CERTIFIED ZERO-SHOT TRANSFER OF INFORMATION BOTTLENECK POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Compression is known to improve policy transfer, yet when and why a compressed policy transfers to a new task remains poorly understood. Current empirical approaches evaluate compressed policies on held-out environments, which reveals the outcome of a transfer but not its cause, while theoretical approaches either bound value changes from the two tasks without reference to the policy or relate compression to lost value within a single task. This work aims to understand when and why a policy compressed with the information bottleneck (IB) on a source task transfers to a target task. Given models of both tasks, we answer when with a sufficient condition for near-optimal target return, and why with its terms, each of which names one cause: compression on the source, representation shift, or disagreement between the tasks' optimal policies. We evaluate the condition in two settings. On tabular grid-worlds, where the IB is solved exactly, it certifies 88.6% of the successful transfers across 13,200 configurations with no false certifications. On neural IB policies trained on MiniGrid images, where the learned representation can itself shift between tasks, it remains sound, and its terms separate failures caused by representation shift from those caused by task disagreement.

1 INTRODUCTION

Zero-shot policy transfer (ZSPT), training a policy on a source task and deploying it directly on a target task (Taylor et al., 2007), is essential for real-world reinforcement learning (RL), where interactions are costly, unsafe, or infeasible (Dulac-Arnold et al., 2021). A policy optimal on the source can be suboptimal on the target, since agents overfit to task-specific cues (Cobbe et al., 2019) and dynamics or rewards may change. Compression is known to improve transfer by discarding task-irrelevant information (Igl et al., 2019). Given the models of both tasks, can we guarantee that a compressed policy transfers near-optimally? And when it does not, can we tell whether compression, representation shift, or task disagreement is responsible?

When and why a compressed policy transfers remains poorly understood. Compressed policies are evaluated ad-hoc on held-out environments, which reveals the outcome of transfer but not its cause (Igl et al., 2019). Bisimulation metrics bound the value lost to a change of task from the two MDPs alone, without reference to a policy or its compression (Ferns et al., 2004). State abstraction and imitation bounds relate compression or imitation error to lost value, but within a single task (Abel et al., 2019). Neither tells us what caused a compressed policy to succeed or fail on a given target.

In this work, we give a theoretical guarantee for when a compressed policy transfers based on a state-policy information bottleneck (Tishby et al., 2000). It compresses the state space while imitating the source-optimal action. The guarantee is a sufficient condition for near-optimal return on the target. Transfers that satisfy it are certified to succeed, and the others may still succeed (Figure 1, right). The condition compares a bound on the transfer error, built from how far the policy is from optimal behaviour on the target and measured through the bottleneck, with the return the source-optimal policy keeps on the target. The bound splits into one term per cause of failure (compression, representation shift or task disagreement), which lets us attribute a failed transfer to the terms that dominate it. We evaluate the condition first on tabular grid-worlds, where the bottleneck is solved exactly, and then on neural policies that learn a representation from images, where representation shift enters. Our contributions are:

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.

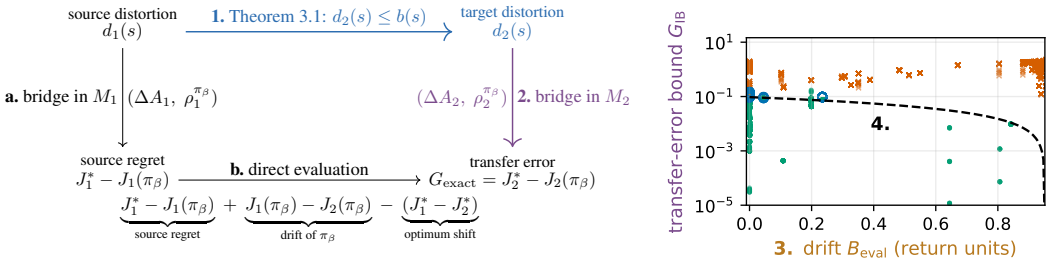

Figure 1: **Certifying and explaining transfer.** *Left:* from the compressed policy's source distortion, the budget (1) bounds its target distortion, and the bridge (2) turns this into the bound G_{IB} on its transfer error; compression enters only at step 1. The classical route (a, then b) has no term involving compression. *Right:* each tabular transfer placed by G_{IB} and the drift B_{eval} (3). Transfers below the dashed line (4) are certified to keep at least 90% of the optimal return; none of the 809 certified transfers fails (green: success, orange: failure, blue rings: certified only by the direct bound A').

- a bound on the transfer error in terms of how far the IB policy is from target-optimal behaviour, split into terms that each name a cause: compression, representation shift, or task disagreement weighted by the decoder's confidence. The compressed policy enters its budget only through quantities measured on the source, and the target only through where the tasks' optimal policies disagree. It holds on all 13,200 tabular configurations, with median looseness $1.89\times$.
- a sufficient condition for near-optimal transfer that compares this bound with the return the source-optimal policy keeps on the target. It certifies 88.6% of the near-optimal transfers, with zero false certifications.
- an interpretability analysis on neural policies, for which the condition remains sufficient on a sampled policy (zero false certifications on 210 pixel configurations), allowing its terms to attribute each failed transfer to one of these causes.

2 Background

We consider a source task, the finite MDP $M_1 = (\mathcal{S}, \mathcal{A}, P_1, r_1, \gamma, \mu)$, and a target task $M_2 = (\mathcal{S}, \mathcal{A}, P_2, r_2, \gamma, \mu)$. They share the state space $\mathcal{S}$, the set of $m = |\mathcal{A}| \ge 2$ actions, the discount $\gamma \in [0, 1)$ and the initial-state distribution μ, and differ only in their transition kernels P_1, P_2 and rewards r_1, r_2 (and, in §5, in how states are observed). A policy $\pi(a \mid s)$ has expected discounted return $J_1(\pi) = \mathbb{E}_{M_1,\pi,\mu}\big[\sum_{t\ge 0} \gamma^t r_1(s_t, a_t)\big]$ on the source and $J_2(\pi)$, defined the same way with M_2 and r_2, on the target. We write π_1^* and π_2^* for optimal policies of M_1 and M_2, and $J_1^* = J_1(\pi_1^*)$ and $J_2^* = J_2(\pi_2^*)$ for their optimal returns. The drift of a fixed policy π is $J_1(\pi) - J_2(\pi)$, how much its return changes when it is moved unchanged from the source to the target.

In ZSPT, a policy π is trained on the source M_1 and deployed directly on the target M_2 without further training. Given a pairwise *instance* (M_1, M_2, π), composed of a source–target pair and a policy, the transfer succeeds at tolerance $\varepsilon \in (0, 1)$ when its *transfer error*, the regret of π on the target $G_{\text{exact}}(\pi) := J_2^* - J_2(\pi) \ge 0$ (Kakade, 2001), satisfies $G_{\text{exact}}(\pi) \le \varepsilon J_2^*$ (for $J_2^* > 0$).

To learn a compressed policy for ZSPT, we use the state-policy information bottleneck. An encoder $p(z \mid s)$ compresses the state into a code z, a decoder $q(a \mid z)$ maps the code to an action, and the resulting IB policy is defined as $\pi_\beta(a \mid s) := \sum_z p(z \mid s) q(a \mid z)$, with transfer error $G_{\text{exact}} := G_{\text{exact}}(\pi_\beta)$. The pair (p, q) is trained on the source to imitate the source-optimal policy π_1^* by minimising $I(S; Z) - \beta I(Z; A)$, where $A \sim \pi_1^*(\cdot \mid S)$, I is mutual information, and $\beta \ge 0$ trades compression against imitation (smaller β compresses more). We measure how well the bottleneck reproduces a reference policy π at state s by its *distortion* $d(\pi; s) := \sum_z p(z \mid s)\, D_{\text{KL}}\pi(\cdot \mid s) q(\cdot \mid z)$. With each task's optimal policy as reference, $d_1(s) := d(\pi_1^*; s)$ is the *source distortion*, which training tries to keep small, and $d_2(s) := d(\pi_2^*; s)$ the *target distortion*, which drives the transfer error.

3 THEORY: CERTIFYING THE TRANSFER OF A COMPRESSED POLICY

Our goal is to understand what makes a compressed policy trained on the source succeed or fail on a target. Prior work bounds the regret at home, converting imitation error to return within the source (Ross & Bagnell, 2010), and carries it to the target with a simulation-lemma bound (Kearns & Singh, 2002), charging the task shift in return units wherever the models differ, even where the optimal action does not change (route a→b in Figure 1, left). Return alone cannot attribute the outcome to its causes. First, it hides which components of the compression drive the transfer error: the bottleneck may imitate the source-optimal policy poorly or commit too confidently to actions the target does not want, and the target's rewards and dynamics can make such mistakes cheaper or costlier. We bound the transfer error with terms that separate these causes (§3.1–3.2; arrows 1 and 2). Second, it hides how useful the source-optimal policy remains on the target, where even without compression it may lose much of its return; the drift quantifies this loss and lower-bounds the target optimum (§3.3; 3). Section 3.4 combines the two into a sufficient condition for near-optimal transfer.

3.1 THE DISTORTION BUDGET: BOUNDING THE TARGET DISTORTION

Whether a compressed policy transfers depends on what it kept and discarded on the source. To attribute the outcome to a cause, we therefore work with the distortions d_2 of §2 rather than the return (arrow 1 in Figure 1, left): distortion can be split into contributions of the compression that generates it, whereas a return cannot. The *target distortion* $d_2(s) := d(\pi_2^*; s)$ measures how far the encoder and decoder are from the target-optimal policy at s; if it vanishes at every state, the transfer error is zero. The *source distortion* $d_1(s) := d(\pi_1^*; s)$ measures how much imitation of the source-optimal policy the compression gave up. To link the two, the first step bounds $d_2(s)$ with a budget $b(s)$ composed of $d_1(s)$ and further terms (Theorem 3.1). The bound is not needed to estimate $d_2(s)$, which is computable; its purpose is to split it into these contributions, which is what lets us say which part of the compression a transfer outcome is due to. The decomposition costs about 1% in the final bound at the median (Appendix D).

The natural first candidate for the gap between the two distortions is how much the tasks disagree: where both optimal policies agree, $d_2(s) = d_1(s)$, so any excess comes from disagreement. We measure it by the *task disagreement* $T(s) := \frac{1}{2}\|\pi_1^*(\cdot \mid s) - \pi_2^*(\cdot \mid s)\|_1 \in [0, 1]$, the total variation between the tasks' optimal policies. Disagreement alone cannot price the excess, however: no bound $d_2(s) \le d_1(s) + f(\mathrm{div}(\pi_1^*, \pi_2^*))$, with f finite and div any divergence between the optimal policies at s, holds for all decoders (Proposition C.1). If the decoder gives probability zero to an action that π_2^* needs and π_1^* does not, $d_2(s)$ is infinite while $d_1(s)$ and every such divergence stay finite. The disagreement must therefore be weighted by how confidently the decoder commits, since a confident decoder is expensive to be wrong about. We measure this by the *decoder sensitivity* $\bar{L}(s) := \sum_z p(z \mid s) L(z)$ with $L(z) := \max_a[-\log q(a \mid z)]$. With the binary entropy $h(t) := -t \log t - (1-t)\log(1-t)$, the budget $b(s)$ takes the following form.

Theorem 3.1. Pointwise source-to-target distortion bound For every state s, with $m \ge 2$ and positive decoder probabilities,

$$d_2(s) \le b(s) := \underbrace{d_1(s)}_{\text{source distortion}} + \underbrace{2\,T(s)\,\bar{L}(s)}_{\substack{\text{task disagreement} \\ \times \text{ decoder sensitivity}}} + \underbrace{h(T(s)) + T(s)\log(m-1)}_{\substack{\text{entropy correction:} \\ \text{which action instead}}}. \tag{1}$$

Proof sketch. Expanding both divergences against the same decoder row for each z gives $d_2(s,z) - d_1(s,z) = [H(\pi_1^*) - H(\pi_2^*)] + \sum_a (\pi_1^* - \pi_2^*)(a \mid s) \log q(a \mid z)$. The second term is at most $\|\pi_1^* - \pi_2^*\|_1 \max_a |\log q(a \mid z)| = 2T(s)L(z)$ by Hölder, and the first is at most $h(T(s)) + T(s)\log(m-1)$ by the Fannes–Audenaert inequality. Averaging over $z \sim p(\cdot \mid s)$ gives equation 6. See Appendix B.1 for the full proof. □

The target distortion at a state is thus the source distortion plus a price for disagreeing there. The encoder and decoder enter the budget only through $d_1(s)$ and $\bar{L}(s)$, quantities measured on the source that training controls, and the tasks only through $T(s)$. The first term is the price already paid by compressing on the source. Averaged over states it is the distortion term of the bottleneck objective (Tishby et al., 2000), so it is what training minimises. Where the tasks disagree, π_β can imitate the source perfectly and still pick the wrong action for the target, and the second term prices

this. The entropy $h(T(s))$ accounts for the cost of not knowing which action the target switches to, and $T(s)\log(m-1)$ for the cost induced by the number of actions it could switch to. A unit of disagreement at s thus costs up to $2\bar{L}(s) + \log(m-1)$ plus the binary-entropy allowance, which separates the agent's vulnerability to a change of task, set by its decoder, from the change itself, $T(s)$. Compression acts on both: it typically raises d_1 and softens the decoder, lowering $\bar{L}$. Where the tasks disagree, a softer decoder is therefore protective; where they agree, only d_1 matters.

3.2 The bridge: bounding the transfer error by the target distortion

Theorem 3.1 bounds how closely π_β imitates π_2^*, not how much return π_β loses on the target. The same behaviour can earn very different returns on two tasks that differ only in where the reward sits. A distortion matters little where every action earns about the same, and a lot where a wrong action is costly and π_β often goes. The bridge (2) therefore converts the target distortion into a bound on the transfer error G_{exact}, weighting each state by two properties of the target MDP. First, the advantage $A_2^*(s,a) := Q_2^*(s,a) - V_2^*(s) \le 0$ measures what action a costs at s, and the advantage range $\Delta A_2(s) := \max_a A_2^*(s,a) - \min_a A_2^*(s,a)$, the gap between the best and worst action at s, is used to measure how much a wrong action at s costs. Second, the occupancy $\rho_2^{\pi_\beta}(s)$ is the discounted probability, normalised to sum to one, of visiting state s while following π_β in M_2 from the initial state distribution μ (Appendix A). Both come from the target's rewards and dynamics, and the behaviour of the IB policy alone cannot replace them (Proposition C.2).

Write $O(s) = \arg\max_a Q_2^*(s,a)$ for the set of optimal actions at s. We assume that π_2^* is supported on it, $\pi_2^*(O(s) \mid s) = 1$, which is the case for any tie-breaking among optimal actions. For the optimal reference the advantage range is the largest action gap, $\Delta A_2(s) = \max_a[V_2^*(s) - Q_2^*(s,a)]$.

Lemma 3.1. Distortion-to-return bridge Assume that the target-optimal policy is supported on the optimal actions, $\pi_2^*\big(\mathcal{O}(s) \,\big|\, s\big) = 1$ for every s, then for the deployed policy $\pi_\beta(a \mid s) = \mathbb{E}_{z\sim p(\cdot|s)}\big[q(a \mid z)\big]$ with any mixing distribution $p(\cdot \mid s)$,

$$J_2^* - J_2(\pi_\beta) \le \tfrac{1}{1-\gamma}\,\mathbb{E}_{s\sim\rho_2^{\pi_\beta}}\big[\Delta A_2(s)\big(1 - e^{-d_2(s)}\big)\big]. \tag{2}$$

Proof sketch. At each s, convexity of the divergence gives $D_{\mathrm{KL}}\pi_2^*(\cdot|s)\pi_\beta(\cdot|s) \le d_2(s)$; since π_2^* puts all its mass on $\mathcal{O}(s)$, data processing to the partition $\{\mathcal{O}(s), \mathcal{O}(s)^c\}$ gives $\pi_\beta(\mathcal{O}(s)^c \mid s) \le 1 - e^{-d_2(s)}$; the action gap is zero on $\mathcal{O}(s)$ and at most $\Delta A_2(s)$ elsewhere, so the expected gap under $\pi_\beta(\cdot \mid s)$ is at most $\Delta A_2(s)(1 - e^{-d_2(s)})$. The performance-difference lemma turns the occupancy-weighted expected gap into the regret. See Appendix B.2 for the full proof. □

Because Theorem 3.1 holds at every state and $x \mapsto 1 - e^{-x}$ is increasing, $d_2(s)$ may be replaced by its budget inside the expectation, with no change of measure.

Theorem 3.2. Transfer return bound Under the hypotheses of Theorem 3.1 and Lemma 3.1,

$$J_2^* - J_2(\pi_\beta) \le G_{\text{IB}} := \tfrac{1}{1-\gamma}\,\mathbb{E}_{s\sim\rho_2^{\pi_\beta}}\big[\Delta A_2(s)\big(1 - e^{-b(s)}\big)\big]. \tag{3}$$

Proof. By Theorem 3.1, $d_2(s) \le b(s)$ at every state s. Since $x \mapsto 1 - e^{-x}$ is increasing on $[0,\infty]$ and $\Delta A_2(s) \ge 0$, $\Delta A_2(s)(1 - e^{-d_2(s)}) \le \Delta A_2(s)(1 - e^{-b(s)})$ at every s; taking the expectation under $\rho_2^{\pi_\beta}$ and using equation 2 gives equation 3. □

G_{IB} is the vertical axis of Figure 1 (right). How the tasks' dynamics and rewards differ enters G_{IB} through ΔA_2 and the occupancy as well as T: a change of slip probability that preserves the optimal actions leaves $T(s) = 0$ and $d_2(s) = d_1(s)$, yet changes ΔA_2 and the occupancy, and G_{IB} registers it.

3.3 Lower-bounding the target's optimal return with the drift B_{eval}

Theorem 3.2 bounds the transfer error in absolute return, but whether a loss is small depends on how much return there is to lose: a transfer error of 0.05 is negligible where $J_2^* = 1$ and fatal where $J_2^* = 0.06$. Near-optimality is therefore relative to the target optimum, and the third step (3) supplies the scale against which G_{IB} is compared. Any lower bound $\ell \le J_2^*$ serves, since underestimating

J_2^* only makes the condition stricter, including J_2^* itself, which is available once the target has been solved for π_2^*.[1] We instead take the scale from the uncompressed policy.

Definition 3.1. Drift and scale The drift of the source-optimal policy, the *reference-policy drift*, and the scale are

$$B_{\text{eval}} := [J_1^* - J_2(\pi_1^*)]_+, \qquad \ell := J_1^* - B_{\text{eval}} = \min\{J_1^*, J_2(\pi_1^*)\} \le J_2^*. \tag{4}$$

B_{eval}, the horizontal axis of Figure 1 (right), is the drop in return that π_1^* suffers when moved to the target unchanged, clipped at zero since the target may be easier. It needs one policy evaluation of π_1^* in M_2. The following two facts about ℓ will be useful to prove the near-optimal certificate on the next section:

Lemma 3.2. Lower bound on the target optimum Given the scale ℓ defined in 4, one has

$$\ell = \min\{J_1^*,\ J_2(\pi_1^*)\},\ \text{and}\ \ \ell \le J_2^* \tag{5}$$

Proof sketch. $J_1^* - [J_1^* - J_2(\pi_1^*)]_+ = \min\{J_1^*, J_2(\pi_1^*)\}$, and $J_2(\pi_1^*) \le J_2^*$ because π_1^* is a policy of M_2 and J_2^* is the best return in M_2. See Appendix B.3 for the full proof. □

Appendix B.5 derives B_{eval} via the drift identity of Kearns & Singh (2002), and Appendix E compares it with looser bounds. We prefer it to J_2^* for two reasons. It measures, in return units, how much the task changed for a policy unrelated to compression, so it does not depend on β: along that axis, compression moves a transfer only vertically. And when the target is not easier than the source, $\ell = J_2(\pi_1^*)$ is the return of the uncompressed counterpart of π_β, which yields guarantee (ii) below.

3.4 Certifying near-optimal transfer

Near-optimality on the target, our success criterion throughout, is the statement $J_2(\pi_\beta) \ge (1-\varepsilon)J_2^*$ for a fixed tolerance $\varepsilon \in (0,1)$. As §3.3 showed, turning the transfer-error bound G_{IB} into this statement requires only a lower bound on the target optimum, which we chose to be ℓ. The next step c§2q3wombines the two.

Theorem 3.3. Sufficient condition for near-optimal transfer Let $\varepsilon \in (0,1)$ and $\ell = J_1^* - B_{\text{eval}} > 0$. If $G_{\text{IB}} \le \varepsilon\ell$, then (i) $J_2(\pi_\beta) \ge (1-\varepsilon)J_2^*$ and (ii) $J_2(\pi_\beta) \ge (1-\varepsilon)\min\{J_1^*, J_2(\pi_1^*)\}$.

Proof sketch. By Theorem 3.2, $J_2(\pi_\beta) \ge J_2^* - G_{\text{IB}} \ge J_2^* - \epsilon\,\ell$. The first claim follows from $\ell \le J_2^*$ (Lemma 3.2), the second from $J_2^* \ge \ell$ applied to the first term. See Appendix B.4 for the full proof. □

(i) is near-optimality on the target; (ii) says the compressed policy keeps a fraction $1-\epsilon$ of what its uncompressed counterpart earns there. By (ii), when the target is not easier than the source, π_β keeps at least a fraction $1-\varepsilon$ of what π_1^* earns there. The left-hand side of the condition needs π_2^* (through T, ΔA_2 and O), the occupancy $\rho_2^{\pi_\beta}$ and the source-side terms d_1 and $\bar{L}$, and the right-hand side needs J_1^* and $J_2(\pi_1^*)$.

In Figure 1 (right) the condition is the dashed boundary, $G_{\text{IB}} = \varepsilon(J_1^* - B_{\text{eval}})$, which falls as the drift B_{eval} leaves less return to certify against. Each transfer is tested against its own J_1^*; the figure draws the boundary at the median J_1^* over the transfers shown. The condition compares bounds, $G_{\text{IB}} \ge G_{\text{exact}}$ and $\ell \le J_2^*$, rather than the quantities themselves, so a near-optimal transfer whose bounds are loose can lie above the boundary: the condition is sufficient, not necessary. Section 4 reports its recall and its false certifications, which the theorem guarantees to be zero, and compares G_{IB} with a tighter bound that ignores compression.

4 Tabular evaluation

Setup We first validate the sufficient condition on 10×10 grid-worlds, where the transfer error $G_{\text{exact}} = J_2^* - J_2(\pi_\beta)$ is computed exactly and the bottleneck is solved exactly by Blahut–Arimoto

[1] With $\ell = J_2^*$, Theorem 3.3 certifies 93.6% of the near-optimal tabular transfers with zero false certifications. We keep the form below for its reading.

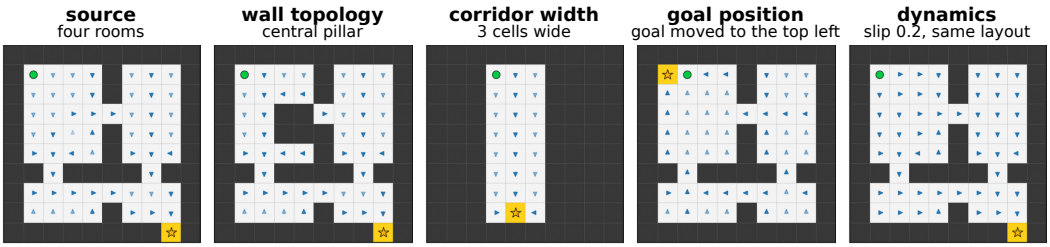

Figure 2: Tabular source `four_rooms` and one target per shift family (central pillar, 3-cell corridor, moved goal, slip 0.2; star: goal, dot: start, arrows: optimal policy); all environments in Appendix F.

Setting	Bound	Recall	False cert.	Med. looseness
Tabular	G_{IB}	88.6%	0	1.89×
Tabular	G_{base}	97.5%	0	1.58×
Pixel	G_{IB}	68.4%	0	1.74×

Table 1: **The decomposition trades recall for interpretable terms.** Tabular: recall over all 1,830 successes among 13,200 configurations; looseness on zero-floor rows with transfer error $> 10^{-4}$ (median $G_{\mathrm{IB}}/G_{\mathrm{base}}$ is 1.12). Pixel (§6.1): 54 of 79 successes certified among 210 configurations. Full distributions: Figure 9.

(Tishby et al., 2000). Targets vary wall topology, corridor width, goal position, or slip dynamics (Figure 2; all environments in Appendix F); 8 sources against 33 targets, 5 compression levels, 2 decoder floors and 5 seeds give 13,200 configurations (Appendix F.1).

For each configuration we compare two upper bounds of the same transfer error G_{exact}: G_{IB}, built from the budget b in §3.1–3.4, and $G_{\mathrm{base}} := \frac{1}{1-\gamma}\,\mathbb{E}_{d_2^{\pi_\beta}}\left[\mathrm{TV}(\pi_2^*, \pi_\beta)\,(s)\,\Delta A_2(s)\right]$ (the direct bound A' of Appendix A.3), which is tighter but gives no attribution. It follows from the same performance-difference identity as Lemma 3.1, bounding the per-state loss by disagreement times advantage range rather than through the distortion. We compare against it rather than the lower route a→b of Figure 1 (left) because it differs from G_{IB} only in the per-state factor, $\mathrm{TV}(\pi_2^*, \pi_\beta)(s)$ in place of $1-e^{-b(s)}$, so substituting it in the sufficient condition isolates the price of the IB decomposition. We report *recall*, how many successes each bound certifies (a success reaches 90% of the optimal target return), the number of false certifications, and *looseness*, the bound divided by the true transfer error where that error exceeds 10^{-4}.

4.1 What the decomposition costs

Across all 13,200 configurations, neither bound ever certifies a failed transfer (Table 1). Missed successes come from the nature of the certificate: when it holds the transfer is guaranteed to be near-optimal, and when it does not the question is merely left open.

What the decomposition costs is coverage, and the cost is one-directional. Among 1830 successful transfers, G_{base} certifies every configuration G_{IB} does and 164 more. So the decomposition does not trade one region of the space for another, it shrinks the certified region. Its price is 8.9 points of recall, 88.6% against 97.5%. The 164 configurations certified by G_{base} alone are near-misses, not blind spots: they are the near-optimal transfers with the largest true regret (median 36% of the allowed budget, against 2% for certified ones), where G_{base} already sits close to the threshold (blue rings just above the boundary in Figure 1, right); the certificate still fires on 21 of the 22 targets with any near-optimal transfer. That price is paid in coverage, not in looseness. At the median G_{IB} overshoots the true transfer error by 1.89× and G_{base} by 1.58× (Figure 9); the decomposition itself adds a median factor of 1.12, the price of a budget that can be interpreted term by term (§5.2) rather than a single number.

5 THEORY: POLICIES WITH A LEARNED REPRESENTATION

We now move from the tabular setting to the more realistic one of high-dimensional image observations, where a variational information bottleneck (VIB) has to learn a representation (Alemi et al., 2019). Two things follow. The terms of the budget $b(s)$ become estimates, since the encoder is now a distribution the network samples from. And the budget gains a representation shift term, because the same state can be represented differently on the target. A mismatch between what the agent observes and the information needed to solve the task is an additional cause of failure in this setting. Since the budget separates the two, it can identify this cause rather than detecting a general failure.

The encoder $p(z \mid o)$ of the neural agent reads an observation o and produces a Gaussian distribution on a continuous code space $\mathcal{Z}$. We employ a shared-sample estimator that for each state s draws *one* set of $K \geq 1$ codes $z_1(s), \ldots, z_K(s)$ i.i.d. from $p(\cdot \mid o_2(s))$, the encoder applied to the target observation of s. Every quantity below is computed from this same set. For each sample, $\tilde{q}_k(\cdot \mid s)$ is the decoder row $q(\cdot \mid z_k(s))$ clipped at $\epsilon_{\text{clip}} > 0$ and renormalised over $\mathcal{A}$; thus $\tilde{q}_k(a \mid s) > 0$ for all a and $\sum_a \tilde{q}_k(a \mid s) = 1$. The policy deployed on the target is the fixed stochastic policy $\hat{\pi_\beta}(\cdot \mid s) = K^{-1} \sum_k q(\cdot \mid z_k)$. The other quantities $\hat{d}_i(s)$, $\hat{L}(s)$, $\hat{b}(s)$ and $\hat{G}_{\text{IB}}$ are similarly obtained by mixing the decoder $\tilde{q}_k(\cdot \mid s)$. Since the target is tabular, $\rho_2^{\hat{\pi_\beta}}$, ΔA_2, $T(s)$ and $J_2(\hat{\pi_\beta})$ are exact and $\ell = \min\{J_1^*, J_2(\pi_1^*)\}$ is as in Theorem 3.2.

5.1 CERTIFYING A SAMPLED POLICY

The neural encoder raises a natural question: if now the budget is Monte-Carlo estimated, couldn't estimation noise make the certificate wrong? The answer is no, by construction. We approximate the policy and its distortion with the same latent samples, so that the Theorems 3.1 and 3.3 can certify a specific empirical policy exactly rather than with a Monte Carlo estimate of a population policy.

Proposition 5.1. deterministic validity of the shared-sample certificate For every state s and *every* realisation of the samples $z_1(s), \ldots, z_K(s)$: (i) $\hat{d}_2(s) \leq \hat{b}(s)$; (ii) $\mathrm{KL}(\pi_2^*(\cdot \mid s) \,\|\, \hat{\pi_\beta}(\cdot \mid s)) \leq \hat{d}_2(s)$; (iii) $J_2^* - J_2(\hat{\pi_\beta}) \leq \hat{G}_{\text{IB}}$; (iv) if $\ell > 0$ and $\hat{G}_{\text{IB}} \leq \epsilon\,\ell$, then $J_2(\hat{\pi_\beta}) \geq (1-\epsilon) J_2^*$ and $J_2(\hat{\pi_\beta}) \geq (1-\epsilon)\,\ell$. In particular, sampling affects which policy is certified and the value of $\hat{G}_{\text{IB}}$, but it cannot make a certificate that fires unsound; no confidence level or estimation-error term is needed.

Proof sketch. $\hat{p}_s(z) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\{z = z_k(s)\}$ is a mixing distribution over codes and $\tilde{q}_k$ are strictly positive decoder rows, so Theorem 3.1, Lemma 3.1, Theorem 3.2 and Theorem 3.3 apply to them verbatim; under $\hat{p}_s$ the distortions, peakiness and mixture policy are exactly $\hat{d}_i(s)$, $\hat{L}(s)$ and $\hat{\pi_\beta}$, which gives (i)–(iv). Since these results hold for *any* mixing distribution, they hold for every realisation of the samples. See Appendix B.6 for the full proof. □

5.2 READING THE BUDGET: COMPRESSION, REPRESENTATION SHIFT, OR TASK DISAGREEMENT

Three things can cost transfer return at a state: compressing, the encoder seeing a different frame, and the task requiring a different action. The budget of Theorem 3.1 separates them. For a pixel agent every quantity in $b(s)$ is estimated from the shared samples of the neural estimator, drawn through the target frame $o_2(s)$ as the theorem requires. Write $\hat{d}_1^{(2)}(s)$ for the source distortion so estimated, $\hat{d}_1^{(1)}(s)$ for the same quantity through the source frame $o_1(s)$ (which is the training loss up to the constant $H(\pi_1^*)$). Denote also with $\hat{\Delta}_{\text{rep}}(s) := \hat{d}_1^{(2)}(s) - \hat{d}_1^{(1)}(s)$ the *representation shift*. Then

$$\hat{b}(s) \;=\; \underbrace{\hat{d}_1^{(1)}(s)}_{\text{compression}} \;+\; \underbrace{\hat{\Delta}_{\text{rep}}(s)}_{\text{representation shift}} \;+\; \underbrace{2\,T(s)\,\hat{L}_2(s)}_{\substack{\text{task disagreement} \\ \times\ \text{decoder sensitivity}}} \;+\; \underbrace{h(T(s)) + T(s)\log(m-1)}_{\substack{\text{entropy correction:} \\ \text{which action instead}}}.$$

Read as diagnoses, the first distortion term $\hat{d}_1^{(1)}(s)$ is the price of compression itself, paid even when nothing differs. When it is large means that $\hat{\pi_\beta}$ imitated the source poorly, so transfer fails for a

reason unrelated to the target. The representation shift $\hat{\Delta}_{\mathrm{rep}}$ holds the reference policy π_1^* and the decoder fixed and changes only what the encoder reads, so it isolates the cost of the decoder seeing a different observation of the same state. It vanishes whenever the two observations coincide, for instance when only the dynamics is changed or in fully observable environments, and it is large when only the appearance changes and the encoder, not the task, is at fault, as under a colour swap that leaves the MDP intact. The last three terms vanish when the two tasks agree at s. When they are large means that the target needs actions the source-trained decoder excludes, which no change of representation can repair, such as when goals positions are moved. Each term thus points to one cause of a failed transfer. $\hat{\Delta}_{\mathrm{rep}}$ is signed, since a target frame can happen to be read better than a source frame. Compression is meant to help here: discarding task-irrelevant detail can make the code insensitive to appearance, so representation shift can fall as compression increases, at the price of higher source distortion.

6 PIXEL EVALUATION

Setup. We evaluate a VIB (Alemi et al., 2019) on image-based 8×8 grid-worlds (details in Appendix F). All pairs share a MiniGrid source room (Chevalier-Boisvert et al., 2023), with seven targets varying slip, walls, goal, or colours (Figure 4, Appendix F); slip preserves the rendered frames and colour swap preserves the MDP. We sweep 6 compression levels and 5 seeds (210 configurations), fixing each trained agent across targets. The deployed policy is a sampled mixture certified by Proposition 5.1.

6.1 CERTIFYING THE SAMPLED POLICY

Does the sufficient condition of Theorem 3.3 hold for the estimator of §5.1 once $\hat{\pi_\beta}$ is a neural agent? Across all 210 pixel configurations, 79 transfers succeed by the same 90%-of-optimal criterion as the tabular study, and the condition certifies 54 of them (68.4% recall) with zero false certifications (Table 1). Its median looseness, $1.74\times$, is close to the tabular $1.89\times$ despite sampling; $\hat{G_{\mathrm{IB}}}$ never underestimates the transfer error (Figure 10), and the per-state bound tracks the per-state loss (Spearman 0.81, Figure 8).

6.2 ATTRIBUTION RESULTS

Figure 3(a) shows the budget of §5.2 term by term for each target at $\beta_{\mathrm{vib}} = 0.1$ (values in Table 6, Appendix F.2): the controlled interventions separate causes in four patterns. Colour swap preserves the MDP – $B_{\mathrm{eval}} = 0$, so the source-optimal policy is still target-optimal – yet it is never certified, because representation shift alone accounts for the gap. Slip preserves the frames and, for these targets, the optimal actions: both shift contributions vanish even though reference-policy drift changes, and the transfer is certified on source distortion alone. Goal moves exhaust the drift margin regardless of the encoder ($B_{\mathrm{eval}} = 0.58$ and 0.88): task disagreement dominates the budget and no transfer can be certified there, no matter how compression is tuned. Wall changes fall between the two: they move both the representation and the task terms.

Compression changes the trade-off. Figure 3(b) shows how the diagnosis moves with the compression knob. On the wall-cell target the two shift terms respond in opposite directions: from $\beta_{\mathrm{vib}} = 0.001$ to 0.5 the representation shift falls from 3.6 to 0.4 bits, because a coarser code no longer resolves the frame differences that a wall adds, while the source distortion rises from 0.01 to 0.9 bits, the ordinary price of compressing, and task disagreement falls from 2.8 to 0.5 bits. At $\beta_{\mathrm{vib}} = 0.5$ the transfer is certified for four of five seeds. At $\beta_{\mathrm{vib}} = 2$ the budget is smaller still, but the code carries no information about the state ($I(S;Z) = 0$), the policy keeps only 37% of the optimal source return, and the transfer is no longer near-optimal. The same trade-off appears on colour swap, where the representation shift falls from a peak of 21 bits at $\beta_{\mathrm{vib}} = 0.02$ to zero at $\beta_{\mathrm{vib}} = 2$, but only once the source return has collapsed, so no level is certified. The diagnoses of the two cases differ, and G_{base}, which reports one number per target, could not tell them apart.

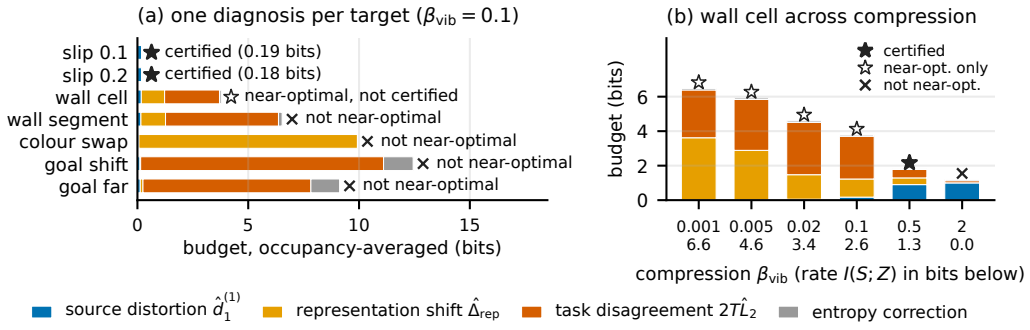

Figure 3: Budget attribution on the pixel benchmark. *(a)* The budget terms of §5.2 for every target at $\beta_{\text{vib}} = 0.1$, each bar ending in its verdict. *(b)* Wall cell across compression. Bars stack per-term medians over five seeds, occupancy-averaged, in bits; verdicts are majorities of seeds.

7 Related work

Bisimulation is a metric that bounds value differences between states (Ferns et al., 2004). Our closest relative, cross-MDP bisimulation, supports policy transfer with quality guarantees (Castro & Precup, 2010). These bounds hold for a pair of MDPs and describe what any policy could achieve given state similarity, not a policy compressed on the source. We bound the return of one compressed policy, at its own compression level, and attribute its transfer error to compression or to the task change.

The **information bottleneck** (Tishby et al., 2000; Alemi et al., 2019) compresses a representation, keeping task-relevant information. in RL it is a regularizer that improves generalization empirically (Goyal et al., 2019; Igl et al., 2019; Goyal et al., 2020; Lu et al., 2020) , with IBAC closest to our work (Igl et al., 2019). None of it turns the compression level into a sufficient condition for near-optimal transfer. Where it finds that compression helps, we show per target and in bits which part of the transfer error compression removes (representation shift) and which it cannot (task disagreement; §6.2).

Benchmarks for zero-shot generalisation and transfer in RL test a policy on held-out variations, such as procedurally generated levels (Cobbe et al., 2020) or perturbed physical parameters (Packer et al., 2018), chosen mostly ad hoc, as is how far test tasks lie from training tasks. Müller-Brockhausen et al. (2021) call for a universal measure of task similarity. We measure per target how much a task change costs in return and which part of the transfer error the policy causes.

8 Discussion

Our main limitation is that the theory applies to finite MDPs and needs both task models and optimal policies, making it an audit of transfer rather than a prediction. Scaling to Atari (Bellemare et al., 2013), continuous control (Duan et al., 2016) and procedural benchmarks (Cobbe et al., 2020), is future work.

Benchmarks for transfer and generalisation are designed ad hoc (Kirk et al., 2023), obscuring which changes make a method fail. Representation shift, task disagreement and drift give these changes interpretable axes: task families along which they grow gradually would raise difficulty explicitly, e.g. to sequence curricula (Klink et al., 2022) or build directed transfer-difficulty graphs (Zamir et al., 2018). Oracle information matters less here: the designer can solve every task, and the terms give a ground truth for evaluating methods.

Oracle-free prediction of transfer, without solving the target, is the most ambitious goal. Our construction isolates the task-level quantities it must supply (Appendix P), bounding them from a task-family description, or estimating them from interaction data, would make the condition computable before deployment and beyond finite MDPs. Behavioural divergences already rank transfers well without a target solution (Appendix M), but, being reward-blind, cannot replace a guarantee.

REPRODUCIBILITY STATEMENT

Every result is stated with its assumptions in the main text and proved in Appendix B. The tabular and pixel benchmarks, the sweeps and the training settings are specified in Appendix F, and per-target results are listed in Appendix K. The code and the result tables from which every figure and table in the paper is regenerated are provided as supplementary material.

AI USE STATEMENT

We used generative AI tools to suggest proof directions. The authors developed the proofs manually and verified each argument. We also used these tools to generate code, whose correctness we checked through tests and manual review, and to restructure and edit the manuscript, clarify terminology, design diagrams, identify relevant literature, and check the compiled document. The authors take responsibility for the final text, proofs, code and results.

REFERENCES

David Abel, Dilip Arumugam, Kavosh Asadi, Yuu Jinnai, Michael L. Littman, and Lawson L.S. Wong. State Abstraction as Compression in Apprenticeship Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3134–3142, July 2019. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v33i01.33013134. URL https://ojs.aaai.org/index.php/AAAI/article/view/4179.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep Variational Information Bottleneck, October 2019. URL http://arxiv.org/abs/1612.00410. arXiv:1612.00410 [cs, math].

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013. ISSN 1076-9757. doi: 10.1613/jair.3912. URL https://jair.org/index.php/jair/article/view/10819.

Pablo Castro and Doina Precup. Using Bisimulation for Policy Transfer in MDPs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1):1065–1070, July 2010. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v24i1.7751. URL https://ojs.aaai.org/index.php/AAAI/article/view/7751.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and J. Terry. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, December 2023. doi: 10.52202/075280-3209. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e8916198466e8ef218a2185a491b49fa-Abstract-Datasets_and_Benchmarks.html.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying Generalization in Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 1282–1289. PMLR, May 2019. URL https://proceedings.mlr.press/v97/cobbe19a.html.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging Procedural Generation to Benchmark Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 2048–2056. PMLR, November 2020. URL https://proceedings.mlr.press/v119/cobbe20a.html.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking Deep Reinforcement Learning for Continuous Control. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1329–1338. PMLR, June 2016. URL https://proceedings.mlr.press/v48/duan16.html.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, September 2021. ISSN 1573-0565. doi: 10.1007/s10994-021-05961-4. URL https://doi.org/10.1007/s10994-021-05961-4.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for Finite Markov Decision Processes. In *UAI*, volume 4, pp. 162–169, 2004. URL https://cdn.aaai.org/AAAI/2004/AAAI04-124.pdf.

Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. InfoBot: Transfer and Exploration via the Information Bottleneck, April 2019. URL http://arxiv.org/abs/1901.10902. arXiv:1901.10902 [cs, stat].

Anirudh Goyal, Yoshua Bengio, Matthew Botvinick, and Sergey Levine. THE VARIATIONAL BANDWIDTH BOTTLENECK: STOCHASTIC EVALUATION ON AN INFORMATION BUDGET. 2020.

Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. Generalization in Reinforcement Learning with Selective Noise Injection and Information Bottleneck, October 2019. URL http://arxiv.org/abs/1910.12911. arXiv:1910.12911 [cs].

Sham Kakade. Optimizing Average Reward Using Discounted Rewards. In David Helmbold and Bob Williamson (eds.), *Computational Learning Theory*, pp. 605–615, Berlin, Heidelberg, 2001. Springer. ISBN 978-3-540-44581-4. doi: 10.1007/3-540-44581-1_40.

Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pp. 267–274, San Francisco, CA, USA, July 2002. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-873-3.

Michael Kearns and Satinder Singh. Near-Optimal Reinforcement Learning in Polynomial Time. *Machine Learning*, 49(2):209–232, November 2002. ISSN 1573-0565. doi: 10.1023/A:1017984413808. URL https://doi.org/10.1023/A:1017984413808.

Satinder Kearns, Michael and Singh. Near-Optimal Reinforcement Learning in Polynomial Time. 49:209–232, 2002.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, 76:201–264, January 2023. ISSN 1076-9757. doi: 10.1613/jair.1.14174. URL https://www.jair.org/index.php/jair/article/view/14174. arXiv:2111.09794 [cs].

Pascal Klink, Haoyi Yang, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. Curriculum Reinforcement Learning via Constrained Optimal Transport. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 11341–11358. PMLR, June 2022. URL https://proceedings.mlr.press/v162/klink22a.html.

Lucas Lehnert, Michael L. Littman, and Michael J. Frank. Reward-predictive representations generalize across tasks in reinforcement learning. *PLOS Computational Biology*, 16(10):e1008317, October 2020. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1008317. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008317.

Xingyu Lu, Kimin Lee, Pieter Abbeel, and Stas Tiomkin. Dynamics Generalization via Information Bottleneck in Deep Reinforcement Learning, August 2020. URL http://arxiv.org/abs/2008.00614. arXiv:2008.00614 [cs, stat].

Matthias Müller-Brockhausen, Mike Preuss, and Aske Plaat. Procedural Content Generation: Better Benchmarks for Transfer Reinforcement Learning. In *2021 IEEE Conference on Games (CoG)*, pp. 01–08, August 2021. doi: 10.1109/CoG52621.2021.9619000. URL http://arxiv.org/abs/2105.14780. arXiv:2105.14780 [cs.LG].

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, V. Koltun, and D. Song. Assessing Generalization in Deep Reinforcement Learning. *ArXiv*, September 2018. URL https://www.semanticscholar.org/paper/caea502325b6a82b1b437c62585992609b5aa542.

Stephane Ross and Drew Bagnell. Efficient Reductions for Imitation Learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, March 2010. URL https://proceedings.mlr.press/v9/ross10a.html.

Matthew E. Taylor, Shimon Whiteson, and Peter Stone. Transfer via inter-task mappings in policy search reinforcement learning. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pp. 1–8, New York, NY, USA, May 2007. Association for Computing Machinery. ISBN 978-81-904262-7-5. doi: 10.1145/1329125.1329170. URL https://dl.acm.org/doi/10.1145/1329125.1329170.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, April 2000. URL http://arxiv.org/abs/physics/0004057. arXiv:physics/0004057.

Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning, April 2018. URL http://arxiv.org/abs/1804.08328. arXiv:1804.08328 [cs].

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning Invariant Representations for Reinforcement Learning without Reconstruction, April 2021. URL http://arxiv.org/abs/2006.10742. arXiv:2006.10742 [cs].

A Preliminaries

A.1 Occupancy and action gaps

Occupancy $\rho_i^\pi(s) := (1-\gamma)\sum_{t\geq 0}\gamma^t \Pr_{M_i,\pi,\mu}(s_t = s)$ is the normalised discounted fraction of time π spends at s under M_i: it turns a per-state quantity into a return-relevant average, weighting states by how often the policy actually visits them rather than by their worst case. The distortion quantities in §3.1 are defined statewise and can subsequently be averaged under any such occupancy. The target action gap is $\Delta_2(s,a) = V_2^*(s) - Q_2^*(s,a) \geq 0$; its maximum $\Delta A_2(s) = \max_a \Delta_2(s,a)$ measures the cost of the worst action at s and is used from §3.2 on. Occupancy and action gaps together determine how much an imitation error at a state matters for return.

A.2 Two return identities

The *performance-difference identity* (Kakade & Langford, 2002) states that for any two policies π, π' in the same MDP, $J(\pi') - J(\pi) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim\rho^{\pi'}}[\sum_a \pi'(a \mid s)A^\pi(s,a)]$. This identity expresses the return difference as the occupancy-weighted expected advantage. For an optimal reference policy, all advantages are non-positive.

The *fixed-policy drift identity* (Kearns, 2002) states that for one policy π evaluated in two MDPs sharing $(\mathcal{S}, \mathcal{A}, \gamma, \mu)$, $J_1(\pi) - J_2(\pi) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim\rho_1^\pi}[\delta(s)]$ with $\delta(s) = r_1^\pi(s) - r_2^\pi(s) + \gamma\langle (P_1^\pi - P_2^\pi)(\cdot \mid s), V_2^\pi\rangle$, where $P_i^\pi(\cdot \mid s) = \sum_a \pi(a \mid s)P_i(\cdot \mid s,a)$ and $r_i^\pi(s) = \sum_a \pi(a \mid s)r_i(s,a)$. Like the first, it is exact, and $\delta(s)$ may take either sign.

A.3 The direct bound

Specialised to the target-optimal reference, the performance-difference identity gives the transfer error exactly, with action gap $\Delta_2(s,a) := -A_2^*(s,a)$: $G_{\text{exact}} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_2^{\pi_\beta}}[\sum_a \pi_\beta(a \mid s)\Delta_2(s,a)]$. Bounding the inner sum by Hölder's inequality (ℓ_1 against ℓ_∞) gives, at every state, $\sum_a \pi_\beta(a \mid s)\Delta_2(s,a) \leq \mathrm{TV}(\pi_2^*, \pi_\beta)(s)\Delta A_2(s)$, and hence

$$G_{\text{exact}} \leq A' := \tfrac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_2^{\pi_\beta}}[\mathrm{TV}(\pi_2^*, \pi_\beta)(s)\Delta A_2(s)].$$

Unlike G_{IB}, A' never routes through the encoder–decoder decomposition: it compares π_β to π_2^* directly. §4 substitutes it for G_{IB} in the sufficient condition to measure what the decomposition costs.

A.4 Representation shift

Definition A.1. Representation shift Let $o_i(s)$ be the observation task i renders at state s, so the encoder consumes $p(z \mid o_i(s))$ rather than s directly ($o_1 = o_2 = s$ when states are fully observed, as in the tabular setting). Evaluating the source-reference distortion on task i's observation gives $d_1^{(i)}(s) := \mathbb{E}_{z\sim p(\cdot|o_i(s))}[D_{\mathrm{KL}}\pi_1^*(\cdot \mid s)q(\cdot \mid z)]$, with $d_1(s) = d_1^{(1)}(s)$. *Representation shift* is the change this induces at the target observation, holding the reference policy π_1^* and the decoder fixed:

$$\Delta_{\mathrm{rep}}(s) := d_1^{(2)}(s) - d_1^{(1)}(s).$$

It isolates the part of source distortion caused by the encoder seeing a different observation, as distinct from $T(s)$, which compares $\pi_1^*(\cdot \mid s)$ and $\pi_2^*(\cdot \mid s)$ directly and does not depend on o. Applying Theorem 3.1 with the target encoder distribution replaces the source term by $d_1^{(2)}(s) = d_1(s) + \Delta_{\mathrm{rep}}(s)$ and evaluates $\bar{L}(s)$ on $o_2(s)$ as well. This splits the budget into source error, representation shift, and task disagreement. The signed quantity $\Delta_{\mathrm{rep}}(s)$ can be negative when the new observation improves imitation of the source reference. In the tabular setting $\Delta_{\mathrm{rep}} \equiv 0$; Section 6.2 measures it for a pixel encoder, where it can dominate the budget even at zero task disagreement.

B Proofs

This appendix collects the proofs of the results in the main text: Theorem 3.1 (§B.1), Theorem 3.2 (§B.2), Theorem 3.3 (§B.4), Eq. 4 (§B.5) and Proposition 5.1 (§B.6). Results stated in the other appendices are proved directly below their statements.

B.1 Theorem 3.1 (pointwise transfer radius)

Theorem B.1. Pointwise source-to-target distortion bound or every state s, with $m \geq 2$ and positive decoder probabilities,

$$d_2(s) \leq b(s) := \underbrace{d_1(s)}_{\text{source distortion}} + \underbrace{2\,T(s)\,\bar{L}(s)}_{\substack{\text{task disagreement} \\ \times \text{ decoder sensitivity}}} + \underbrace{h(T(s)) + T(s)\log(m-1)}_{\substack{\text{entropy correction:} \\ \text{which action instead}}}. \tag{6}$$

The proof uses one standard inequality:

Fact 1 (entropy continuity; Fannes–Audenaert, classical case; Audenaert (2007)). For probability distributions u, v on $m \geq 2$ atoms with $T = \frac{1}{2}\|u - v\|_1$,

$$|H(u) - H(v)| \;\leq\; h(T) + T\log(m-1).$$

Proof. Fix s and z, and abbreviate $\pi_i = \pi_i^*(\cdot \mid s)$ and $q = q(\cdot \mid z)$. Since $q(a) > 0$ for all a, both divergences are finite and

$$d_i(s,z) := \sum_a \pi_i(a)\log\frac{\pi_i(a)}{q(a)} = -H(\pi_i) - \sum_a \pi_i(a)\log q(a).$$

Subtracting the two expressions,

$$d_2(s,z) - d_1(s,z) = \big[H(\pi_1) - H(\pi_2)\big] + \sum_a \big(\pi_1(a) - \pi_2(a)\big)\log q(a). \tag{7}$$

Cross entropy term. By Hölder's inequality (ℓ_1–ℓ_∞),

$$\Big|\sum_a \big(\pi_1(a) - \pi_2(a)\big)\log q(a)\Big| \leq \|\pi_1 - \pi_2\|_1 \cdot \max_a |\log q(a)| = 2T(s)\cdot L(z),$$

because $\log q(a) \le 0$ for every a, so $\max_a |\log q(a)| = \max_a(-\log q(a)) = L(z)$.

Entropy term. By Fact 1 applied to π_1, π_2 on the m atoms of $\mathcal{A}$, whose total-variation distance is $T(s)$,

$$H(\pi_1) - H(\pi_2) \le h\big(T(s)\big) + T(s)\log(m-1).$$

Inserting both bounds in equation 7 gives

$$d_2(s,z) \;\le\; d_1(s,z) \;+\; 2\,T(s)\,L(z) \;+\; h\big(T(s)\big) + T(s)\log(m-1). \tag{8}$$

Averaging. Inequality equation 8 holds for every z. Taking $\mathbb{E}_{z\sim p(\cdot|s)}$ of both sides, and using $\mathbb{E}_z[d_i(s,z)] = d_i(s)$, $\mathbb{E}_z[L(z)] = \bar{L}(s)$, and that $T(s)$ does not depend on z, yields equation 6. All expectations are of nonnegative quantities and $\bar{L}(s) < \infty$, so they are well defined and finite. □

B.2 Lemma 3.1

Lemma B.1. Distortion-to-return bridge ssume that the target-optimal policy is supported on the optimal actions, $\pi_2^*\big(\mathcal{O}(s)\,\big|\,s\big) = 1$ for every s, then for the deployed policy $\pi_\beta(a \mid s) = \mathbb{E}_{z\sim p(\cdot|s)}\big[q(a \mid z)\big]$ with any mixing distribution $p(\cdot \mid s)$,

$$J_2^* - J_2(\pi_\beta) \le \tfrac{1}{1-\gamma}\,\mathbb{E}_{s\sim\rho_2^{\pi_\beta}}\big[\Delta A_2(s)\big(1 - e^{-d_2(s)}\big)\big]. \tag{9}$$

The proof of Lemma 3.1 uses three standard facts.

Fact 2 (convexity of the divergence in its second argument). For a distribution u on $\mathcal{A}$ and a family $\{q_z\}$ of distributions on $\mathcal{A}$ with $q_z(a) > 0$, mixed under any distribution p on z: $\mathrm{KL}(u \,\|\, \mathbb{E}_z q_z) \le \mathbb{E}_z \mathrm{KL}(u \,\|\, q_z)$. (Cover and Thomas 2006, Thm. 2.7.2.)

Fact 3 (mass of a sure event under another distribution). For distributions u, v on $\mathcal{A}$ with $v > 0$ and an event $\mathcal{O} \subseteq \mathcal{A}$ with $u(\mathcal{O}) = 1$: $v(\mathcal{O}) \ge e^{-\mathrm{KL}(u\,\|\,v)}$, i.e. $v(\mathcal{O}^c) \le 1 - e^{-\mathrm{KL}(u\,\|\,v)}$. (Log-sum inequality, Cover and Thomas 2006, Thm. 2.7.1, applied to the two vectors restricted to $\mathcal{O}$.)

Fact 4 (performance-difference lemma; Kakade and Langford 2002). For any two policies π, π' in the same MDP,

$$J(\pi) - J(\pi') = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^\pi}\Big[\sum_a \pi(a \mid s)\, A^{\pi'}(s,a)\Big].$$

Proof. Fix a state s and abbreviate $u = \pi_2^*(\cdot \mid s)$, $v = \pi_\beta(\cdot \mid s)$.

Step 1 (mixture). By $\pi_\beta(a \mid s) = \mathbb{E}_{z\sim p(\cdot|s)}\big[q(a \mid z)\big]$ v is the mixture of the rows $q(\cdot \mid z)$ under $p(\cdot \mid s)$, so by Fact 2

$$D_{\mathrm{KL}}uv \;\le\; \mathbb{E}_{z\sim p(\cdot|s)} D_{\mathrm{KL}}uq(\cdot \mid z) \;=\; d_2(s).$$

Step 2 (optimal mass). By $\pi_2^*\big(\mathcal{O}(s)\,\big|\,s\big) = 1$, $u(\mathcal{O}(s)) = 1$, so Fact 3 and Step 1 give

$$\pi_\beta\big(\mathcal{O}(s)^c\,\big|\,s\big) \;\le\; 1 - e^{-D_{\mathrm{KL}}uv} \;\le\; 1 - e^{-d_2(s)},$$

the last step because $x \mapsto 1 - e^{-x}$ is increasing.

Step 3 (per-state regret). The gap $\Delta_2(s,a)$ is zero for $a \in \mathcal{O}(s)$ and at most $\Delta A_2(s)$ otherwise, hence

$$\sum_a \pi_\beta(a \mid s)\,\Delta_2(s,a) \;=\; \sum_{a\notin\mathcal{O}(s)} \pi_\beta(a \mid s)\,\Delta_2(s,a) \;\le\; \Delta A_2(s)\,\pi_\beta\big(\mathcal{O}(s)^c\,\big|\,s\big) \;\le\; \Delta A_2(s)\,\big(1 - e^{-d_2(s)}\big). \tag{10}$$

Step 4 (assembly). Fact 4 with $\pi = \pi_\beta$ and $\pi' = \pi_2^*$, using $A_2^*(s,a) = -\Delta_2(s,a)$, gives the identity

$$J_2^* - J_2(\pi_\beta) \;=\; \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim\rho_2^{\pi_\beta}}\Big[\sum_a \pi_\beta(a \mid s)\,\Delta_2(s,a)\Big].$$

Substituting equation 10 inside the expectation, which is legitimate because equation 10 holds at every state and $\rho_2^{\pi_\beta}$ is a probability distribution, yields equation 9. □

B.3 Lemma 3.2 (Lower bound on the target optimum)

Lemma B.2. Lower bound on the target optimum Given the scale ℓ defined in 4, one has

$$\ell = \min\{J_1^*,\, J_2(\pi_1^*)\},\ \text{and}\ \ \ell \le J_2^* \tag{11}$$

The proof of Lemma 3.2 uses one standard fact.

Fact 5 (the optimal return dominates every policy). In a finite discounted MDP, $V^*(s) \ge V^\pi(s)$ for every policy π and every state s; hence $J^* = \mathbb{E}_\mu[V^*] \ge \mathbb{E}_\mu[V^\pi] = J(\pi)$. (Puterman 1994, Thm. 6.2.2(b), with V^* the supremum over policies as defined in his §6.2)

Proof. For any real x, y, $x - [x-y]_+ = \min\{x, y\}$: if $x \ge y$ the left side is $x - (x-y) = y$, otherwise it is $x - 0 = x$. With $x = J_1^*$ and $y = J_2(\pi_1^*)$ this gives $\ell = \min\{J_1^*, J_2(\pi_1^*)\}$. Since π_1^* is a policy of M_2, Fact 5 gives $J_2(\pi_1^*) \le J_2^*$, and therefore $\ell \le J_2(\pi_1^*) \le J_2^*$. □

B.4 Theorem 3.3 (sufficient condition for near-optimal transfer)

Theorem B.2. Sufficient condition for near-optimal transfer et $\varepsilon \in (0,1)$ and $\ell = J_1^* - B_{\text{eval}} > 0$. If $G_{\text{IB}} \le \varepsilon\ell$, then (i) $J_2(\pi_\beta) \ge (1-\varepsilon)J_2^*$ and (ii) $J_2(\pi_\beta) \ge (1-\varepsilon)\min\{J_1^*, J_2(\pi_1^*)\}$.

Proof. Theorem 2 states $J_2^* - J_2(\pi_\beta) \le G_{\text{IB}}$, i.e. $J_2(\pi_\beta) \ge J_2^* - G_{\text{IB}}$. With the hypothesis $G_{\text{IB}} \le \varepsilon\ell$,

$$J_2(\pi_\beta)\ \ge\ J_2^* - G_{\text{IB}}\ \ge\ J_2^* - \epsilon\,\ell. \tag{12}$$

For (i): by Lemma 3.2, $\ell \le J_2^*$, and $\epsilon > 0$, so $\epsilon\,\ell \le \epsilon\, J_2^*$ and equation 12 gives $J_2(\pi_\beta) \ge J_2^* - \epsilon J_2^* = (1-\epsilon)J_2^*$. *For (ii):* by Lemma 3.2, $J_2^* \ge \ell$, so equation 12 gives $J_2(\pi_\beta) \ge \ell - \epsilon\,\ell = (1-\epsilon)\,\ell = (1-\epsilon)\min\{J_1^*, J_2(\pi_1^*)\}$. □

B.5 Equation 4 (the drift term)

Proof. Fact A.2 with $\pi = \pi_1^*$ gives $J_1(\pi_1^*) - J_2(\pi_1^*) = \frac{1}{1-\gamma}\mathbb{E}_{\rho_1^{\pi_1^*}}[\delta(s)]$ as an identity, and $[x]_+ \ge x$. Since $J_2^* \ge J_2(\pi_1^*)$, $J_1^* - J_2^* \le J_1^* - J_2(\pi_1^*) = J_1(\pi_1^*) - J_2(\pi_1^*) \le B_{\text{eval}}$. □

B.6 Proposition 5.1 (deterministic validity of the shared-sample certificate)

Proposition B.1. deterministic validity of the shared-sample certificate For every state s and *every* realisation of the samples $z_1(s), \dots, z_K(s)$: (i) $\hat{d}_2(s) \le \hat{b}(s)$; (ii) $\mathrm{KL}(\pi_2^*(\cdot \mid s) \,\|\, \hat{\pi}_\beta(\cdot \mid s)) \le \hat{d}_2(s)$; (iii) $J_2^* - J_2(\hat{\pi}_\beta) \le \hat{G}_{\text{IB}}$; (iv) if $\ell > 0$ and $\hat{G}_{\text{IB}} \le \epsilon\,\ell$, then $J_2(\hat{\pi}_\beta) \ge (1-\epsilon)J_2^*$ and $J_2(\hat{\pi}_\beta) \ge (1-\epsilon)\,\ell$. In particular, sampling affects which policy is certified and the value of $\hat{G}_{\text{IB}}$, but it cannot make a certificate that fires unsound; no confidence level or estimation-error term is needed.

Proof. The proposition contains no new inequality. It is Theorems 3.1, 3.2, 3.3 and Lemma 3.1 applied with the mixing distribution $p(\cdot \mid s) := \hat{p}_s = \frac{1}{K}\sum_{k=1}^K \mathbf{1}\{z = z_k(s)\}$ and the decoder $q := \tilde{q}$, for which the population quantities *are* the hat quantities.

Fix a realisation of the samples at every state. Under the mixing distribution $\hat{p}_s$ and the decoder $\tilde{q}$, the population quantities of Theorem 1 and Lemma 1 evaluate to the hat quantities:

- $\mathbb{E}_{z\sim\hat{p}_s}\big[D_{\text{KL}}\pi_i^*(\cdot \mid s)\tilde{q}(\cdot \mid z)\big] = \hat{d}_i(s)$
- $\mathbb{E}_{z\sim\hat{p}_s}\big[\max_a(-\log\tilde{q}(a \mid z))\big] = \hat{L}(s)$
- $\mathbb{E}_{z\sim\hat{p}_s}\big[\tilde{q}(\cdot \mid z)\big] = \hat{\pi_\beta}(\cdot \mid s)$,

because an expectation under the uniform distribution on $\{z_1, \dots, z_K\}$ is the average over k. The hypotheses of the earlier results hold: $|\mathcal{A}| = m \ge 2$; $\tilde{q}_k(a \mid s) > 0$ for all a, k by construction; $\hat{d}_1, \hat{d}_2, \hat{L}$ are all averages under the same $\hat{p}_s$; π_2^* is supported on $\mathcal{O}(s)$; $\hat{L}(s) < \infty$ since it is a finite average of finite numbers.

(i) is the averaged form of Theorem 3.1 under $\hat{p}_s$: $d_2(s) = \hat{d}_2(s)$, $d_1(s) = \hat{d}_1(s)$, $\bar{L}(s) = \hat{L}(s)$, hence $b(s) = \hat{b}(s)$. (ii) is Step 1 of the proof of Lemma B.2 (convexity of the divergence over the mixture equation **??**). (iii) is Theorem 3.2 with $\pi_\beta = \hat{\pi_\beta}$ and $b = \hat{b}$; equivalently, Lemma B.2 gives $J_2^* - J_2(\hat{\pi_\beta}) \le \frac{1}{1-\gamma}\mathbb{E}_{\rho_2^{\hat{\pi_\beta}}}[\Delta A_2(s)(1 - e^{-\hat{d}_2(s)})]$ and (i) with monotonicity of $x \mapsto 1 - e^{-x}$ gives $\hat{G}_{\mathrm{IB}}$. (iv) is Theorem 3 with $G_{\mathrm{IB}} = \hat{G}_{\mathrm{IB}}$ and $\pi_\beta = \hat{\pi_\beta}$; Lemma 3.2 is unchanged since ℓ does not involve the agent.
Nothing above depends on how the samples were obtained, so the conclusions hold for every realisation. □

C WHY THE BOUND MUST LOOK LIKE THIS

Theorem 3.1 raises a natural question: why is the bound so complicated, and why is it not simply some distance between the two tasks? The terms it keeps are not free to drop. Two simpler alternatives fail by counterexample, and each failure rules out dropping one ingredient, which is why $b(s)$ takes exactly this form and not a lighter one. Table 2 summarises what a sufficient condition for near-optimal transfer must depend on and what is provably insufficient.

Condition for	Must depend on	Provably insufficient	Shown in
target distortion $d_2(s)$	the decoder, through $\bar{L}(s)$	any function of a divergence between π_1^* and π_2^* alone, including $2T(s)$	Proposition C.1
transfer error	rewards or dynamics, or π_2^* through the action gaps	any reward-blind rollout statistic of the deployed policy	Proposition C.2

Table 2: What a sufficient condition must depend on. Each row rules out one simpler bound by counterexample.

Distortion bounds must see the decoder. Could a task distance alone replace the task-disagreement penalty $2T(s)\bar{L}(s)$ of Theorem 3.1? That would give a bound of the form

$$d_2(s) \overset{?}{\le} d_1(s) + f\big(\mathrm{div}(\pi_1^*(\cdot \mid s), \pi_2^*(\cdot \mid s))\big). \tag{13}$$

The following proposition rules it out.

Proposition C.1. Distortion bounds must see the decoder At a state s where the two optimal policies differ, no bound of the form Eq. 13, with f finite and div any divergence between the two optimal policies' action distributions, holds uniformly over decoders with positive probabilities.

Counterexample. Let the decoder assign probability zero, or arbitrarily close to it, to an action that π_2^* requires at s but π_1^* does not. Then $d_2(s)$ is infinite, or grows without bound, while $d_1(s)$ stays finite and every divergence between $\pi_1^*(\cdot \mid s)$ and $\pi_2^*(\cdot \mid s)$ stays bounded (the Jensen–Shannon divergence is at most $\log 2$, for instance). So the decoder must enter the bound, which is what the sensitivity $\bar{L}(s)$ does.

Proof. Write $d_i(s, z) := D_{\mathrm{KL}}\pi_i^*(\cdot \mid s)q(\cdot \mid z)$ for the per-code KL divergence against task i's optimal policy, so $d_1(s)$ and $d_2(s)$ are its $p(z \mid s)$-average for $i = 1, 2$. Expanding d_1, d_2, as in the proof of Theorem 3.1,

$$d_2(s, z) - d_1(s, z) = \underbrace{H(\pi_1^*) - H(\pi_2^*)}_{\text{fixed: depends only on } \pi_1^*, \pi_2^*} + \sum_a \big(\pi_1^*(a \mid s) - \pi_2^*(a \mid s)\big) \log q(a \mid z).$$

Since $\pi_1^*(\cdot \mid s) \neq \pi_2^*(\cdot \mid s)$ and both are normalised, some action a has $\pi_2^*(a \mid s) > \pi_1^*(a \mid s)$. Hold every other decoder probability proportional to a fixed positive distribution and let $q(a \mid z) = \eta \downarrow 0$ (renormalising the rest to keep $q(\cdot \mid z)$ valid and everywhere positive). The a-term of the sum is $\big(\pi_1^*(a \mid s) - \pi_2^*(a \mid s)\big) \log \eta = \big(\pi_2^*(a \mid s) - \pi_1^*(a \mid s)\big) \log(1/\eta) \to \infty$ as $\eta \to 0$, every other

summand stays fixed because its own q-probability does, and the entropy difference is a constant. So $d_2(s,z) - d_1(s,z) \to \infty$ along this one-parameter family of decoders: $d_2(s) - d_1(s)$ is unbounded across it. But $f(\mathrm{div}(\pi_1^*(\cdot \mid s), \pi_2^*(\cdot \mid s)))$ never changes along this family – it is a fixed function of the two task policies, never of q. For η small enough, $d_2(s,z) - d_1(s,z)$ exceeds f by any margin, contradicting Eq. 13. □

This is exactly why Theorem 3.1 keeps $\bar{L}(s)$ rather than a decoder-free function of $T(s)$: dropping it leaves the bound unbounded on exactly the decoders that matter. A uniform decoder floor is one way to keep $\bar{L}(s)$ itself finite: $q_\eta(a \mid z) \geq \eta/m$ implies $L(z) \leq \log(m/\eta)$, at a cost of at most $\log(1/(1-\eta))$ nats relative to the unsmoothed pair. This is a sufficient construction, not the only way to control decoder probabilities, but some such control is unavoidable.

Reward information is also necessary. The model requirements above motivate cheaper behavioural diagnostics built from rollouts alone. However, state–action trajectories alone omit reward information needed to distinguish some successful transfers from failures.

Proposition C.2. Reward-blind statistics do not determine transfer error Across finite MDPs with no common reward-scale bound, no finite function of a fixed policy's reward-blind state–action trajectory distribution uniformly bounds its transfer error. *Construction:* take one state, two self-looping actions, and a fixed policy that always chooses action a. Give a reward zero and the other action reward $R > 0$. The policy's state–action trajectories do not depend on R, but its transfer error is $R/(1-\gamma)$.

Counterexample in words. Take two MDPs with identical dynamics that differ only in where the reward sits, among the states the deployed policy visits. The policy induces the same occupancy in both, so every rollout statistic, including the diagnostic $D_{\mathrm{JS}}^{\mathrm{enc}}$ of Appendix M, is exactly the same, while the transfer error can be made arbitrarily large.

Proof. The one-state policy always chooses its zero-reward action, so its return is zero and its trajectory law is fixed. The optimal policy chooses the action with reward R forever, giving $J^* = R/(1-\gamma)$. Any function of the trajectory law is therefore constant in R, while the transfer error grows without bound. With a fixed reward bound, the construction shows only that behaviour does not identify the transfer error, not that the error is unbounded. □

With a known bounded reward range, a worst-case transfer-error bound is available; the construction instead shows that identical behaviour can accompany different transfer errors. A small behavioural divergence alone does not imply near-optimal transfer, which is why our condition is built from exact models and target-optimal actions rather than from rollouts alone. This argument concerns reward-blind statistics, not all methods using rollouts, known reward bounds, or additional model assumptions. It is also why the rollout-based diagnostic of Appendix M cannot be promoted to a sufficient condition: a sufficient condition must hold in the worst case, whereas a diagnostic is judged by average-case accuracy, and no reward-blind statistic can provide the former.

D What early aggregation costs

The pointwise budget can be averaged under any target occupancy. Replacing its ingredients with global constants generally loses information about which states and codes are visited. Table 3 summarises measured costs from the frozen ablations. "Close pairs" means $10^{-4} < G_{\mathrm{exact}} < 0.05$ at zero decoder floor. The intermediate bound G_{local} of Appendix B, computed using measured target distortion, has median looseness $1.80\times$ on the zero-floor rows with transfer error $> 10^{-4}$; substituting the transfer budget raises it to $1.89\times$. Per configuration, the substitution multiplies the bound by 1.005 at the median and by 1.44 at the 90th percentile. This confirms that slack is present before the substitution.

Relaxation	Measured effect
$\bar{L}(s)$ to $\log(1/q_{\min})$	Eight certified instances lost in a separate distortion-budget study of 2,728 operating points.
$\Delta A_2(s)$ to $\max_s \Delta A_2(s)$, then $2V_{\max}$	90th-percentile looseness $1.99\times \to 25.92\times \to 1486\times$ in the 792-configuration return-bound study.
Pointwise budget to value-weighted scalar budget	Median close-pair looseness $3.5\times \to 44.9\times$; the scalar form saturates on 81% of the full sweep.
B_{eval} to global Lipschitz drift	Near-optimality recall $88.6\% \to 40.6\%$ on the same 13,200 configurations.

Table 3: Costs of early aggregation. The rows use distinct ablations and are not a single sequential experiment.

E THE DRIFT LADDER

Fix $\pi = \pi_1^*$ and let $\delta(s)$ be the drift in Fact **??**. Write $\rho = \rho_1^\pi$, $\Delta r = r_1^\pi - r_2^\pi$, and $V = V_2^\pi$. Five nonnegative upper bounds on $J_1^* - J_2^*$ are

$$B_{\text{eval}} = \tfrac{1}{1-\gamma}[\mathbb{E}_\rho \delta]_+, \qquad B_+ = \tfrac{1}{1-\gamma}\mathbb{E}_\rho[\delta]_+, \quad B_{\text{abs}} = \tfrac{1}{1-\gamma}\mathbb{E}_\rho|\delta|,$$
$$B_{W,V} = \tfrac{1}{1-\gamma}\mathbb{E}_\rho\big[|\Delta r| + \gamma W_1(V_\# P_1^\pi, V_\# P_2^\pi)\big],$$
$$B_{W,L} = \tfrac{1}{1-\gamma}\mathbb{E}_\rho\big[|\Delta r| + \gamma \,\mathrm{Lip}(V) W_1(P_1^\pi, P_2^\pi)\big].$$

Here $V_\# P$ is the distribution of next-state values under P; W_1 uses absolute distance for values and Manhattan distance for grid states. $\mathrm{Lip}(V)$ is the Lipschitz constant under that ground metric. Convexity of the positive part, the triangle inequality, and the transport inequality give $B_{\text{eval}} \le B_+ \le B_{\text{abs}} \le B_{W,V} \le B_{W,L}$. Substitution of $B_{W,L}$ for B_{eval} reduces recall from 88.6% to 40.6%. All rungs evaluate the same source-optimal policy; none requires optimising a target policy.

F BENCHMARK DETAILS

Figure 4 gives an overview of both benchmarks; the full sets of environments follow in Figures 5 and 6.

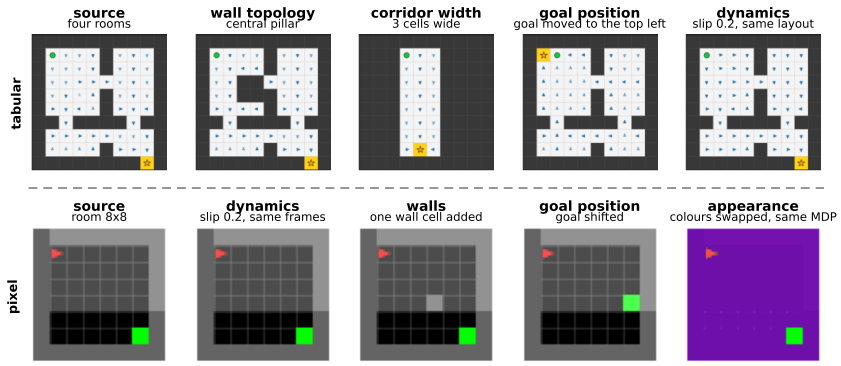

Figure 4: Controlled shifts. *Top:* the tabular source `four_rooms` and targets varying topology, corridor width, goal position, and dynamics via slip (star = goal, dot = start, arrows = optimal policy). *Bottom:* pixel source together with slip, wall, goal, and colour variants (§5, red triangle = agent, green square = goal). Slip preserves the rendered frames and the colour swap preserves the MDP structure.

F.1 TABULAR BENCHMARK: TRAINING AND EVALUATION

The tabular benchmark comprises 34 grid-worlds with at most 100 states and four compass actions (Figure 4), varying wall topology, corridor width, goal position, and stochastic dynamics (moves

are deterministic and bounce off walls, except in the slip variants, where a move can slip to either of the two perpendicular directions); a single absorbing goal state gives reward $+1$ on entry and every other reward is zero, with $\gamma = 0.99$. Pairing the eight same-size sources below with the 33 other targets, at five compression levels, two decoder floors, and five seeds, gives 13,200 configurations. The eight sources are four_rooms, open_room, corridor_w4, single_door, reward_top_left, distractor_pillars, dynamics_slip_02, and blocked_room. Only same-size pairs are used. Table 4 groups all 34 environments by family, and Figure 5 shows every one of them.

Family	Environments	What varies vs. four_rooms
Base / open (2)	four_rooms, open_room	—
Wall & door topology (18)	wall_shifted_doors, wall_extra_segment, wall_two_rooms, one_door_moved, all_doors_moved, three_doors, offset_vertical_wall, l_barrier, zigzag_walls, diagonal_walls, central_pillar, distractor_pillars, blocked_room, blocked_top_right, narrow_passage, single_door, wide_doorways, extra_corridor	wall layout / doorway position (blocked transitions; reward position fixed)
Corridors (4)	corridor_w2, corridor_w3, corridor_w4, corridor_w5	corridor width 2–5
Reward moves (5)	reward_top_left, reward_top_right, reward_bottom_left, open_room_goal_left, open_room_goal_topright	goal position (reward function)
Stochastic dynamics (5)	dynamics_slip_01, dynamics_slip_02, dynamics_slip_03, open_room_slip_02, combined_doors_slip	transition stochasticity (pure-dynamics shift)

Table 4: The 34 tabular environments, by family. All share 100 states, four compass actions, and $\gamma = 0.99$. The eight sources named above are among these 34; every other same-size environment in the pool is a target for each source.

Tabular compression coefficients are $\{2, 5, 10, 20, 50\}$, decoder floors are $\{0, 0.01\}$, and seeds are 42–46. Blahut–Arimoto, as adapted to the bottleneck objective by Tishby et al. (2000), solves the tabular bottleneck exactly, with three restarts, at most 500 iterations, and stopping tolerance 10^{-6}. Its training joint is estimated from 5,000 source transitions, with episodes capped at 200 steps. Evaluation quantities and those of the sufficient condition use exact model computations. The smoothed decoder is $q_\eta = (1 - \eta)q + \eta/m$, and the policy is rebuilt from that decoder. The initial distribution is uniform over shared floor states. Goal states are absorbing with zero subsequent reward; slip changes both transitions and the expected immediate reward of entering a goal.

F.2 Pixel benchmark: training and evaluation

The full-frame CNN uses a two-dimensional Gaussian latent and a decoder with hidden width 64. It is trained as a variational bottleneck (Alemi et al., 2019), whose rate term only upper-bounds $I(S; Z)$; this does not affect validity, since the sufficient condition is evaluated on the pair as deployed. Training uses soft source-optimal action labels, learning rate 10^{-3}, 1,500 epochs, and full batches of valid states. Neural compression coefficients are $\{0.001, 0.005, 0.02, 0.1, 0.5, 2\}$ and seeds are 42–46. This coefficient weights the compression penalty, so larger values mean *more* compression – the opposite direction to tabular β. Certification fixes $K = 200$ shared latent samples per state. A separate sensitivity sweep varies K over $\{10, 25, 50, 100, 200, 400, 800\}$ for colour and wall targets. The full frame encodes the finite position–orientation state; the partial-view ablation can alias states. Table 6 gives the per-target budget terms behind §6.2. Table 5 reports the seven main targets, and Figure 6 shows each as the agent sees it; counts aggregate all 30 configurations per target.

G Per-target results, all 34 targets

Table 10 lists every target; Table 13 (Appendix O) groups them by recall.



Figure 5: The 34 tabular environments, in the family order of Table 4: walls, goal, a start cell, and the optimal policy as arrows. The eight sources are in bold.

Target	J_2^*	B_{eval}	0.1ℓ	Successes	Certified
colourswap	0.9465	0.0000	0.0947	0	0
goal_far	0.9464	0.8747	0.0072	0	0
goal_shift	0.9556	0.5786	0.0367	0	0
slip_01	0.9387	0.0078	0.0939	25	25
slip_02	0.9292	0.0173	0.0929	25	25
wall_cell	0.9454	0.0827	0.0864	25	4
wall_segment	0.9454	0.1786	0.0769	4	0

Table 5: Pixel targets: model-derived return scale and observed success and certification counts. Values are medians across rows; the reference-policy quantities are independent of compression.

Target	$d_1^{(1)}$	Repr. shift	$2T\bar{L}$	B_{eval}	Largest component
colourswap	0.07	**9.86**	0.00	0.000	representation shift
slip_01	0.19	0.00	0.00	0.008	source distortion, certified
slip_02	0.19	0.00	0.00	0.017	source distortion, certified
wall_cell	0.18	1.05	2.48	0.083	representation and task
wall_segment	0.16	1.11	5.10	0.179	task disagreement
goal_shift	0.12	0.03	**11.0**	0.579	task disagreement
goal_far	0.13	0.12	**7.58**	0.875	task disagreement

Table 6: Occupancy-weighted averages of the distortion-budget terms over the target's states in bits, and drift B_{eval} in return units, medians over five seeds at VIB compression coefficient 0.1. The task column is $2T\bar{L}$, and entropy corrections are omitted.

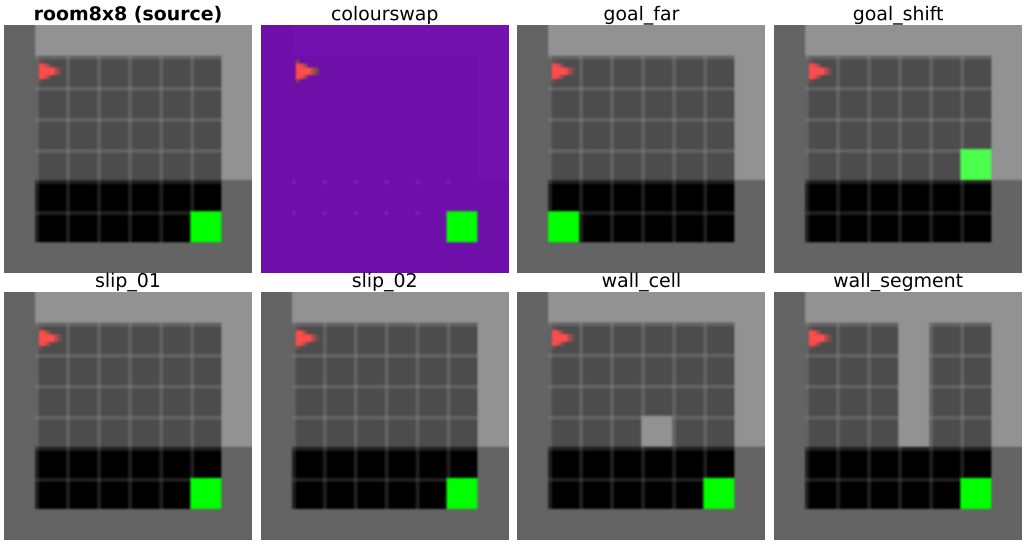

Figure 6: The pixel benchmark: the MiniGrid source room and its seven targets, each rendered from the start state as the encoder sees it.

Target	Rows	Successes	Certified	False	Recall	$A\prime$
all_doors_moved	400	0	0	0	—	0
blocked_room	350	0	0	0	—	0
blocked_top_right	400	100	23	0	0.230	85
central_pillar	400	0	0	0	—	0
combined_doors_slip	400	0	0	0	—	0
corridor_w2	400	50	50	0	1.000	50
corridor_w3	400	50	50	0	1.000	50
corridor_w4	350	0	0	0	—	0
corridor_w5	400	50	50	0	1.000	50
diagonal_walls	400	100	100	0	1.000	100
distractor_pillars	350	100	100	0	1.000	100
dynamics_slip_01	400	100	90	0	0.900	100
dynamics_slip_02	350	50	40	0	0.800	40
dynamics_slip_03	400	100	80	0	0.800	90
extra_corridor	400	100	100	0	1.000	100
four_rooms	350	50	50	0	1.000	50
l_barrier	400	100	100	0	1.000	100
narrow_passage	400	100	100	0	1.000	100
offset_vertical_wall	400	50	50	0	1.000	50
one_door_moved	400	10	0	0	0.000	0
open_room	350	100	100	0	1.000	100
open_room_goal_left	400	0	0	0	—	0
open_room_goal_topright	400	0	0	0	—	0
open_room_slip_02	400	120	45	0	0.375	120
reward_bottom_left	400	0	0	0	—	0
reward_top_left	350	0	0	0	—	0
reward_top_right	400	0	0	0	—	0
single_door	350	100	100	0	1.000	100
three_doors	400	0	0	0	—	0
wall_extra_segment	400	100	100	0	1.000	100
wall_shifted_doors	400	0	0	0	—	0
wall_two_rooms	400	100	93	0	0.930	100
wide_doorways	400	100	100	0	1.000	100
zigzag_walls	400	100	100	0	1.000	100

Table 7: Full tabular results. Success is target-relative near-optimality at tolerance 0.1. A dash means no successful transfer was observed.

H PER-STATE BUDGET TRACKING

The per-state bound is $\Delta A_2(s)(1 - e^{-b(s)})$ and the actual per-state loss is $\sum_a \hat{\pi}_\beta(a \mid s)\Delta_2(s, a)$, before occupancy weighting. The reported median Spearman correlation is 0.81, and Figure 8 plots the two quantities.

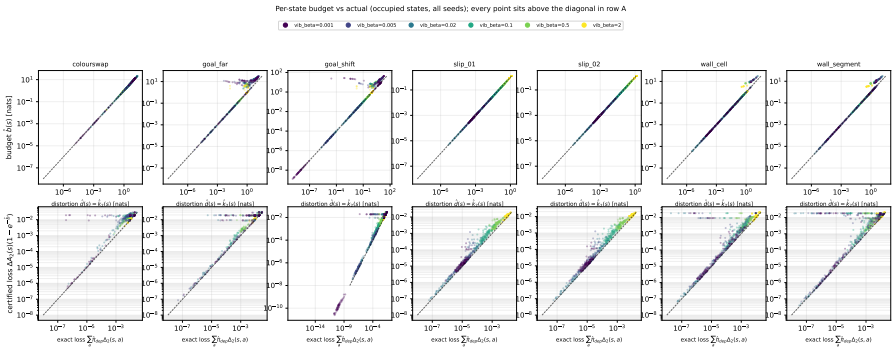

Figure 7: Per-state bound and loss in the neural study. This exhibit uses saved per-state outputs; the scalar configuration tables alone do not reconstruct it.

I BEHAVIOURAL DIAGNOSTICS WITHOUT A TARGET SOLUTION

A *diagnostic* is a complementary object to a certificate: a predictor of transfer success from divergences computable between the two tasks, judged by its average accuracy on a set of instances rather than by a proof. A diagnostic may be wrong on some instances; a certificate may not. Reward changes can leave a fixed policy's state–action trajectories, and hence any reward-blind diagnostic built from them, unchanged while transfer error grows without bound, which is why a diagnostic cannot in general be strengthened into a certificate (Proposition C.2). When target-optimal information is unavailable, the certificate itself cannot be evaluated. We therefore assess cheaper diagnostics as predictors here, using ranking accuracy rather than a soundness guarantee.

For this separate study, success means retaining 90% of the deployed policy's *source* return, $J_2(\pi_\beta) \geq 0.9J_1(\pi_\beta)$, using rollout estimates. These labels differ from the certificate's target-optimality criterion, so the reported accuracy is not evidence of near-optimality prediction.

The encoder-level metric $D_{\mathrm{JS}}^{\mathrm{enc}}$ averages the Jensen–Shannon divergence between empirical action distributions from source and target rollouts of the same policy, weighted by their mean state occupancy. It requires no target solution. In this tabular setting the exact conditional policy is unchanged, so this empirical metric depends on finite sampling and state coverage; it is not a distance between different exact policy functions. The task-level variant compares optimal policies instead.

We compared twelve divergences on 792 configurations grouped by source and compression level. We report orientation-aware AUC, $\max\{\mathrm{AUC}, 1 - \mathrm{AUC}\}$, and *frontier purity*: the fraction of successes inside the smallest axis-aligned rectangle containing all successes in the plane of a policy metric and a dynamics metric. Because this rectangle uses the observed success labels, purity describes separation within this dataset; it is not a held-out guarantee. Table 11 shows three representative metrics; the full ranking is in Appendix N. The encoder-level divergence $D_{\mathrm{JS}}^{\mathrm{enc}}$ achieved mean oriented AUC 0.972 and frontier purity 0.941 without solving the target. The dynamics baseline, $D_{\mathrm{Dyn}} = \max_a\{|\mathbb{E}_s[r_1 - r_2]| + \gamma\mathbb{E}_s[W_1(P_1(\cdot \mid s, a), P_2(\cdot \mid s, a))]\}$, uses the models directly and follows the reward–transition comparison used in state-similarity methods (Ferns et al., 2004). Although its overall ranking was weaker, it detected dynamics changes that left the policy-based metrics unchanged, motivating its use as the second frontier axis.

High ranking accuracy does not make $D_{\mathrm{JS}}^{\mathrm{enc}}$ a certificate. Reward changes can leave a fixed policy's state–action trajectories unchanged, giving zero behavioural divergence despite positive transfer error (Proposition C.2). Additional assumptions or reward information would be needed to turn this diagnostic into a sufficient condition for transfer.

Metric	Target solved?	oriented AUC	2-D frontier purity
$D_{\mathrm{JS}}^{\mathrm{enc}}$ — rollout-based predictor	**no**	**0.972**	**0.941**
$D_{\mathrm{JS}}^{\mathrm{task}}$ — optimal-policy divergence	yes	0.967	0.868
D_{Dyn} — bisimulation-style baseline	no (models only)	0.689	0.348

Table 8: Behavioural prediction. Means over the 21 (source, β) groups of 792 configurations. $D_{\mathrm{JS}}^{\mathrm{enc}}$ compares how differently the *same* compressed policy behaves in the two tasks, needing no target solution; the encoder-level metric achieved higher mean purity in this study.

J THE FULL TWELVE-METRIC RANKING

The diagnostic study uses rollout-estimated source-relative success, a single training seed, and 21 source–compression groups. The 192 successes among 792 configurations are not the near-optimality labels used for certification. AUC orientation and frontier rectangles are selected using observed labels; the summaries below are descriptive, not held-out calibration. Table 12 ranks all twelve metrics.

For either variant, let u_1, u_2 be the compared action distributions and ρ_1, ρ_2 their state weights. The task variant uses optimal references; the encoder variant uses empirical action frequencies from the fixed policy's rollouts. With $T(s) = \mathrm{TV}(u_1, u_2)(s)$ and $\bar{\rho} = (\rho_1 + \rho_2)/2$, the metrics are

$$
\begin{aligned}
D_{\mathrm{JS}} &= \mathbb{E}_{\bar{\rho}}\mathrm{JSD}(u_1, u_2), & \mathrm{TV}_{\mathrm{avg}} &= \mathbb{E}_{\rho_2} T, \\
H_{\mathrm{gap}} &= \mathbb{E}_{\rho_2}[h(T) + T\log(m-1)], & L_{\mathrm{eff}} &= \mathbb{E}_{\rho_2}\bar{L}, \\
X_\beta^{\mathrm{TV}} &= 2\log(1/q_{\min})\mathrm{TV}_{\mathrm{avg}} + H_{\mathrm{gap}}, & & \\
X_\beta^{\mathrm{TV,loc}} &= 2\mathbb{E}_{\rho_2}[T\bar{L}] + H_{\mathrm{gap}}, & X_\beta^{\mathrm{TV,eff}} &= 2L_{\mathrm{eff}}\mathrm{TV}_{\mathrm{avg}} + H_{\mathrm{gap}}.
\end{aligned}
$$

Here $q_{\min}$ is the smallest decoder probability on active codes, and ρ_2 is renormalised on shared valid states for the TV-based quantities. The effective variant is a heuristic: separating T and $\bar{L}$ need not preserve an upper bound. The dynamics metric in Appendix M uses uniform weights over shared valid states and Manhattan transport cost.

Metric	Oriented AUC	Frontier purity
$D_{\mathrm{JS}}^{\mathrm{enc}}$	0.972	0.941
$D_{\mathrm{JS}}^{\mathrm{task}}$	0.967	0.868
$X_\beta^{\mathrm{TV,loc,task}}$	0.947	0.817
$X_\beta^{\mathrm{TV,eff,task}}$	0.953	0.814
$\mathrm{TV}_{\mathrm{avg}}^{\mathrm{task}}$	0.953	0.814
$X_\beta^{\mathrm{TV,task}}$	0.952	0.814
$X_\beta^{\mathrm{TV,eff,enc}}$	0.870	0.744
$X_\beta^{\mathrm{TV,enc}}$	0.868	0.744
$X_\beta^{\mathrm{TV,loc,enc}}$	0.867	0.744
$\mathrm{TV}_{\mathrm{avg}}^{\mathrm{enc}}$	0.858	0.739
$L_{\mathrm{eff}}^{\mathrm{enc}}$	0.791	0.484
D_{Dyn}	0.689	0.348

Table 9: Full diagnostic ranking, ordered by mean two-dimensional frontier purity over the 21 groups. Each frontier combines the listed metric with dynamics distance.

K PER-TARGET RESULTS, ALL 34 TARGETS

Table 10 lists every target; Table 13 (Appendix O) groups them by recall.

Target	Rows	Successes	Certified	False	Recall	$A\prime$
all_doors_moved	400	0	0	0	—	0
blocked_room	350	0	0	0	—	0
blocked_top_right	400	100	23	0	0.230	85
central_pillar	400	0	0	0	—	0
combined_doors_slip	400	0	0	0	—	0
corridor_w2	400	50	50	0	1.000	50
corridor_w3	400	50	50	0	1.000	50
corridor_w4	350	0	0	0	—	0
corridor_w5	400	50	50	0	1.000	50
diagonal_walls	400	100	100	0	1.000	100
distractor_pillars	350	100	100	0	1.000	100
dynamics_slip_01	400	100	90	0	0.900	100
dynamics_slip_02	350	50	40	0	0.800	40
dynamics_slip_03	400	100	80	0	0.800	90
extra_corridor	400	100	100	0	1.000	100
four_rooms	350	50	50	0	1.000	50
l_barrier	400	100	100	0	1.000	100
narrow_passage	400	100	100	0	1.000	100
offset_vertical_wall	400	50	50	0	1.000	50
one_door_moved	400	10	0	0	0.000	0
open_room	350	100	100	0	1.000	100
open_room_goal_left	400	0	0	0	—	0
open_room_goal_topright	400	0	0	0	—	0
open_room_slip_02	400	120	45	0	0.375	120
reward_bottom_left	400	0	0	0	—	0
reward_top_left	350	0	0	0	—	0
reward_top_right	400	0	0	0	—	0
single_door	350	100	100	0	1.000	100
three_doors	400	0	0	0	—	0
wall_extra_segment	400	100	100	0	1.000	100
wall_shifted_doors	400	0	0	0	—	0
wall_two_rooms	400	100	93	0	0.930	100
wide_doorways	400	100	100	0	1.000	100
zigzag_walls	400	100	100	0	1.000	100

Table 10: Full tabular results. Success is target-relative near-optimality at tolerance 0.1. A dash means no successful transfer was observed.

L PER-STATE BUDGET TRACKING

The per-state bound is $\Delta A_2(s)(1 - e^{-b(s)})$ and the actual per-state loss is $\sum_a \hat{\pi}_\beta(a \mid s)\Delta_2(s, a)$, before occupancy weighting. The reported median Spearman correlation is 0.81, and Figure 8 plots the two quantities.

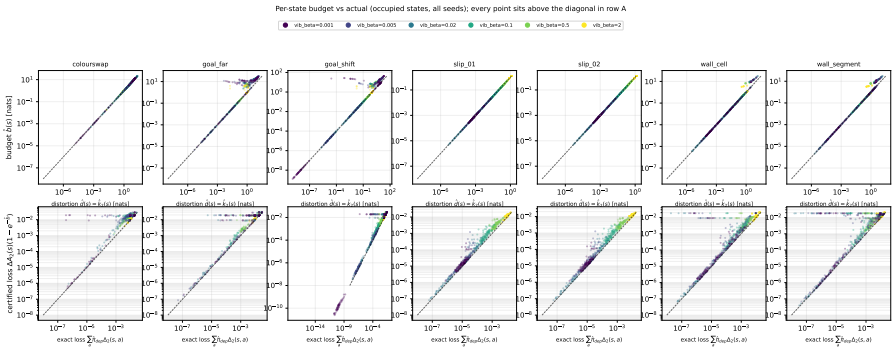

Figure 8: Per-state bound and loss in the neural study. This exhibit uses saved per-state outputs; the scalar configuration tables alone do not reconstruct it.

M BEHAVIOURAL DIAGNOSTICS WITHOUT A TARGET SOLUTION

A *diagnostic* is a complementary object to a certificate: a predictor of transfer success from divergences computable between the two tasks, judged by its average accuracy on a set of instances rather than by a proof. A diagnostic may be wrong on some instances; a certificate may not. Reward changes can leave a fixed policy's state–action trajectories, and hence any reward-blind diagnostic built from them, unchanged while transfer error grows without bound, which is why a diagnostic cannot in general be strengthened into a certificate (Proposition C.2). When target-optimal information is unavailable, the certificate itself cannot be evaluated. We therefore assess cheaper diagnostics as predictors here, using ranking accuracy rather than a soundness guarantee.

For this separate study, success means retaining 90% of the deployed policy's *source* return, $J_2(\pi_\beta) \geq 0.9J_1(\pi_\beta)$, using rollout estimates. These labels differ from the certificate's target-optimality criterion, so the reported accuracy is not evidence of near-optimality prediction.

The encoder-level metric $D_{\mathrm{JS}}^{\mathrm{enc}}$ averages the Jensen–Shannon divergence between empirical action distributions from source and target rollouts of the same policy, weighted by their mean state occupancy. It requires no target solution. In this tabular setting the exact conditional policy is unchanged, so this empirical metric depends on finite sampling and state coverage; it is not a distance between different exact policy functions. The task-level variant compares optimal policies instead.

We compared twelve divergences on 792 configurations grouped by source and compression level. We report orientation-aware AUC, $\max\{\mathrm{AUC}, 1 - \mathrm{AUC}\}$, and *frontier purity*: the fraction of successes inside the smallest axis-aligned rectangle containing all successes in the plane of a policy metric and a dynamics metric. Because this rectangle uses the observed success labels, purity describes separation within this dataset; it is not a held-out guarantee. Table 11 shows three representative metrics; the full ranking is in Appendix N. The encoder-level divergence $D_{\mathrm{JS}}^{\mathrm{enc}}$ achieved mean oriented AUC 0.972 and frontier purity 0.941 without solving the target. The dynamics baseline, $D_{\mathrm{Dyn}} = \max_a\{|\mathbb{E}_s[r_1 - r_2]| + \gamma\mathbb{E}_s[W_1(P_1(\cdot \mid s, a), P_2(\cdot \mid s, a))]\}$, uses the models directly and follows the reward–transition comparison used in state-similarity methods (Ferns et al., 2004). Although its overall ranking was weaker, it detected dynamics changes that left the policy-based metrics unchanged, motivating its use as the second frontier axis.

High ranking accuracy does not make $D_{\mathrm{JS}}^{\mathrm{enc}}$ a certificate. Reward changes can leave a fixed policy's state–action trajectories unchanged, giving zero behavioural divergence despite positive transfer error (Proposition C.2). Additional assumptions or reward information would be needed to turn this diagnostic into a sufficient condition for transfer.

Metric	Target solved?	oriented AUC	2-D frontier purity
$D_{\mathrm{JS}}^{\mathrm{enc}}$ — rollout-based predictor	**no**	**0.972**	**0.941**
$D_{\mathrm{JS}}^{\mathrm{task}}$ — optimal-policy divergence	yes	0.967	0.868
D_{Dyn} — bisimulation-style baseline	no (models only)	0.689	0.348

Table 11: Behavioural prediction. Means over the 21 (source, β) groups of 792 configurations. $D_{\mathrm{JS}}^{\mathrm{enc}}$ compares how differently the *same* compressed policy behaves in the two tasks, needing no target solution; the encoder-level metric achieved higher mean purity in this study.

N THE FULL TWELVE-METRIC RANKING

The diagnostic study uses rollout-estimated source-relative success, a single training seed, and 21 source–compression groups. The 192 successes among 792 configurations are not the near-optimality labels used for certification. AUC orientation and frontier rectangles are selected using observed labels; the summaries below are descriptive, not held-out calibration. Table 12 ranks all twelve metrics.

For either variant, let u_1, u_2 be the compared action distributions and ρ_1, ρ_2 their state weights. The task variant uses optimal references; the encoder variant uses empirical action frequencies from the fixed policy's rollouts. With $T(s) = \mathrm{TV}(u_1, u_2)(s)$ and $\bar{\rho} = (\rho_1 + \rho_2)/2$, the metrics are

$$
\begin{aligned}
D_{\mathrm{JS}} &= \mathbb{E}_{\bar{\rho}}\mathrm{JSD}(u_1, u_2), & \mathrm{TV}_{\mathrm{avg}} &= \mathbb{E}_{\rho_2} T, \\
H_{\mathrm{gap}} &= \mathbb{E}_{\rho_2}[h(T) + T\log(m-1)], & L_{\mathrm{eff}} &= \mathbb{E}_{\rho_2}\bar{L}, \\
X_\beta^{\mathrm{TV}} &= 2\log(1/q_{\min})\mathrm{TV}_{\mathrm{avg}} + H_{\mathrm{gap}}, & & \\
X_\beta^{\mathrm{TV,loc}} &= 2\mathbb{E}_{\rho_2}[T\bar{L}] + H_{\mathrm{gap}}, & X_\beta^{\mathrm{TV,eff}} &= 2L_{\mathrm{eff}}\mathrm{TV}_{\mathrm{avg}} + H_{\mathrm{gap}}.
\end{aligned}
$$

Here $q_{\min}$ is the smallest decoder probability on active codes, and ρ_2 is renormalised on shared valid states for the TV-based quantities. The effective variant is a heuristic: separating T and $\bar{L}$ need not preserve an upper bound. The dynamics metric in Appendix M uses uniform weights over shared valid states and Manhattan transport cost.

Metric	Oriented AUC	Frontier purity
$D_{\mathrm{JS}}^{\mathrm{enc}}$	0.972	0.941
$D_{\mathrm{JS}}^{\mathrm{task}}$	0.967	0.868
$X_\beta^{\mathrm{TV,loc,task}}$	0.947	0.817
$X_\beta^{\mathrm{TV,eff,task}}$	0.953	0.814
$\mathrm{TV}_{\mathrm{avg}}^{\mathrm{task}}$	0.953	0.814
$X_\beta^{\mathrm{TV,task}}$	0.952	0.814
$X_\beta^{\mathrm{TV,eff,enc}}$	0.870	0.744
$X_\beta^{\mathrm{TV,enc}}$	0.868	0.744
$X_\beta^{\mathrm{TV,loc,enc}}$	0.867	0.744
$\mathrm{TV}_{\mathrm{avg}}^{\mathrm{enc}}$	0.858	0.739
$L_{\mathrm{eff}}^{\mathrm{enc}}$	0.791	0.484
D_{Dyn}	0.689	0.348

Table 12: Full diagnostic ranking, ordered by mean two-dimensional frontier purity over the 21 groups. Each frontier combines the listed metric with dynamics distance.

O SUPPORTING EXHIBITS

P FURTHER DISCUSSION

Oracle-free prediction. Our construction isolates what a prediction of transfer made without solving the target must supply: the target solution enters only through task-level quantities that do not

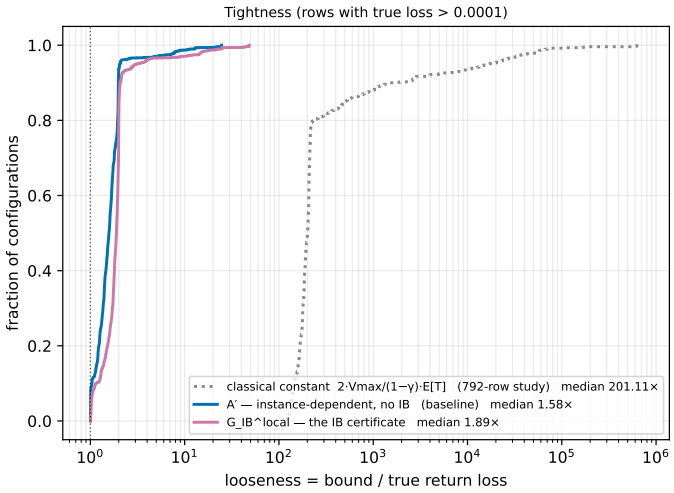

Figure 9: Slack in the transfer-error bounds used by Theorem 3.3. Curves show cumulative distributions of bound divided by actual transfer error, excluding transfer error $\leq 10^{-4}$; further left is tighter. The zero-floor tabular sweep gives medians $1.89\times$ for G_{IB} and $1.58\times$ for A' (Table 1). The classical comparison ($201\times$) uses the separate 792-configuration return-bound study.

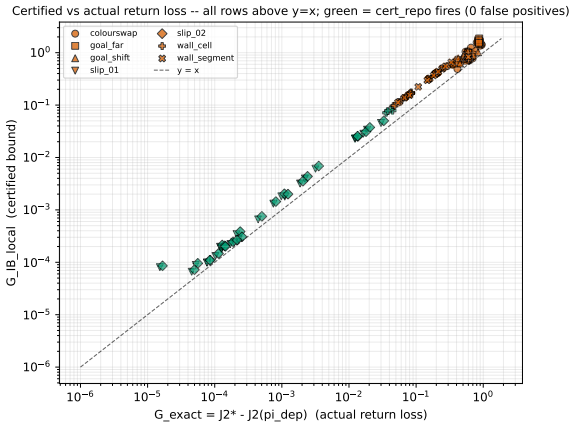

Figure 10: Transfer-error bound versus actual transfer error for the empirical neural policy, on logarithmic axes (summarised in Table 1). All plotted bounds lie above the diagonal; green points satisfy the sufficient condition of Theorem 3.3. Figure 3 (§6.2) extends this by decomposing the budget into source distortion, representation shift, and task disagreement for three representative targets.

depend on the compressed policy, and the drift needs only the return of the source-optimal policy on the target. Some target information is unavoidable (Appendix C); how much can be obtained without solving the target remains open.

State representation learning. Representation-learning objectives encourage invariance to irrelevant observation changes through bisimulation (Zhang et al., 2021) or learn abstractions that generalise across tasks by predicting reward (Lehnert et al., 2020). Our decomposition suggests targeting explicit terms of the transfer bound instead: penalising representation shift for invariance to observation changes, and controlling decoder sensitivity to limit the cost of task disagreement. That no tested compression level made the full-frame encoder colour-invariant (§6.2) motivates such objectives.

target group	rows	successes	certified	recall	cert. by A'
15 targets at full recall	5,800	1,250	1,250	1.00	1,250
`wall_two_rooms`	400	100	93	0.93	100
`dynamics_slip_01`	400	100	90	0.90	100
`dynamics_slip_03`	400	100	80	0.80	90
`dynamics_slip_02`	350	50	40	0.80	40
`open_room_slip_02`	400	120	45	0.375	120
`blocked_top_right`	400	100	23	0.23	85
`one_door_moved`	400	10	0	0.00	0
12 targets with no genuine success	4,650	0	0	—	0
all 34 targets	**13,200**	**1,830**	**1,621**	**0.886**	**1,785**

Table 13: Recall at zero false positives. The fifteen targets at full recall are listed individually in Appendix K, as are the twelve on which no compression level transfers at all — there is nothing there to certify.