{\text{CE}}$, and the discretization error is bounded by Theorem 1 and reduced by $\mathcal{L}_{\text{bim}}$ and $\mathcal{L}_{\text{ent}}$.

E EXTENDED RESULTS

E.1 BASELINES

All tree baselines receive the output of the teacher's frozen CNN feature extractor, the same input as LUCID, but no SAE. **SA-DT** is a CART tree trained by behavioural cloning on the teacher's actions. **VIPER** (Bastani et al., 2018) follows the original algorithm with two adaptations: PPO provides no Q-function, so the cost-sensitive resampling weights each state by the teacher's action-probability gap $\max_a \pi(a \mid s) - \min_a \pi(a \mid s)$ instead of the Q-value gap; and among the DAgger iterates we keep the tree with the highest success rate over 20 online episodes (return > 0 on Atari). **Soft-DT** (Frosst & Hinton, 2017) is a soft decision tree with L1-sparse routing. SA-DT and Soft-DT are limited to a maximum depth of 2 on Dynamic-Obstacles, 4 on DoorKey and PixelCartPole, and 6 on Atari. VIPER was given a looser limit on MiniGrid (8 on DoorKey, 4 on Dynamic-Obstacles), but its trees stop growing at the same depths as the other trees (Table 9), so the depth budget does not constrain it. Within these limits, their remaining hyperparameters were tuned extensively by hand, and we report the best configuration among those whose size (feature tests and binary concepts,

Table 8: Full version of Table 1, with success rates. Score and success rate: mean over five evaluation seeds (42–46); action fidelity: agreement with the teacher on the teacher rollouts used for training (for LUCID this includes its training states; tree baselines are scored on a 20% split held out from their training data). On independent rollouts (Table 5) LUCID's hard fidelity is within 0.13 points of the values here. LUCID: training seed 42. **Bold**: best post-hoc interpretable surrogate (LUCID Hard, VIPER, SA-DT, Soft-DT).

Metric	PPO	LUCID (Soft)	LUCID (Hard)	VIPER	SA-DT	Soft-DT	LICORICE
DoorKey-6x6 (5×200 ep)							
Score	0.968	0.968	**0.968**	0.955	**0.968**	0.937	0.968
Success rate (%)	100.0	100.0	**100.0**	98.6	**100.0**	96.0	100.0
Action fidelity (%)	–	100.00	**99.92**	99.16	99.86	84.55	–
Dynamic-Obs-5x5 (5×200 ep)							
Score	0.891	0.891	0.891	0.891	0.891	0.891	0.891
Success rate (%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Action fidelity (%)	–	100.00	**100.00**	98.37	**100.00**	**100.00**	–
PixelCartPole (5×40 ep)							
Score	500.0	475.5	442.5	402.5	370.5	**494.0**	500.0
Action fidelity (%)	–	**99.99**	97.13	95.67	96.25	99.10	–
Pong (5×20 ep)							
Score	21.0	21.0	**21.0**	20.9	0.3	**21.0**	21.0
Action fidelity (%)	–	99.90	**99.85**	63.02	62.34	88.85	–
Boxing (5×20 ep)							
Score	92.2	94.1	**92.5**	90.1	74.7	92.0	92.6
Action fidelity (%)	–	99.78	**96.80**	69.20	69.96	86.43	–

Table 9: Hyperparameters of the tree baselines (reported runs). Values shared by all environments are given once.

Method	**Setting**	**DoorKey-6x6**	**Dynamic-Obs-5x5**	**PixelCartPole**	**Pong**	**Boxing**
All trees	Depth of the reported tree	4	2	4	6	6
SA-DT	Min. samples per leaf	1				
	Train / test split	80% / 20% of the teacher rollouts				
VIPER	Maximum depth allowed	8	4	4	6	6
	DAgger iterations	75	15	20	15	15
	Episodes per iteration	50	20	15	20	20
	Initial teacher episodes	100	40	30	40	40
	Max. steps per episode	100	100	500	27,000	27,000
	Min. samples per leaf	4	5	5	5	5
	Cost-complexity pruning α	10^{-3}				
	Iterate selection	success rate over 20 online episodes				
Soft-DT	Training epochs	1000	1000	500	200	200
	Optimizer / learning rate	Adam / 10^{-3}				
	Batch size	256				
	L1 weight on routing parameters	10^{-3}				
	Train / test split	80% / 20% of the teacher rollouts				
All	Evaluation episodes (per seed $\times$ seeds)	200×5	200×5	40×5	20×5	20×5

Table 10) is comparable to or larger than LUCID's. The trees are fitted on 80% of the teacher rollouts and their fidelity is measured on the remaining 20%. **LICORICE** (Ye et al., 2025) learns its own concept-bottleneck policy from scratch with PPO rather than distilling our teacher, so action fidelity is not defined for it; we retrained it with the authors' code and the hyperparameters reported by Ye et al. (2025), changing only the DoorKey grid from 7×7 to 6×6 to match our setting.

Table 9 lists the settings of the reported runs; values not listed are the scikit-learn or PyTorch defaults (Gini criterion, no cost-complexity pruning for SA-DT). Returns are averaged over the same five evaluation seeds (42–46) as LUCID.

E.2 Performance and Rule Complexity

Table 8 adds success rates to Table 1, and Table 10 breaks rule size down into concepts, rules and literals.

E.3 Controlled Ablations

The baselines in Tables 1–10 differ from LUCID in both representation and rule learner, so they cannot isolate which component matters. We therefore run four controls on the same frozen teacher, the same validation split as LUCID (10% of the training rollouts, on which LUCID's checkpoint is selected), and LUCID's own hyperparameters in each environment: (i) *two-stage*, a reconstruction-only SAE frozen before training the same bottleneck and DNF layer; (ii) *no SAE*, the bottleneck and DNF layer applied directly to the teacher's CNN features; and (iii) SAE+CART and (iv) SAE+L1, a decision tree and an L1-regularized multinomial logistic regression on the reconstruction-only SAE concepts, each swept over its complexity knob (cost-complexity pruning α and inverse regularization strength C). Complexity is counted uniformly as the number of feature tests: literals for DNF, internal nodes for trees, non-zero coefficients for linear models. Counting internal nodes rather than root-to-leaf literals is the convention most favourable to trees.

Table 10: Structural complexity (same models as Table 1). *Conc.*: binary concepts, i.e. the binarized SAE concepts used by the rules (LUCID) or the threshold tests at internal nodes (trees); *Rules*: clauses (DNF) or leaves (trees); *Lit.*: feature tests summed over clauses or root-to-leaf paths (Soft-DT: non-zero split weights). Identical clauses of the same action are counted once.

	DoorKey-6x6			**Dynamic-Obs-5x5**			**PixelCartPole**			**Pong**			**Boxing**		
Method	Conc.	Rules	Lit.	Conc.	Rules	Lit.	Conc.	Rules	Lit.	Conc.	Rules	Lit.	Conc.	Rules	Lit.
LUCID (Hard)	12	6	**16**	4	3	**4**	10	9	45	17	18	**19**	43	53	**93**
VIPER	8	9	30	2	3	5	9	10	**34**	33	34	190	38	39	214
SA-DT	11	12	44	2	3	5	15	16	64	41	42	240	62	63	377
Soft-DT	15	16	396	3	4	48	15	16	204	63	64	5766	63	64	8496

Table 4 gives the numbers behind the conclusions of Section 4. Two details are not stated there: LUCID's extracted rules stay within 2.6 points of the teacher in every environment, and SAE+L1, besides never reaching LUCID's fidelity on Atari, needs 4–8$\times$ more non-zero weights to match it on MiniGrid.

Three caveats apply. LUCID's checkpoint is selected on this validation split, while the controls select on a separate split carved from their training states; this can only favour LUCID, so the two close comparisons, SAE+CART on DoorKey (1.2 points) and SAE+L1 on PixelCartPole (0.6 points), should be read as ties. PixelCartPole is also LUCID's least stable case, with a literal count of 32.0 ± 17.0 across seeds. Finally, hard fidelity here and in Table 2 is measured on this validation split with universal clauses dropped and uncovered states sent to the fallback (the most frequent action of the training split), whereas Tables 1 and 15 evaluate the full hard policy of Eq. (4) on all training rollouts. The two agree within 0.6 points except on Boxing (98.39% vs. 96.98%), where the universal-clause prior agrees with the teacher on only 4.9% of the states it decides (Section E.7).

E.4 Rule Structure: Coverage, Conflicts and Clause Weights

Two properties of the hard policy in Eq. (4) could make its literal count an unfair measure of complexity: universal clauses (convention (ii)) fire on every state and are not counted, and clause weights $\sigma(\tilde{b}_{a,i})$ arbitrate when clauses of several actions fire, acting as a hidden linear layer. Table 2 measures both. Universal clauses act as default-action priors: removing them and sending uncovered states to a_{fallback} leaves fidelity at 97–100% in every environment, within 1.4 points of the full hard policy of Tables 1 and 15. We report the full policy as the main result because it is the rule set extracted from the trained model. Rule coverage is correspondingly not uniform: on DoorKey the non-universal clauses fire on only 46.5% of states, and the remaining states are handled by the default action `Forward`; on the other four environments coverage exceeds 95%. Conflicts are absent on MiniGrid, rare on PixelCartPole, and frequent on Atari (5.3% on Pong, 18.4% on Boxing), yet replacing weights by clause counts changes fidelity by at most 0.26% on any seed. The extracted policies therefore behave as unweighted rule sets with a default action, and the literal counts in Table 10 compare representations of the same kind.

E.5 Empirical Fidelity Validation

We evaluate Theorem 1 on an independent set of teacher rollouts per environment, collected with a different environment seed from the data used to train and test LUCID and stored together with

the raw observations; none of these states was seen during training or checkpoint selection. We use $\tau = \theta = 0.66$ and compute all quantities in double precision. For every state we compute $\gamma(x)$, the per-clause bounds $\Delta_{a,i}(x)$ and membership in $\mathcal{X}_{\text{cert}}$ under the pairwise condition (25); hard ties count as uncertified. Agreement on certified states is guaranteed by the proof, and no state violates Lemmas 1–2 or the theorem in any environment. The informative question is how tight the bound $1 - \Pr[x \in \mathcal{X}_{\text{cert}}]$ is relative to the measured disagreement $\Pr[\pi^{\text{soft}} \neq \pi^{\text{hard}}]$ (Table 5).

The bound is tight on both MiniGrid tasks and Pong, certifying 99.1–100% of states and exceeding the measured disagreement by at most 0.7 points; on DoorKey, 95% of the uncertified states are genuine disagreements. It is loose on PixelCartPole and Boxing, which leave 12.5% and 31.1% of states uncertified while only about 3% disagree. In both cases the slack comes from false clauses, whose bound keeps only the most violated literal, $\sigma(\tilde{b}_{a,i} + \log \ell_{a,i,j^*})$, and is small only when $\ell_{a,i,j^*} \ll e^{-\tilde{b}_{a,i}}$. On PixelCartPole only 71% of rule-concept activations are clean, against more than 99% elsewhere, so violated literals keep a large soft value: the mean false-clause bound is 0.64, against 0.01–0.03 on the MiniGrid tasks and Pong. On Boxing the per-clause bound is moderate (0.18), but with 18 actions and 130 non-universal clauses (53 distinct ones, Table 10) it accumulates: the false clauses of the worst rival action alone sum to 6.7 on average, against a mean margin of 6.2. The triangle bound (27) holds everywhere but inherits this slack, so it is informative only where the certificate is tight (DoorKey: 0.091% bound vs. 0.087% measured hard error).

E.6 Do the Concepts Name Simulator Factors?

A compact rule set is only interpretable if its atoms correspond to something a human can name. We therefore test the learned dictionary against the simulator's own state variables, which are available for all five environments and were never seen during training.

Protocol. For each environment we fix a set of *parent factors* extracted from the simulator: object and agent predicates for MiniGrid (`has_key`, `door_open`, `wall_ahead`, ...), the four state variables for PixelCartPole, and object coordinates recovered with OCAtari (Delfosse et al., 2023a) for the Atari games (ball and paddle positions, their difference, and ball velocity for Pong; player and opponent positions and their difference for Boxing). Each parent is discretised into child bins by tertiles and sign. For every dictionary dimension j and parent factor p we compute the one-vs-rest AUC P_{jp} of the concept activation as a detector of p, folded so that $P_{jp} \geq 0.5$. Writing $a_1 \geq a_2$ for the two largest entries of row j, we label the dimension *monosemantic* (**G1**) if $a_1 \geq \kappa_{\text{hi}}$ and $a_1 - a_2 \geq \kappa_{\text{gap}}$; *polysemantic* (**G2**) if $a_2 \geq \kappa_{\text{hi}}$, i.e. it detects two factors equally well; *unaligned* (**G3**) if $a_1 < \kappa_{\text{lo}}$, i.e. it detects no factor at all; and *unclear* (**U**) otherwise. We report the prespecified setting $(\kappa_{\text{hi}}, \kappa_{\text{gap}}, \kappa_{\text{lo}}) = (0.75, 0.10, 0.65)$. *Completeness* is the mean over factors of the best AUC achieved by any dimension, i.e. how much of the factor set the dictionary covers at all.

We compare the LUCID dictionary against three representations of the same rollout data: the teacher's penultimate CNN features, a random projection, and PCA, each with the same number of live dimensions where possible. As a sanity check we recompute every AUC after shuffling the factor labels; the mean shuffled AUC is 0.500 in all five environments, confirming the measurement itself is unbiased.

Results. Table 11 reports the prespecified setting, averaged over three seeds per environment.

Three observations, two of them negative, follow.

First, in both MiniGrid environments the LUCID dictionary is substantially *less* polysemantic than the teacher's own features: it triples the monosemantic fraction (26.1% vs. 7.8% on DoorKey; 30.5% vs. 11.9% on Dynamic-Obstacles) and cuts the polysemantic fraction by a factor of two to four. The effect is consistent across all three seeds of each environment and is confirmed by the threshold-free test on the second-best factor AUC (CLES 0.26–0.33, $p \leq 3.2\text{e}{-4}$).

Second, this advantage does not transfer to the remaining environments. On PixelCartPole the ordering reverses outright: the CNN reaches 46.2% monosemantic dimensions against 9.0% for LUCID, and 82.7% of LUCID dimensions align with no factor at all. On the two Atari games the picture is more extreme still, with only 1–2% of LUCID dimensions passing the G1 test and 84–92% failing to reach κ_{lo} for any factor. We note that the CNN also performs poorly there (4.8% and 7.8% G1, with about 40% of its dimensions unaligned), which suggests that the object-coordinate factor set

Table 11: Concept–factor alignment at the prespecified threshold setting $(0.75, 0.10, 0.65)$, mean $\pm$ s.d. over three seeds. G1 = monosemantic, G2 = polysemantic, G3 = aligned with no factor. CNN, random and PCA baselines are deterministic given the rollout data, hence without spread. CLES is the common-language effect size of the one-sided Mann–Whitney test on the second-best factor AUC (LUCID vs. CNN, lower is better); it is computed only over dimensions whose best AUC reaches κ_{lo}.

Environment	Repr.	Completeness	G1 (%)	G2 (%)	G3 (%)	CLES	p
Dynamic-Obs-5x5	LUCID	0.809 ± 0.005	$\mathbf{30.50 \pm 2.62}$	38.56 ± 3.23	12.52 ± 3.55	0.26–0.33	$\leq 3.2\text{e}{-4}$
	CNN	**0.873**	11.94	80.60	4.48	–	–
DoorKey-6x6	LUCID	0.825 ± 0.001	$\mathbf{26.05 \pm 3.50}$	11.52 ± 2.08	28.81 ± 4.36	0.26–0.27	$\leq 1.6\text{e}{-7}$
	CNN	**0.893**	7.79	49.35	20.78	–	–
PixelCartPole	LUCID	0.703 ± 0.007	9.03 ± 2.14	0.00	82.72 ± 1.84	0.48–0.58	n.s.
	CNN	**0.741**	**46.23**	1.31	38.36	–	–
Pong	LUCID	0.680 ± 0.007	1.07 ± 0.61	0.13	91.93 ± 1.10	**0.14–0.24**	$\leq 1.0\text{e}{-5}$
	CNN	**0.801**	**4.80**	1.79	39.30	–	–
Boxing	LUCID	0.698 ± 0.010	2.33 ± 1.21	0.00	84.27 ± 0.99	0.21–0.23	$\leq 1.6\text{e}{-8}$
	CNN	**0.786**	**7.79**	9.70	40.30	–	–

we defined does not capture what either representation encodes, rather than that LUCID is uniquely misaligned. Confirming this would require a richer factor vocabulary for Atari, which we leave to future work.

Third, and independently of the alignment rate, the dimensions that *do* align are cleaner in LUCID than in the CNN in four of the five environments, and the effect is strongest precisely where the alignment rate is lowest (Pong, CLES 0.14–0.24 on only 19–24 retained dimensions; Boxing, 0.21–0.23). The two questions — *how many* concepts correspond to a named factor, and *how cleanly* those that do correspond — are therefore empirically distinct, and we report them separately throughout.

We also record a negative result on a stricter criterion. We had prespecified the conjunction "LUCID $\geq$ every baseline on both completeness and %G1" as a single pass/fail test over the full threshold grid. It fails in every grid cell of every environment (0/12 on MiniGrid, 0/36 elsewhere), because the LUCID dictionary's completeness is consistently *below* the CNN's: the sparse dictionary trades coverage of the factor set for purity of the dimensions it does cover. This trade-off is intrinsic to the sparsity objective, and we report it rather than relaxing the criterion post hoc. On the threshold-free test the same caution applies in the other direction: LUCID does not beat the random-projection and PCA baselines on second-best AUC in any environment (CLES 0.54–0.84 against LUCID). That comparison is however confounded on MiniGrid, where restricting to dimensions above κ_{lo} retains only 57–58% of the random-projection dimensions against 67–92% of LUCID's, and the retained subset is selected for alignment; there the comparison against the CNN, whose retention rate is closer to LUCID's (79–96%), is the sound one.

E.7 What the Concepts Do: Causal Intervention

Alignment with a factor is an associational statement. A stronger requirement for an explanation is that the concepts *carry* the decision: editing a concept should move the policy in the direction the rule predicts. We test this directly on the frozen teacher.

Protocol. Given a held-out state, we decode a concept edit back into the teacher's feature space, $\hat{h} = h + W_d\,(z' - z) \odot \sigma_{\text{feat}}$, and re-run the PPO head on $\hat{h}$; the reported quantity is the rate at which the teacher's arg-max action changes. We intervene at three granularities: a single concept, all literals of the winning clause, and all clauses supporting the chosen action. *Necessity* turns the relevant concepts off; *sufficiency* turns on the concepts of a clause supporting a different action and asks whether the teacher switches to that action. Because $\hat{h}$ must remain a plausible activation, we clamp it to the non-negative orthant, and report the pre-clamp negative fraction (0.32–0.37 throughout) for transparency.

The comparison is against a paired dictionary-direction baseline (DIRB): the same Δz magnitudes applied to the decoder directions of randomly chosen live concepts *outside* the winning clauses. This controls for the sheer size of the perturbation, which a naive comparison against "no intervention" would not; we additionally report an isotropic Gaussian baseline with matched $\|\Delta z\|$. All numbers

Table 12: Causal intervention: change in the teacher's action relative to the paired DIRB baseline, mean ± s.d. over three seeds. The last column counts seeds whose 95% bootstrap CI lies above zero at the clause level. Negative entries mean the concept edit moved the teacher *away* from the predicted action relative to a norm-matched random edit.

Environment	Necessity			Sufficiency	Seeds CI>0
	concept	clause	action		
PixelCartPole	$+0.019 \pm 0.002$	$\mathbf{+0.422 \pm 0.177}$	$\mathbf{+0.575 \pm 0.232}$	$\mathbf{+0.562 \pm 0.107}$	3/3
Boxing	$+0.036 \pm 0.036$	$+0.124 \pm 0.081$	$+0.123 \pm 0.129$	$+0.049 \pm 0.043$	3/3
Pong	$+0.039 \pm 0.022$	$+0.040 \pm 0.020$	-0.008 ± 0.009	$+0.159 \pm 0.031$	3/3
DoorKey-6x6	$+0.017 \pm 0.004$	$+0.038 \pm 0.034$	$+0.029 \pm 0.016$	≈ 0	2/3
Dynamic-Obs-5x5	-0.008 ± 0.024	-0.006 ± 0.024	-0.006 ± 0.026	$\mathbf{-0.023 \pm 0.010}$	1/3

below are differences against DIRB on the same rows, with 95% confidence intervals from a cluster bootstrap over episodes ($B = 2000$).

Results. Table 12 reports necessity at the three levels and sufficiency at the strongest intervention setting.

We call concepts causally effective in an environment when the clause-level necessity effect is significantly positive (95% CI excluding zero) in all three seeds; by this criterion they carry the decision in three of five environments. The criterion measures consistency, not size: the clause-level effect on Pong ($+0.040$) is as small as on DoorKey ($+0.038$), but it is significant in every seed. PixelCartPole is the clearest case: switching off the literals of a winning clause changes the teacher's action 42 percentage points more often than a norm-matched random edit, and the effect is monotone in intervention strength. Crucially, it is not an artefact of a larger perturbation — the concept edit has a *smaller* feature-space displacement than the baseline it beats ($\|\Delta\hat{h}\| = 2.90$ vs. 4.16 at the clause level, 3.60 vs. 4.93 at the action level) while changing the action roughly twenty times as often. Boxing and Pong show the same pattern at a smaller magnitude, with all three seeds significant at the clause level, and PixelCartPole and Pong also show a clean dose–response in sufficiency.

Two environments do not. On DoorKey the effect is present but weak and not stable across seeds at the clause and action levels, and sufficiency is indistinguishable from zero. On Dynamic-Obstacles the sign reverses: in two of three seeds the necessity effect is significantly *negative* at all three levels, and sufficiency is significantly negative in all three seeds. This is not a data artefact — on those runs the rules reproduce the teacher almost exactly (π^{teacher} agreement 1.000, hard fidelity 0.999–1.000) and the shuffled-factor sanity check passes. A surrogate can therefore match a policy's behaviour essentially perfectly while its concepts fail to be the causal substrate of that behaviour.

Finally, we quantify how much of the rule policy's behaviour is actually routed through concepts, as opposed to through the learned per-action bias of the empty clause. In four environments 92–100% of decisions are determined by at least one active concept literal (Dynamic-Obstacles 100%, Pong 99.9%, Boxing 97.0%, PixelCartPole 92.2%). DoorKey is the exception at 47.2%: in the remaining states the action is decided by the learned prior of the universal clauses, which happens to agree with the teacher 99.8% of the time there — the prior encodes "move forward when nothing is detected" and, separately, suppresses the `drop` action, which is useless in DoorKey and is never selected in 25,160 held-out states. This prior is benign only in DoorKey: where it decides the action in the other environments, it agrees with the teacher just 4.9% (Boxing), 30.5% (Pong) and 61.9% (PixelCartPole) of the time, i.e. there it is a failure mode rather than a learned regularity.

E.8 Two Notions of Concept Quality, and Why They Diverge

Taken together, Sections E.6 and E.7 evaluate the same dictionaries along two axes that are usually assumed to coincide: whether concepts *name* something a human specified, and whether concepts *drive* the decision. Across five environments they never coincide.

The environments where LUCID's concepts best match hand-specified factors are exactly those where intervening on them least affects the policy, and vice versa. We report this as an observation rather than a law: five environments cannot establish a systematic trade-off, and at least three confounds are plausible. The factor vocabularies differ in kind (symbolic predicates in MiniGrid, continuous

Table 13: The two evaluations disagree in all five environments. "Aligned" is the monosemantic fraction relative to the CNN baseline; "Causal" is whether the clause-level necessity effect is positive in all three seeds.

Environment	Aligned with factors	Causally effective	Concept-routed decisions
Dynamic-Obs-5x5	yes (30.5 vs. 11.9)	**no** (negative, 2/3 seeds)	100%
DoorKey-6x6	yes (26.1 vs. 7.8)	no (2/3 seeds)	47.2%
PixelCartPole	**no** (9.0 vs. 46.2)	yes (3/3)	92.2%
Pong	**no** (1.1 vs. 4.8)	yes (3/3)	99.9%
Boxing	**no** (2.3 vs. 7.8)	yes (3/3)	97.0%

state variables in PixelCartPole, object coordinates in Atari) and may simply be a better description of gridworld state than of pixel-based control; the environments differ in action-space size, which mechanically inflates intervention effects when there are only two actions, as in PixelCartPole; and all dictionaries come from LUCID, whose objective shapes concepts for imitation rather than for factors. What the divergence does establish is methodological: neither evaluation alone licenses the claim that a surrogate's atoms are the policy's reasons. High behavioural fidelity — 96–100% in every environment we study — constrains neither axis, and on Dynamic-Obstacles a fidelity of 1.000 coexists with concepts that are causally counterproductive.

F Ablation Studies

This appendix reports the sensitivity of the extracted rules to the two extraction thresholds, and their variability across training seeds, in all five environments.

Robustness to the Extraction Thresholds (τ, θ). The two thresholds of Section C.6 play different roles, and we test them separately (Table 14). *Selector threshold* τ. In all five environments no selector probability $p_{a,i,j}$ or $n_{a,i,j}$ lies in $(0.5, 0.9]$, so by Lemma 4 the extracted rule set, and every quantity computed from it, is identical for all $\tau \in [0.5, 0.9]$. *Concept threshold* θ. Changing θ within $[0.5, 0.9]$ can alter the hard action only on states where some rule concept has $c_j^{\text{soft}} \in (0.5, 0.9]$: none on Dynamic-Obstacles, 1.5% of states on DoorKey, 2.3% on Pong, 7.0% on Boxing and 13.4% on PixelCartPole. Hard fidelity is accordingly non-increasing in θ but changes by at most 1.6 points over the whole range, and by at most 0.42 points between $\theta = 0.5$ and our $\theta = 0.66$, which was fixed before this analysis; $\theta = 0.5$ is marginally better in every environment. Theorem 1 holds at every θ with no violating state. PixelCartPole, whose rule concepts are the least polarized (Table 5), is the most θ-sensitive, as expected from the bimodality remark in Section D.

Table 14: Sensitivity to the concept threshold θ at $\tau = 0.66$ (the rule set is identical for every $\tau \in [0.5, 0.9]$), on the held-out states of Table 5. *Hard fid.*: agreement with the teacher (%). *Cert.*: $\Pr[x \in \mathcal{X}_{\text{cert}}]$ (%). *θ-sens.*: states with a rule-concept activation in $(0.5, 0.9]$ (%), the only states whose hard action can depend on θ in this range.

Environment	Metric	$\theta = 0.5$	0.6	**0.66**	0.7	0.8	0.9	**θ-sens.**
Dynamic-Obs-5x5	Hard fid.	100.00	100.00	100.00	100.00	100.00	100.00	0.00
	Cert.	100.00	100.00	100.00	100.00	100.00	100.00	
DoorKey-6x6	Hard fid.	99.97	99.97	99.91	99.84	99.60	99.01	1.54
	Cert.	99.98	99.98	99.92	99.85	99.60	99.02	
PixelCartPole	Hard fid.	97.42	97.14	97.00	96.83	96.42	95.83	13.43
	Cert.	88.34	87.92	87.55	87.28	86.46	84.97	
Pong	Hard fid.	99.87	99.85	99.85	99.84	99.82	99.66	2.34
	Cert.	99.25	99.18	99.15	99.04	98.71	97.85	
Boxing	Hard fid.	96.83	96.81	96.77	96.71	96.56	96.02	6.96
	Cert.	69.40	69.10	68.95	68.72	67.98	66.68	

Empirical Robustness Across Multiple Seeds. To assess the stability of LUCID, we trained it with three seeds (42, 43, 44) and evaluated each model on five evaluation seeds (42–46); Table 15 reports every run. The fidelity of the hard rules is stable: it varies by at most two points within an environment (99.92–99.94% on DoorKey, 99.95–100% on Dynamic-Obstacles, 95.82–97.82% on PixelCartPole, 99.77–99.90% on Pong, 96.34–97.81% on Boxing), and the soft surrogate stays above 99.7% everywhere. Return is equally stable on DoorKey, Pong and Boxing, whereas PixelCartPole remains the sensitive case, with hard-rule returns between 407.1 and 442.5 (81–88% of the teacher). Rule size varies more than fidelity: from 9 to 22 literals on DoorKey and from 8 to 45 on PixelCartPole, with fidelity changing by at most two points. Different seeds therefore recover different, almost equally faithful rule sets; a single extracted rule set is one faithful description of the teacher rather than the unique one.

Table 15: LUCID across three training seeds, each evaluated on five evaluation seeds (42–46). Score and success rate: mean over evaluation seeds; fidelity: agreement with the teacher on the teacher rollouts used for training. Last two columns: mean ± s.d. over the three training seeds.

		Seed 42		Seed 43		Seed 44		Mean ± s.d.	
Environment	**Metric**	Soft	Hard	Soft	Hard	Soft	Hard	Soft	Hard
DoorKey-6x6 (5×200 ep)	Score	0.968	0.968	0.968	0.968	0.968	0.964	0.968 ± 0.000	0.967 ± 0.002
	Success rate (%)	100.0	100.0	100.0	100.0	100.0	99.6	100.0 ± 0.0	99.9 ± 0.2
	Fidelity (%)	100.00	99.92	100.00	99.94	100.00	99.92	100.00 ± 0.00	99.93 ± 0.01
	Concepts	–	12	–	12	–	9	–	11.0 ± 1.4
	Rules	–	6	–	8	–	4	–	6.0 ± 1.6
	Literals	–	16	–	22	–	9	–	15.7 ± 5.3
Dynamic-Obs-5x5 (5×200 ep)	Score	0.891	0.891	0.891	0.891	0.891	0.891	0.891 ± 0.000	0.891 ± 0.000
	Success rate (%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0 ± 0.0	100.0 ± 0.0
	Fidelity (%)	100.00	100.00	100.00	99.95	100.00	100.00	100.00 ± 0.00	99.98 ± 0.02
	Concepts	–	4	–	5	–	3	–	4.0 ± 0.8
	Rules	–	3	–	5	–	3	–	3.7 ± 0.9
	Literals	–	4	–	6	–	3	–	4.3 ± 1.2
PixelCartPole (5×40 ep)	Score	475.5	442.5	481.1	412.9	467.3	407.1	474.6 ± 5.7	420.8 ± 15.5
	Fidelity (%)	99.99	97.13	99.98	97.82	99.97	95.82	99.98 ± 0.01	96.92 ± 0.83
	Concepts	–	10	–	9	–	6	–	8.3 ± 1.7
	Rules	–	9	–	11	–	4	–	8.0 ± 2.9
	Literals	–	45	–	43	–	8	–	32.0 ± 17.0
Pong (5×20 ep)	Score	21.0	21.0	21.0	21.0	21.0	21.0	21.0 ± 0.0	21.0 ± 0.0
	Fidelity (%)	99.90	99.85	99.82	99.77	99.95	99.90	99.89 ± 0.05	99.84 ± 0.05
	Concepts	–	17	–	12	–	14	–	14.3 ± 2.1
	Rules	–	18	–	12	–	14	–	14.7 ± 2.5
	Literals	–	19	–	12	–	14	–	15.0 ± 2.9
Boxing (5×20 ep)	Score	94.1	92.5	95.4	91.2	94.1	94.1	94.5 ± 0.6	92.6 ± 1.2
	Fidelity (%)	99.78	96.80	99.77	96.34	99.79	97.81	99.78 ± 0.01	96.98 ± 0.61
	Concepts	–	43	–	41	–	48	–	44.0 ± 2.9
	Rules	–	53	–	50	–	56	–	53.0 ± 2.4
	Literals	–	93	–	97	–	107	–	99.0 ± 5.9

G Extended Limitations

Several limitations bound the scope of these conclusions.

Scope of the architecture. LUCID is restricted to discrete action spaces, and our attempt to go beyond them failed at two separate points. First, we could not obtain a pixel-based teacher for continuous control: SAC on MountainCarContinuous from pixels (1M steps) did not solve the task under fixed entropy coefficients (0.3, 0.1, 0.01), velocity-based shaping (which was reward-hacked by oscillating in place), potential-based shaping on position and on mechanical energy, or gSDE exploration, where stochastic rollouts reached the goal but the deterministic policy did not. Second, even with a teacher, discretizing the action space does not scale: binning the three action dimensions of CarRacing into five levels each already gives 125 joint actions, far beyond the number of actions for which a per-action DNF layer stays compact. The second obstacle is a limitation of the method rather than of our training budget. To our knowledge, no existing rule-extraction method handles continuous actions from pixel observations: ORCAID (Lopez-Miguel et al., 2026) extracts DNF rules with affine action consequents for continuous control, but from low-dimensional state vectors (at most 17 dimensions), and combining such consequents with concept discovery from pixels remains open. The

DNF representation is also intrinsically flat, and policies with hierarchically decomposable decisions may benefit from hierarchical DNF structures or automated clause summarisation.

The dictionary trades coverage for purity. We prespecified the conjunction "LUCID $\geq$ every baseline on both completeness and %G1" as a single test over the full threshold grid, and it fails in every cell of every environment (0/12 on MiniGrid, 0/36 elsewhere): completeness is consistently below the CNN's, because the sparse dictionary covers fewer factors even where the dimensions it does have are cleaner. We report the failed criterion as stated rather than relaxing it post hoc. Relatedly, LUCID does not beat random-projection or PCA baselines on the threshold-free polysemanticity test in any environment, although that particular comparison is confounded by the selection step it requires (Section E.6).

The factor vocabulary may be the binding constraint outside MiniGrid. On PixelCartPole and Atari, 83–92% of LUCID dimensions align with no factor we defined. On Atari the teacher's CNN features fare only marginally better (4.8–7.8% monosemantic), which suggests the object-coordinate and state-variable vocabularies are too coarse for those domains rather than that the dictionary is uniquely misaligned — but our data cannot separate the two hypotheses, and a richer factor set for Atari is needed to do so.

Intervention effects are not comparable across environments. The action-space size differs by an order of magnitude across our benchmarks (2 actions in PixelCartPole, 3 in Dynamic-Obstacles, 6 in DoorKey and Pong, 18 in Boxing), which mechanically inflates the measured effect where there are fewer alternatives to switch to. We therefore compare each environment only against its own paired baseline, and do not read the ranking across environments as a ranking of concept quality.

The negative result on Dynamic-Obstacles is unexplained. There the intervention effect is significantly negative — editing concepts moves the teacher away from the rule-predicted action relative to a norm-matched random edit — in two of three seeds for necessity and all three for sufficiency, despite a hard fidelity of 0.999–1.000 and a passing shuffled-factor sanity check. We can rule out a data or measurement artefact, and a poorly fitted dictionary: the SAE reconstructs the teacher's standardized features almost exactly on this task (reconstruction loss ≤ 0.01), so concept edits stay within the feature space the teacher uses. We cannot identify the cause; the environment has the fewest concepts in its rules (4) and the fewest live dictionary dimensions (78 of 200) of any benchmark, and its CNN features are the most polysemantic we measured (80.6% G2), which may indicate that its decisions depend on transient obstacle configurations rather than on any stable state factor.

Sample size for the divergence claim. Five environments cannot establish that factor alignment and causal efficacy are systematically in tension. We report the divergence as an observation with three identified confounds (factor vocabulary, action-space size, and a single extraction method), not as a law. *Concept labels are only partly validated.* Labels are evaluated by a blind predictive test on the two MiniGrid tasks only (13 concepts), with a single scorer whose own ceiling is limited (mean positive-control AUC 0.79–0.80) and a different namer per task; no human evaluation was run, and the namers sample stochastically, so exact wordings differ between runs (Appendix H). Labels should be read as hypotheses whose reliability is quantified, not as verified semantics.

Verification. Theorem 1 certifies soft–hard agreement state by state, but it is loose where concepts are weakly binarized (PixelCartPole) or many clauses compete (Boxing): 12–31% of states remain uncertified although only about 3% actually disagree. Its false-clause term keeps only the most violated literal; tightening this term, or training with the margin loss of Remark 5, should raise coverage. The certificate also relates the rules to the soft surrogate rather than to the teacher, and holds only on the states it is evaluated on; verifiable worst-case guarantees (Katz et al., 2017) for deployment remain future work. We also evaluate the certificate only on states visited by the teacher; states reached by the hard policy itself, and perturbed or out-of-distribution states, remain to be tested.

H VLM Concept Labelling: Protocol, Prompts, and Results

This appendix specifies the labelling experiment in enough detail to reproduce it. Code, all rendered images, the raw model responses, and the per-image scores are provided as anonymized supplementary material and will be released with the paper; every output file carries a manifest with the model identifiers, generation parameters, API model version, UTC time of the run, library versions, and the full command line.

H.1 DESIGN

The experiment separates *naming* from *evaluation*. A namer model proposes a label for each concept from examples of states where the concept is ON and OFF. A different scorer model then receives only the label and one *held-out* state at a time, and states how likely the label is to hold in that state. The score is compared with the concept's actual activation, so the evaluation is against ground truth rather than a self-assessment. Neither model sees the extracted rules, the actions a concept gates, heatmaps, or any score from the other model.

Models. Namers: `gemini-3.1-pro-preview` for DoorKey (`thinking_level=HIGH`, temperature at the API default of 1.0, as recommended for Gemini 3 models) and `gpt-5.5-2026-04-23` for Dynamic-Obstacles (`reasoning_effort=medium`, default temperature). Scorer: `deepseek-flash` (DeepSeek V4.1 Flash, OpenAI-compatible API, images sent inline as PNG, temperature 0). Each namer comes from a different developer than the scorer, so the scorer shares neither weights nor training pipeline with the model whose labels it tests. API calls were made on 24–25 September 2026. Because the namers sample stochastically, their labels are not bit-reproducible; we release their raw responses so that the evaluation can be re-run on exactly our labels.

Scope. The protocol was run on the two MiniGrid tasks, whose mechanics and simulator factors allow every label to be checked against ground truth: DoorKey with the Gemini namer (two independent naming runs) and Dynamic-Obstacles with the GPT namer (one run). PixelCartPole and the Atari games were not labelled, for API-budget reasons and because on PixelCartPole the scorer's positive control was near chance (AUC 0.51 and 0.60); the code runs unchanged on them.

H.2 DATA

States. For each environment we roll out the frozen teacher (deterministic actions) from environment seed 2000, which differs from the seeds used for training (0) and for the held-out evaluation data (1000): 2000 episodes for the MiniGrid tasks, 20 episodes (every 5th step stored) for PixelCartPole, and 6 episodes (every 4th step) for Pong and Boxing. At every stored step we record the observation, the teacher's CNN features (SB3 `extract_features`, including image normalisation), LUCID's concept activations z, and, where available, the simulator factors. As a sanity check, the LUCID soft policy agrees with the teacher on 99.99% of these steps on DoorKey.

Concepts. We evaluate the concepts that appear as literals in the extracted rules of the seed-42 checkpoint of each environment. A concept is ON when its bottleneck output $c_j > \theta = 0.66$, as in the rules.

Split. Exact duplicate observations are removed. The remaining unique states are split at random (seed 2000) into a naming set (30%) and a test set (70%); no observation occurs in both. For each concept:

- *Naming examples*: 8 ON states, chosen as k-means medoids ($k = 8$, 10 initialisations) of the teacher's CNN features among the (at most) 1000 most strongly activating ON states of the naming set; and 8 OFF states, chosen the same way from at most 1000 randomly sampled OFF states.
- *Test states*: 20 ON and 20 OFF states sampled at random from the test set. Concepts with fewer than 5 test states in either class are excluded and reported (3 of 12 concepts in DoorKey).

Views. Each image shows one state. For MiniGrid it is the agent's 7×7 ego-centric view, which is the only input of the concept; for PixelCartPole and Atari it is the full rendered frame. Images are resized with nearest-neighbour interpolation so that the longer side is 336 pixels. The same view type is used for naming and for testing.

H.3 PROMPTS

<ENV_CONTEXT> is replaced by the environment description below, <image> marks where an image is inserted, and <LABEL> is the label with underscores replaced by spaces.

Naming prompt (one call per concept, 16 images).

```
You are analysing one internal feature ("concept") of an agent that plays a video game.

ENVIRONMENT: <ENV_CONTEXT>

Below are 8 states where the concept is ON, followed by 8 states where it is OFF.

ON example 1:
<image>
...
ON example 8:
<image>
OFF example 1:
<image>
...
OFF example 8:
<image>

Task.
Step 1: for EVERY image above, write one short line describing what is visible and where each
object is (for a grid view: relative to the agent -- in front, left, right, far or near; for a
full frame: positions and tilts of the objects).
Step 2: find the property that holds in all (or almost all) ON images and fails in most OFF
images. Use only what you can see. Do not reuse wording from these instructions unless it
really is the distinguishing property.
Step 3: end your answer with JSON on the last line:
{"label": "short_snake_case_description_of_that_property", "evidence": "one sentence"}
```

The label is taken from the last JSON object in the response.

Scoring prompt (one call per label and test image).

```
ENVIRONMENT: <ENV_CONTEXT>

The image shows one state:
<image>
Statement: "<LABEL>"

Is the statement true in this state?
Reply with ONLY an integer from 0 to 100: your probability that the statement is true in
this state.
```

The first integer in the response, divided by 100, is the score. Unparsable responses receive 0.5 and are counted (0.1% of calls on DoorKey, none on Dynamic-Obstacles).

Environment descriptions.

```
[MiniGrid-DoorKey-6x6-v0]
A small grid world containing a yellow key, a yellow door in a wall, and a green goal
square. Each image is the agent's 7x7 ego-centric view: the agent is the red triangle in the
bottom-centre cell and always faces the top of the image; cells it cannot see are black; an
object the agent is carrying is drawn in the agent's own cell.

[MiniGrid-Dynamic-Obstacles-5x5-v0]
A small grid world containing moving blue balls and a green goal square. Each image is the
agent's 7x7 ego-centric view: the agent is the red triangle in the bottom-centre cell and
always faces the top of the image; cells it cannot see are black; an object the agent is
carrying is drawn in the agent's own cell.

[PixelCartPole]
CartPole: a pole is hinged on a cart that moves left or right along a track. Each image is
the rendered frame of the scene.

[PongNoFrameskip-v4]
Atari Pong: the agent controls the right paddle, the opponent the left paddle, and a small
ball moves between them. Each image is the rendered game screen.

[BoxingNoFrameskip-v4]
Atari Boxing, top-down view of a ring: the agent controls the white boxer, the opponent is
black. Each image is the rendered game screen.
```

H.4 STATISTICS

Label evaluation. For concept c with test states $s_1, \ldots, s_{40}$ and ground truth $y_i = \mathbb{1}[c_c(s_i) > \theta]$, we compute the ROC AUC of the scorer's probabilities for the true label of c (ties count $\frac{1}{2}$). As a null, the same test states are scored with three *distractor* labels: labels that the same namer gave to other concepts of the same environment, textually different from the label of c, drawn with seed c. The statistic is

$$\Delta = \frac{1}{|C|} \sum_{c \in C} \Big(\mathrm{AUC}_c(\ell_c) - \tfrac{1}{3} \sum_d \mathrm{AUC}_c(\ell_d) \Big),$$

and its p-value comes from a permutation test (10,000 permutations) in which, independently for each concept, the "true" label is drawn uniformly from $\{\ell_c\} \cup \{\ell_d\}$; $p = (1 + \#\{\Delta^* \geq \Delta\})/(1 + 10{,}000)$. We also report balanced accuracy at a threshold of 0.5.

Positive control (scorer ceiling). To separate a wrong label from a scorer that cannot read the images, the scorer is also given ground-truth statements about simulator factors on balanced test states (20 per class, same view type). An AUC near 1 means the scorer can verify that property from a single image; the mean over factors bounds what any label can reach. Statements (factors that are not visible in a single image, such as velocities or the agent's absolute direction, are excluded; `goal_visible` is excluded in DoorKey-6x6 because it coincides exactly with `door_open`):

Environment	Factor	Statement
DoorKey	`has_key`	The agent is carrying the key.
	`door_open`	The door is open.
	`key_in_front`	The key is in the cell directly in front of the agent.
	`door_in_front`	The door is in the cell directly in front of the agent.
	`wall_ahead`	A wall is in the cell directly in front of the agent.
Dynamic-Obstacles	`obstacle_ahead`	A blue ball is in the cell directly in front of the agent.
	`obstacle_left`	A blue ball is in the cell directly to the left of the agent.
	`obstacle_right`	A blue ball is in the cell directly to the right of the agent.
PixelCartPole	pole angle > 0	The pole is leaning to the right.
	$x_{\text{cart}} > 0$	The cart is to the right of the centre of the screen.

Atari was not labelled, so no positive control was run for it.

Label–factor consistency. On all unique states we compute the AUC of each concept's activation c_c against each binarised factor. A concept is *aligned* with factor f if $\mathrm{AUC} \geq 0.8$ for f and no other factor has a higher AUC. For aligned concepts we embed the label and every factor statement with four sentence encoders (`all-MiniLM-L6-v2`, `bge-base-en-v1.5`, `e5-base-v2`, `gte-base`), rank the statements by cosine similarity to the label, and report the mean reciprocal rank of the aligned factor against a Monte-Carlo null (10,000 draws of a uniformly random factor per concept). Concepts aligned only negatively ($\mathrm{AUC} \leq 0.2$) are listed but not scored, because sentence encoders handle negation poorly.

H.5 RESULTS

Table 16: Blind predictive evaluation of VLM labels (scorer `deepseek-flash`). Ceiling: mean positive-control AUC of the scorer. Combined: Fisher's method over the two permutation tests.

Environment	Namer	Concepts	Ceiling	AUC (label)	AUC (distr.)	Δ	p
DoorKey	Gemini 3.1 Pro	9/12	0.80	0.72	0.52	0.20	0.002
Dynamic-Obstacles	GPT-5.5	4/4	0.79	0.69	0.43	0.25	0.017
Combined		13					< 0.001

H.6 CONCEPT LABELS

Table 17 lists the labels proposed for the concepts in the extracted rules of the seed-42 DoorKey model (the model of Tables 1–10). Three of its twelve rule concepts (f_{129}, f_{182}, f_{262}) are ON in fewer

Table 17: VLM labels for the DoorKey rule concepts (seed 42), underscores shown as spaces. *AUC*: blind predictive AUC of the run-1 label; *Distr.*: mean AUC of its three distractor labels; *Factor*: simulator factor with AUC ≥ 0.8 against the concept's activation (all unique states), if any.

ID	Label (run 1)	Label (run 2)	AUC	Distr.	Factor
f_{23}	key or door in agent row	target object in same row	0.70	0.39	wall_ahead (0.80)
f_{54}	key directly in front of agent	key directly in front	0.68	0.49	key_in_front (0.98)
f_{87}	yellow object immediately right	yellow object right same row	0.84	0.50	–
f_{92}	door visible and key not on floor in front or right	yellow door visible at corner and key not in front or right	0.64	0.45	–
f_{131}	key directly in front	key directly in front	0.88	0.63	key_in_front (1.00)
f_{170}	yellow object immediately right	yellow object to the right in same row	0.68	0.52	–
f_{237}	key is adjacent left or front and separated from door	blocked on the right	0.68	0.57	key_in_front (0.84)
f_{258}	yellow object right of agent in same row	yellow object immediately right	0.90	0.65	–
f_{294}	clear path to closed door without key	visible closed door key not carried and not adjacent	0.46	0.49	key_in_front (0.82)

than 1.3% of states and could not be labelled and tested. The two runs of the namer are independent samples at temperature 1.0; the AUC columns are from the blind test of run 1.

Table 18 lists the GPT-5.5 labels for all four rule concepts of the seed-42 Dynamic-Obstacles model. f_{32} and f_{133} are ON in 9% of states and have identical factor profiles; accordingly the label of f_{133} predicts f_{32} better than its own label does (AUC 0.72 vs. 0.59). The factor-aligned concepts again receive matching labels: f_{49} (`obstacle_left`, AUC 0.95) is labelled "blue ball not only on right", and f_{146}, which is OFF when an obstacle is ahead (AUC 0.19), is labelled "goal visible, balls on right".

Table 18: GPT-5.5 labels for the Dynamic-Obstacles rule concepts (seed 42), columns as in Table 17; $\neg$ marks a concept that is OFF when the factor holds.

ID	Label	AUC	Distr.	Factor
f_{32}	blue ball right, no goal ahead or left	0.59	0.44	–
f_{49}	blue ball not only on right	0.76	0.37	obstacle_left (0.95)
f_{133}	goal absent or right of agent	0.72	0.42	–
f_{146}	goal visible, balls on right	0.68	0.51	$\neg$obstacle_ahead (0.19)

DoorKey in detail. Across the 9 evaluated concepts the true labels reach a mean AUC of 0.72 (balanced accuracy 0.66) against 0.52 for distractors ($\Delta = 0.20$, $p = 0.002$), i.e. about 90% of the scorer's ceiling of 0.80 (per-factor ceilings: `has_key` 0.91, `key_in_front` 0.88, `door_in_front` 0.76, `door_open` 0.74, `wall_ahead` 0.73). The true label beats the mean of its distractors for 8 of 9 concepts; the exception is f_{294} (`clear_path_to_closed_door_without_key`, AUC 0.46), whose label misses the factor it aligns with most (`key_in_front`, AUC 0.82). Distractors score high exactly where concepts are near-duplicates: the label of f_{87} predicts f_{258} with AUC 0.91 and the label of f_{54} predicts f_{131} with AUC 0.88, matching their near-identical factor profiles. Only 0.1% of scorer responses could not be parsed.

Preliminary runs with open-weight models. Before settling on the models above we ran the same protocol on DoorKey with open-weight VLMs that fit a single 24 GB GPU. As namers, Qwen2.5-VL-7B-Instruct and InternVL3-8B often repeated the same label for different concepts (6 distinct labels for the 7 concepts Qwen labelled, two responses being truncated; 7 distinct labels for 9 concepts from InternVL3). As a scorer, Qwen3-VL-8B-Instruct reached positive-control AUCs of 0.98 (`has_key`) and 0.79 (`key_in_front`) but 0.54 (`door_open`), 0.52 (`door_in_front`) and 0.36 (`wall_ahead`); with this scorer, InternVL3 labels were no more predictive than distractor labels (mean AUC 0.53 vs. 0.51, $p = 0.35$). We report these runs because they show that the scorer's ability to read the view, not only the label, limits what this kind of evaluation can establish, which is why the positive control is part of the protocol.

Scorer selection. On DoorKey, `gemini-2.5-pro` reached a mean positive-control AUC of 0.75 with the minimum thinking budget (128 tokens) and 0.87 with 1024 tokens, and `deepseek-flash`

reached 0.80 (0.79 on Dynamic-Obstacles). We use `deepseek-flash` because it comes from a different developer than either namer, which removes the shared-model-family concern, at the cost of a somewhat lower ceiling.

Label stability. The DoorKey namer was run twice. Labels of the same concept across the two runs are more similar than labels of different concepts (DoorKey: AUC 0.79–0.82 across the four sentence encoders, permutation $p < 0.001$), although the exact wording rarely repeats (1 of 9 identical).

I Implementation Details

All experiments run on a single NVIDIA RTX 4090 GPU; a training run takes about half an hour on average. LUCID is trained on the teacher's CNN features collected from teacher rollouts, standardized per dimension with the mean and standard deviation of the collected features. The collected states are split at random into 90% training and 10% validation (seeded by the training seed), and we keep the checkpoint with the highest validation action accuracy of the soft surrogate, breaking ties by lower training loss. The SAE is initialized with Xavier-uniform weights and unit-norm decoder columns, and its sparse code enters the bottleneck without further normalization. All terms of Eq. (3) are optimized jointly with Adam, with separate learning rates for the SAE, the bottleneck and the logic layer and cosine annealing over the full run; the cross-entropy uses uniform class weights. Table 19 lists every setting.

Table 19: Hyperparameters of the reported LUCID models (training seed 42). Values shared by all environments are given once. A: number of actions modelled (on DoorKey the teacher never selects `Done`, so it is not modelled; on Dynamic-Obstacles, seeds 43 and 44 model all 7 MiniGrid actions).

Description	Symbol	DoorKey-6x6	Dynamic-Obs-5x5	PixelCartPole	Pong	Boxing
Loss coefficients (Eq. 3)						
SAE reconstruction	λ_{rec}		1.0			
SAE L1 sparsity	λ_1		5×10^{-3}			
Action cross-entropy	λ_{CE}		5.0			
Bimodality (max. value)	λ_{bim}	0.3	0.4	0.3	0.3	0.3
Selector L0 (rule sparsity)	λ_{L0}	10^{-4}	4.5	10^{-4}	100	100
Selector entropy (max. value)	λ_{ent}	0.015	0	0.005	0.3	0.3
Discreteness schedule ω_t						
Warmup	T_{w} (epochs)		30			
Linear ramp	T_{r} (epochs)	120	60	200	200	200
Optimization						
Training epochs		600	300	900	600	600
Batch size			256			
Learning rate (SAE / bottleneck / logic)			10^{-3} / 10^{-3} / 3×10^{-3} (Adam, cosine annealing to 10^{-5})			
Gradient clipping (global norm)			5.0			
Train / validation split			90% / 10%			
Architecture and initialization						
Dictionary size	D	300	200	512	512	512
Active concepts per state	k	50	20	50	50	50
Clauses per action	n_a	15	10	20	10	10
Actions	A	6	3	2	6	18
Initial bottleneck sharpness / shift	α_j / β_j		1.5 / 0			
Initial selector logits (ignore logit = 0)	$w^p_{a,i,j}, w^n_{a,i,j}$		$\mathcal{N}(-3, 0.01^2)$, i.e. $u_{a,i,j} \approx 0.91$			
Initial clause bias	$b_{a,i}$		2.0			
Extraction						
Selector threshold (Prop. 1)	τ		0.66			
Concept threshold	θ		0.66			
Clause-bias cap	$b_{\max}$		5			
Log-stability constant	ϵ		10^{-8}			

The bimodality and entropy weights follow the schedule ω_t of Section C.5: they are zero for the first $T_{\text{w}} = 30$ epochs and then grow linearly over T_{r} epochs to the values in Table 19, while λ_{L0} is constant. This lets the SAE settle into a sparse code under reconstruction and imitation before discreteness pressure is applied, within a single joint objective and without freezing any module. The discreteness coefficients (λ_{bim}, λ_{L0}, λ_{ent}, T_{r}) were tuned by hand for each environment, because they mainly set the size of the extracted rule set: the same value yields rules of very different size on environments that differ in dictionary size, number of actions and clause budget. For each environment we kept the most compact rule set whose soft surrogate retained high action accuracy on the 10% validation split

of the training data; configurations were also compared by the online success of the hard rules, so those numbers are not fully independent of tuning; the independent rollouts of Section E.5 were not used at any stage. These coefficients also change which term polarizes the selectors: at $\lambda_{\mathrm{L0}} = 10^{-4}$ the L0 term is numerically negligible, so on DoorKey and PixelCartPole the entropy penalty does this work; on Dynamic-Obstacles the entropy penalty is off and the L0 term alone polarizes the selectors; on Atari both are active. The reported models use training seed 42; seeds 43 and 44 are analysed in Appendix F.