Process Matters more than Output for Distinguishing Humans from Machines

Milena Rmus
Roundtable Technologies Inc.
milena@roundtable.ai

Mathew D. Hardy
Roundtable Technologies Inc.
matt@roundtable.ai

Thomas L. Griffiths
Princeton University
tomg@princeton.edu

Mayank Agrawal
Roundtable Technologies Inc.
mayank@roundtable.ai

Abstract

Reliable human-machine discrimination is becoming increasingly important as Large Language Models and autonomous agents are deployed in online settings. Existing approaches largely evaluate whether a system can produce behavior or responses indistinguishable from those of a human. This approach follows the focus on the *output* of a machine as a criterion for determining whether it can think, as suggested by Alan Turing. Cognitive science provides an alternative approach: considering the *process* by which that behavior is produced. To evaluate whether differences in cognitive mechanisms can be used to reliably distinguish humans and machines, we introduce **CogCaptcha30**, a battery of 30 cognitive tasks designed to provide diagnostic process-level features even when task performance is matched. Across the battery, process-level features provide substantially stronger discriminative signal than performance metrics alone, reliably distinguishing humans from agents even when task performance is matched (mean process-feature classifier AUC = 0.88). To assess agentic process limitations, we conduct a controlled red-teaming study comparing off-the-shelf frontier agents (Claude Sonnet 4.5, GPT-5, Gemini 2.5 Pro), Centaur (a large language model fine-tuned on 10.7M human decisions), and two task-specific fine-tuning methods applied to Qwen2.5-1.5B-Instruct: **action-level supervised fine-tuning (A-SFT)** and **process-level fine-tuning (P-SFT)**, which directly optimizes process features. We find that broad fine-tuning on human choices makes task processes more human-like relative to off-the-shelf frontier agents, and task-specific process-level fine-tuning further improves human-like behavioral mimicry though this advantage diminishes with cross-task transfer when process targets do not naturally generalize across tasks. These results suggest that explicit process-level supervision can substantially improve human behavioral mimicry, but only when appropriate task-specific process representations are available. This highlights process specification as a central bottleneck in achieving human-like cognitive processes in machines.

1 Introduction

Reliable human-machine discrimination is becoming increasingly important as large language models and autonomous agents are deployed in online settings, creating new challenges for security, fraud prevention, and platform integrity [Weidinger et al., 2022, Yao et al., 2024, Park et al., 2024]. Existing approaches to distinguishing humans from machines are largely task performance-based, evaluating whether a system can produce behavior or responses indistinguishable from those of a human. This

Preprint.

paradigm underlies systems such as CAPTCHA-style challenge-response tests and automated content detectors, and reflects the enduring influence of the Turing Test, which answered "Can machines think?" by testing whether machines could produce human-like output [Turing, 1950, Von Ahn et al., 2003].

Modern AI systems increasingly satisfy this output-based bar. GPT-4.5 is judged human 73% of the time in controlled Turing Tests [Jones and Bergen, 2025], frontier vision models can solve reCAPTCHAv2 challenges [Plesner et al., 2024], and specialized AI systems now match or exceed human experts in domains once considered canonical tests of intelligence, including chess and Go [Silver et al., 2017, Campbell et al., 2002]. As AI systems approach human-level performance on many output-based evaluations, output is an increasingly weak criterion for distinguishing humans and machines.

However, even when humans and AI systems produce similar task outputs, the latent processes generating those outputs may differ substantially. Artificial systems can arrive at the same answers through strategies that diverge from those used by humans, including reliance on spurious shortcuts, non-robust features, or memorized mappings [Geirhos et al., 2020, Lapuschkin et al., 2019, Block, 1981, Regan et al., 2014]. This motivates examining whether process-level behavioral features provide discriminative information beyond task performance and outputs alone. Because such signatures reflect latent constraints and strategies governing how decisions are generated rather than merely whether a task is solved successfully, they may remain informative even when human and machine outputs converge.

To leverage process traces for distinguishing humans and agents, we lean on insights and tools from cognitive science, a field that has long used structured behavioral tasks to infer underlying cognitive mechanisms from measurable regularities in behavior. Examples include working memory limitations [Cowan, 2001, Collins and Frank, 2012], systematic trial-to-trial adaptation following errors [Rabbitt, 1966], and individual differences in sensitivity to wins and losses [Nowak and Sigmund, 1993]. These patterns provide candidate *process-level features* that can distinguish humans from agents even when task performance is matched.

However, identifying informative process-level signatures is only part of the challenge: for such signals to serve as robust discriminators, they must remain difficult for agents to reproduce even under targeted adaptation. Modern AI models are highly adaptable, and can be optimized through fine-tuning or reinforcement learning toward target behavioral objectives, including alignment with human preferences and response styles [Ouyang et al., 2022, Bai et al., 2022, Rafailov et al., 2023, Binz et al., 2025]. This raises a deeper question: if process-level features distinguish humans from agents, what information is required for an AI system to reproduce them? More specifically, when does explicit process-level supervision provide advantages beyond large-scale action imitation? We investigate this question in two stages:

First, how much additional discriminative value does process provide beyond output alone? To evaluate this, we introduce **CogCaptcha30**, a battery of 30 cognitive tasks drawn from domains with extensively operationalized process-level behavioral signatures. These tasks were selected to leverage well-characterized differences in *how* humans solve problems—not merely whether they solve them—thereby providing a rich testbed for comparing human and agent behavior. Across the battery, we find that process-level behavioral features provide substantially greater discriminative signal than performance metrics alone (e.g., accuracy, earned points), revealing systematic human-agent differences even when task performance is similar.

Second, what limits an agent's ability to reproduce human-like process? We compare frontier agents (Claude Sonnet 4.5, GPT-5, Gemini 2.5 Pro), Centaur [Binz et al., 2025]—a 70B model fine-tuned on 10.7M human decisions across 160+ tasks—and two task-specific fine-tuning methods applied to the same Qwen2.5-1.5B base model: **action-level supervised fine-tuning** (**A-SFT**), which imitates individual human actions, and **process-level fine-tuning** (**P-SFT**), which directly optimizes task-level behavioral process features.

We find that broad behavioral fine-tuning substantially improves human-like process relative to off-the-shelf frontier agents, with Centaur emerging as the strongest general-purpose human-behavior benchmark. When task-specific process representations are known and aligned to the evaluation task, P-SFT can further improve over both Centaur and A-SFT. However, this advantage diminishes under cross-task transfer when process targets do not naturally generalize across tasks. These findings

suggest that explicit process-level supervision can improve human-like behavioral mimicry when the relevant process representation is known, but that specifying transferable process representations remains a central bottleneck for scalable human-process alignment.

Together, our results show that process equivalence is independent of output equivalence, and that reproducing human-like behavioral process depends not only on optimization method but critically on access to task-aligned process representations.

2 Output Equivalence vs. Process Equivalence

To study differences between human and agent task-solving behavior, we constructed CogCaptcha30, a task battery consisting of 29 cognitive tasks together with a canonical CAPTCHA challenge (Figure 1). We begin by describing the shared human-agent evaluation setup, then use CAPTCHA as an intuitive real-world example motivating the broader benchmark.

2.1 Participants and AI Agents

We administered CogCaptcha30 to 100 human participants recruited via Prolific (all participants provided IRB-approved consent; 97 were retained after excluding runs affected by platform issues) and collected 50 full-battery runs from each of three frontier vision-language agents (GPT-5, Gemini 2.5 Pro, and Claude Sonnet 4.5; 150 total agent runs). Agents interacted with the same browser-based interfaces shown to human participants. On each trial, the agent received a screenshot and returned a structured JSON action (e.g., target coordinates, tile index, or categorical choice), which a Playwright-based execution layer translated into the corresponding DOM event (click, keypress, or input). Example task instructions for both agents and participants are provided in Appendix A.3.

2.2 CAPTCHA as a Motivating Real-World Example

We begin with CAPTCHA, a real-world human-machine discrimination task. Modern AI systems can solve many CAPTCHA variants at near-human accuracy [Plesner et al., 2024]. However, matching CAPTCHA performance does not imply matching the behavioral process by which the challenge is solved. To illustrate this, we compare humans and agents on two CAPTCHA variants: a *Classic* format in which participants select target images from a grid, and a *Cross-Tile* format in which participants identify all tiles containing a target object within a single image. Humans and agents achieve statistically indistinguishable performance on these tasks ($t(98) = -0.29$, $p = 0.77$, Figure 1A), indicating matched performance. However, their interaction patterns differ substantially: process-level features such as click direction, side bias, and row/column exploration reliably distinguish human from agent CAPTCHA behavior. Thus, output-matched agents need not be process-matched.

While CAPTCHA provides an intuitive real-world example of performance/process dissociation, it captures only a narrow and domain-specific slice of human behavior. We therefore extend this analysis to the remaining tasks in CogCaptcha30, which span a broad range of cognitive domains and permit systematic measurement of process-level human-agent differences.

2.3 CogCaptcha30: a CAPTCHA-style Cognitive Task Battery

The remainder of CogCaptcha30 consists of 29 cognitive tasks spanning working memory, decision-making, perception, planning and reasoning (Figure 1B; Table 3). Each task is paired with a set of *process features*: run-level summaries such as exploration patterns, sensitivity to outcomes, and trial-to-trial adaptation that capture how behavior unfolds across trials. These features go beyond whether the final answer is correct or how many points are earned. For example, in the Iowa Gambling Task (IGT), participants repeatedly choose among four card decks that differ in reward and risk structure: two decks offer high immediate rewards paired with larger long-term losses, whereas two offer smaller but safer returns [Bechara, 2001]. Task performance is measured by the total amount of money earned, while process features characterize how that outcome was achieved, including win-stay/lose-shift strategy, exploration rate, choice stickiness, etc. [Worthy et al., 2013]. Task descriptions and feature definitions are provided in Appendix Table 3.

Two design principles guided the construction of CogCaptcha30. First, because human-agent discrimination is often required in short-form verification settings such as CAPTCHA-like interactions,

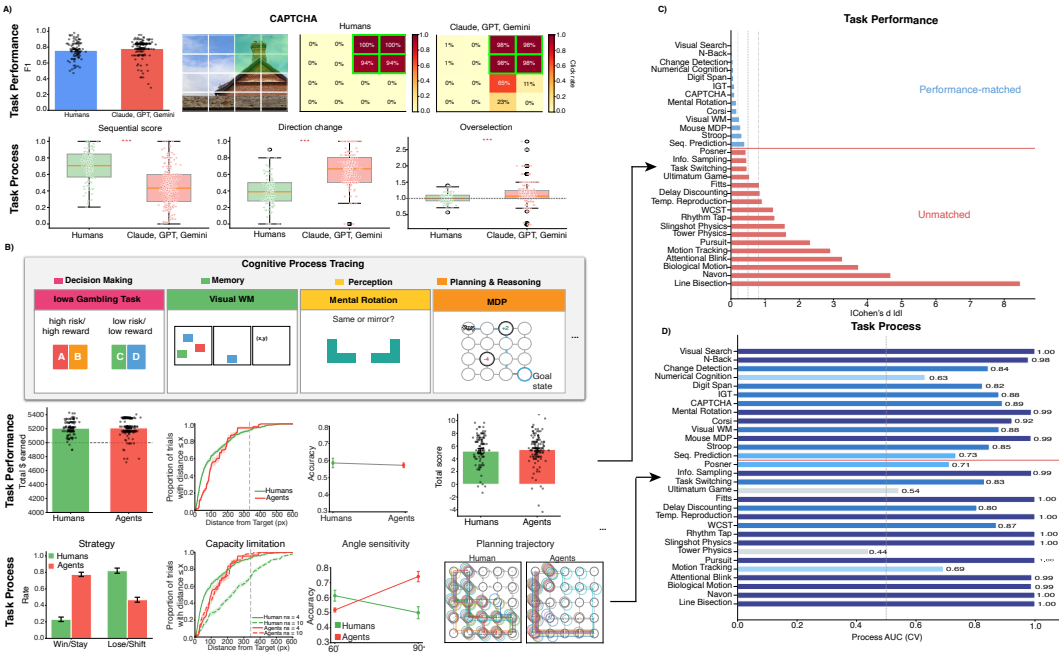

Figure 1: **Process-level behavioral features provide stronger human-agent discriminative signal than task performance alone. A)** CAPTCHA provides a motivating real-world example of output-process dissociation. Humans and frontier agents (Claude Sonnet 4.5, GPT-5, Gemini 2.5 Pro) achieve comparable CAPTCHA performance, yet differ significantly in process-level interaction features including sequential click patterns, direction changes, and overselection behavior. **B)** COGCAPTCHA30 consists of 29 cognitive tasks (in addition to an image CAPTCHA) spanning decision making, memory, perception, and planning/reasoning, each paired with task-specific process features designed to capture how behavior unfolds over time. Representative examples illustrate matched task performance despite divergent process-level signatures across multiple domains. Error bars represent standard errors, individual dots represent individual human or agent participants. **C)** Per-task output distance between humans and agents measured by absolute Cohen's d on each task's primary performance metric (e.g., accuracy, score, reward). Tasks above the red threshold are considered performance-matched. Many tasks show near-human agent performance. **D)** Per-task cross-validated AUC of a Random Forest classifier trained to distinguish humans from agents using process-level features. Despite good performance matching, process-level features remain highly discriminative across most tasks, indicating that output equivalence does not imply process equivalence.

we designed each task to satisfy two comparable practical constraints: (1) at most 10 trials and (2) completion times under one minute. These constraints depart substantially from standard cognitive paradigms, which often require hundreds of trials. Furthermore, all tasks were fully generative: stimuli and conditions were procedurally generated on each task administration, preventing hard-coded strategies that exploit fixed task structure.

All human and agent runs were transformed into a common feature vector comprising a single primary performance output metric (e.g., accuracy, total score), and 129 task-specific process features. These process features were collected over the 30 tasks and extracted using task-specific featurizers that we applied uniformly to humans and agents (Appendix Table 3). Examples of processes and features are shown in Figure 1B.

On output metrics, agents achieved statistical parity with humans on 13 of 30 tasks (Mann–Whitney U, $p \geq 0.05$; median Cohen's $|d| = 0.08$ on matched tasks vs. 1.29 on unmatched tasks; Figure 1C), defining a subset of tasks where output-based discrimination would fail.

Notably, many of the tasks on which agents underperformed humans involved continuous visual stimuli (e.g., biological motion where continuous human figure movement implies direction, and other video-based paradigms). This likely reflects a limitation of current vision-language agents, which typically operate over discrete screen snapshots instead of continuous visual streams, rather than a fundamental inability to perform such tasks. As multimodal systems gain stronger support for continuous visual input, performance gaps on these tasks will likely diminish.

While agents achieved output parity with humans on 13 tasks, process-level features reliably distinguished humans from agents regardless of whether their final outputs were matched. A per-task Random Forest classifier trained to distinguish humans from agents using process features (5-fold stratified cross-validation per task; 200 trees; max depth 5; class-balanced) achieved a mean AUC of 0.88 across output-matched tasks (Figure 1D), comparable to the mean AUC of 0.87 on output-unmatched tasks ($n = 17$). Repeating the same Random Forest analysis using output metrics alone yielded substantially lower discriminative performance (mean AUC = 0.55 on output-matched tasks, mean AUC = 0.80 on output-unmatched tasks, and mean AUC = 0.77 across all tasks). Together, these results indicate that process-level behavioral features provide substantially stronger discriminative signal than output metrics alone, and remain informative even when human and agent task performance converges.

3 Agentic Process Limitations

The preceding results establish that process-level behavioral features reliably distinguish humans from current AI agents even when task performance is matched. A natural next question is what limits an agent's ability to reproduce human-like process. In particular, is the observed process gap primarily due to insufficient exposure to human behavioral data, or does closing it require more direct and task-specific forms of supervision?

To investigate this, we compare two increasingly targeted forms of behavioral adaptation. First, we evaluate **Centaur**, a foundation model trained on large-scale human choice data across diverse cognitive tasks, as a benchmark for broad human-behavior imitation. Second, we study task-specific fine-tuning interventions that provide progressively more direct supervision over the target task, ranging from action-level imitation to explicit process-level alignment.

3.1 Centaur: Large-Scale Human Choice Imitation

Off-the-shelf language models are not explicitly optimized to reproduce human behavior in structured cognitive tasks. By contrast, Centaur [Binz et al., 2025] is a foundation model trained via supervised fine-tuning to predict human decisions using over 10 million choices from more than 160 cognitive experiments. Because Centaur often outperforms domain-specific cognitive models designed to capture human behavior, it provides a strong benchmark for evaluating how closely broad human-choice imitation can approximate human-like process.

Because Centaur operates on text-based prompts rather than visual inputs and requires a structured trial format, we selected three CogCaptcha30 tasks that can be readily converted to text-only form: the Iowa Gambling Task (IGT), Wisconsin Card Sorting Task (WCST), and Information Sampling Task. We evaluated Centaur and frontier agents on these tasks using two complementary metrics:

Distributional distance. For each model and task, we computed Cohen's d between model and human feature distributions for each process feature, and report mean $|d|$ across features together with energy distance between joint distributions [Székely and Rizzo, 2013]; energy results are reported in Appendix Table 4. Lower values indicate closer alignment to human process patterns.

Classifier fool rate. We trained a Random Forest classifier using only features from the three evaluation tasks (10 features total; 200 trees; max depth 5; class-balanced). The classifier was trained to distinguish 97 humans from 120 frontier-agent runs (with 10 held-out runs per frontier model reserved for evaluation), and then applied to held-out Centaur rollouts ($n = 100$). We report each model's *fool rate*: the fraction of runs assigned $P(\text{human}) \geq 0.5$, along with the distribution of classifier-assigned $P(\text{human})$ values.

Centaur more closely matched human process distributions than frontier agents across all evaluated tasks, achieving substantially lower mean distributional distance (Centaur: 0.44; Gemini: 1.07;

GPT: 1.36; Claude: 1.65). Likewise, Centaur achieved substantially higher classifier fool rates than all frontier agents (Centaur fool rate: 79%, remaining models $\leq 20\%$; Centaur $\mathrm{p(Human)}$: 0.67, remaining models $\mathrm{p(Human)} \leq 0.27$).

These results establish Centaur as the strongest broad behavioral benchmark among the models evaluated, showing that large-scale human action imitation improves human-like process relative to off-the-shelf agents. However, the remaining gap between Centaur and human behavior leaves open whether more direct task-specific supervision can further improve process-level alignment.

3.2 Task-Specific Adversarial Mimicry

Human-machine discrimination is inherently an *adversarial* problem. COGCAPTCHA30 demonstrates that process-level behavioral features can distinguish humans from agents even when task outputs are matched, and our comparison to Centaur shows that fine-tuning on large-scale human action data can partially narrow this gap. However, these results do not imply that action imitation represents the upper bound of human-like process mimicry. Any behavioral feature used to distinguish humans from agents may itself become a target for optimization. Accordingly, a detector that succeeds against off-the-shelf agents establishes only a behavioral gap under the current attacker model—not a durable human-verification signal.

This motivates a stronger test: can an agent close the process gap when given increasingly direct access to human behavioral supervision? Rather than asking whether current agents naturally exhibit human-like process, we ask what form of supervision is required for an agent to reproduce it. We test this through two fine-tuning interventions, evaluating at each stage whether (1) the resulting process-feature distributions move closer to the human distribution, and (2) a process-based classifier can still distinguish the modified agent from humans. We consider two forms of fine-tuning:

Task-specific action-level fine-tuning (A-SFT). We fine-tuned an open-source language model (Qwen2.5-1.5B-Instruct) via LoRA [Hu et al., 2021] to predict the human action at each individual decision point. This aligns the model to per-decision human action targets but imposes no explicit objective over run-level behavioral patterns: small per-trial deviations can accumulate across a session, yielding aggregate process features that diverge from the human population. This intervention tests whether *task-specific* action imitation alone is sufficient to reproduce human-like process absent any explicit optimization over aggregate behavioral features.

Process-level fine-tuning (P-SFT). The same model was fine-tuned with an additional loss that explicitly aligned its run-level feature distribution to the human population's.

In P-SFT, we added a second loss term that directly targeted the run-level process feature distribution. At each training step, in addition to the standard cross-entropy loss on individual human actions, we ran the model forward through a complete simulated run of the task and computed its expected process feature vector under its own current policy. The full training objective was:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{diff}} \sum_k w_k \left(\frac{f_k^{\text{model}} - \mu_k^{\text{human}}}{\sigma_k^{\text{human}}} \right)^2, \tag{1}$$

where $\mathcal{L}_{\text{CE}}$ is the standard SFT cross-entropy on human action choices, f_k^{model} is the differentiable estimate of run process feature k under the current policy, $(\mu_k^{\text{human}}, \sigma_k^{\text{human}})$ are the human-population mean and standard deviation of process feature k, and w_k allows up-weighting features that converge more slowly. λ controlled the relative weight of the feature-matching term against the action-level cross-entropy (with λ = 1 meaning equal contribution; hyperparameters are given in Appendix A.2).

P-SFT illustrative example: Iowa Gambling Task. To make the P-SFT mechanism concrete, we illustrate it using the Iowa Gambling Task (IGT), in which participants choose among four card decks (A–D) over 10 trials. Although we use IGT for exposition here, the same procedure was also applied to the Wisconsin Card Sorting Task (WCST) and Information Sampling Task in our broader fine-tuning experiments. The targeted process features are learning slope, stickiness, deck entropy, win-stay rate, lose-shift rate, and good-deck rate. On each trial, the model produces a probability distribution over the four decks — $p(A) = 0.15$, $p(B) = 0.10$, $p(C) = 0.30$, $p(D) = 0.45$. These probabilities feed directly into the feature estimates without waiting to see which deck the model actually picks. For instance, if the model selected deck C on the previous trial, its stickiness estimate for the current trial is simply $p(C) = 0.45$ — the probability of repeating the previous choice. Its

Table 1: Distributional distance to humans measured by mean absolute Cohen's d across process features (lower is more human-like). Centaur more closely matched human process distributions than frontier agents across all evaluated tasks.

Method	Sampling	IGT	WCST	Average
Base Qwen (1.5B)	0.89	0.81	0.88	0.86
Claude Sonnet	1.15	3.32	0.48	1.65
GPT-5	0.27	1.91	1.91	1.36
Gemini 2.5 Pro	1.26	0.56	1.38	1.07
Centaur (70B)	0.15	0.42	0.76	0.44

good-deck-rate estimate is $p(C) + p(D) = 0.75$, since decks C and D are the less risky decks. At the end of the run, per-trial estimates are averaged into run-level features (e.g., mean stickiness across all trials). To construct the prompt for the next trial, one deck is sampled from this distribution and the corresponding outcome is appended to the context (e.g., "You chose deck C and won 50. Balance: 5050."). This sampled action shapes the subsequent prompt but is not differentiated through—gradients flow only through the probability-based feature estimates described above. The mismatch between the resulting run-level features and human means is then penalized via Equation 1.

We fine-tuned both methods using the same 10 process features employed in the Centaur evaluation above. To test whether process-level supervision generalizes beyond the specific behavioral representations used during optimization, we withheld 8 additional process features from the optimization objective and reserved them for evaluation only. These held-out features span distinct behavioral dimensions not directly optimized during training, allowing us to assess whether improvements transfer beyond the supervised process representation. Additionally, we performed a cross-task evaluation in which models were assessed on process features from tasks outside the supervised fine-tuning task. We report held-out-feature and cross-task results separately in the following section. Fine-tuned models were evaluated using the same metrics applied to Centaur and frontier agents: distributional distance and classifier fool rate. In addition, we included the base Qwen2.5-1.5B-Instruct model as a baseline to verify that any observed improvements were not attributable solely to architecture, model family, or parameter scale.

3.3 Results

Distributional distance. Off-the-shelf LLMs remained substantially separated from humans in process-feature space across all evaluated tasks (Table 1). Notably, larger frontier models were not consistently more human-like: Claude Sonnet 4.5 and GPT-5 were among the most distant, while the smallest off-the-shelf model evaluated (base Qwen 1.5B) was the closest on average (Table 1), suggesting that greater capability does not necessarily imply more human-like behavioral process.

When evaluated on the same process features explicitly optimized during training (*observed features*), process-level fine-tuning (P-SFT) achieved the closest match to human behavior, outperforming both task-specific action imitation (A-SFT) and Centaur. This indicates that if the relevant task-level behavioral representation is known and aligned to the evaluation target, explicit process supervision can produce more human-like behavior than large-scale action imitation alone. To assess whether these gains generalize beyond the supervised process representation, we next evaluated on held-out process features from the same tasks that were excluded from optimization (Figure 2A). Under this setting, P-SFT retained a substantial advantage over A-SFT and remained more human-like than Centaur, though the gap narrowed relative to the observed-feature evaluation (Figure 2A).

However, this advantage diminished markedly under *cross-task* evaluation, where models were assessed on tasks whose process representations differed from those used during supervision. In this setting, P-SFT no longer outperformed A-SFT and fell below Centaur, indicating that improvements from explicit process supervision transfer poorly when the target process representation is misaligned with the supervised feature space.

Together, these results suggest that explicit process-level supervision can strongly improve human-like behavioral mimicry when the relevant process representation is known and task-aligned, but that its benefits do not extend for tasks with non-overlapping feature spaces.

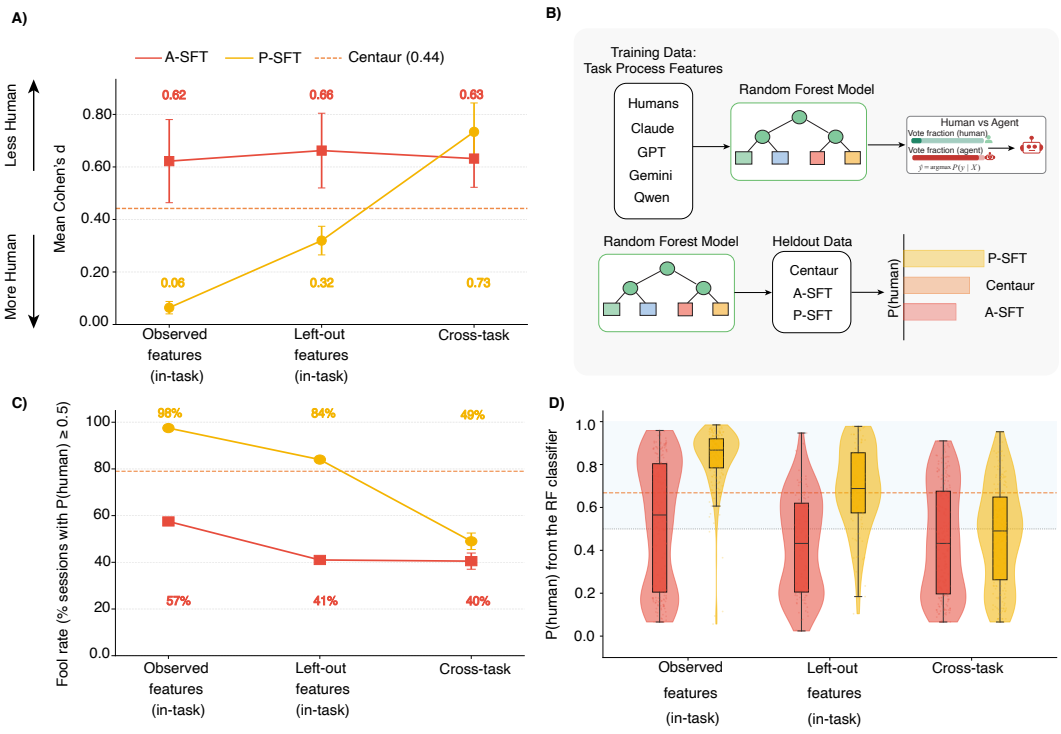

Figure 2: **Task-aligned process supervision improves human-like behavior in-task, but its advantage diminishes under transfer. A)** Mean absolute Cohen's d between model and human process-feature distributions for action-level fine-tuning (A-SFT), process-level fine-tuning (P-SFT), and Centaur (dashed reference line). P-SFT achieves the closest match to humans when evaluated on the process features explicitly optimized during training (*observed features*) and remains superior on held-out features from the same tasks, but this advantage disappears under cross-task evaluation on tasks with distinct process representations. **B)** Evaluation protocol for classifier-based behavioral indistinguishability. A Random Forest classifier is trained to distinguish humans from off-the-shelf agents using task process features, then applied to held-out rollouts from Centaur, A-SFT, and P-SFT to estimate $P(\text{human})$. **C)** Classifier fool rate (percentage of sessions with $P(human) \geq 0.5$). P-SFT substantially outperforms A-SFT in the observed-feature and held-out-feature settings, but this advantage declines sharply in the cross-task setting. **D)** Distribution of classifier-assigned $P(\text{human})$ values across evaluation settings. Process-level supervision yields the most human-like behavior when the target process representation is aligned with the evaluation task, but provides limited benefit when transferring to tasks outside the supervised process space.

Classifier fool rate. Results from the held-out Random Forest classifier mirrored the distributional-distance analysis (Table 2; Figure 2C–D). Off-the-shelf frontier agents were reliably detected, with fool rates near zero. Task-specific action imitation (A-SFT) increased fool rate, indicating that matching individual human actions moves models toward the human region of feature space.

When evaluated on observed process features, P-SFT achieved the highest fool rate of all methods, outperforming both A-SFT and Centaur and approaching behavioral indistinguishability from humans under the held-out classifier. This advantage persisted, though attenuated, when evaluation was restricted to held-out features that weren't used for P-SFT optimization from the same tasks. However, under cross-task evaluation, P-SFT's fool rate dropped sharply and approached that of A-SFT, indicating limited transfer of process-level supervision beyond the behavioral feature space directly optimized during training.

These findings are in line with distributional distance, and support the conclusion that process-level supervision yields its largest gains when the target process representation is accessible, but does not easily scale to tasks with different process features.

Table 2: Combined classifier fool rate. A Random Forest classifier (AUC = 0.970 $\pm$ 0.024) was trained to distinguish humans from four off-the-shelf LLM agents using process features concatenated across all three tasks. Test set off-the-shelf agents, fine-tuned models and Centaur were held out from training and evaluated against the fitted classifier. Higher fool rate indicates more human-like behavior.

Method	Fool rate ↑	Mean $P(\text{human})$
In classifier training		
GPT-5	4.0%	0.11
Gemini 2.5 Pro	0.0%	0.15
Claude Sonnet	0.0%	0.02
Base Qwen (1.5B)	20.0%	0.27
Centaur (70B)	79.0%	0.67

4 Discussion

Across CogCaptcha30, process-level behavioral features provided substantially stronger discriminative signal than output metrics alone, reliably distinguishing humans from agents even when task performance was matched. On a subset of structured decision-making tasks, we further compared three increasingly direct forms of behavioral supervision. Task-specific action imitation (A-SFT) reduced the process gap but did not eliminate it, consistent with the interpretation that matching local action choices does not guarantee matching aggregate behavioral dynamics when small per-trial deviations compound across a session. Centaur provided the strongest general-purpose benchmark for human-like process, yet still failed to fully close the gap. Explicit process-level supervision (P-SFT) produced the closest match to human behavior when the relevant task-specific process representation was available, but its advantage disappeared when evaluated outside the supervised process space.

These findings suggest that broader human-process alignment may require methods capable of learning more transferable or abstract behavioral representations than task-specific feature matching alone. One possibility is that scalable process alignment will require richer forms of supervision, such as latent behavioral embeddings, hierarchical behavioral abstractions, or multi-task objectives that capture shared structure across tasks.

Our experiments suggest that achieving robust human-like process alignment is challenging even under favorable task-specific supervision. Moreover, even if future methods substantially improve process mimicry, robust process-level discrimination need not rely on a fixed battery of behavioral probes. Automated task generation or iterative task redesign could continuously introduce novel behavioral demands and target process dimensions where alignment remains imperfect, creating a dynamic red-teaming loop in which adversaries must repeatedly adapt to newly introduced process-level challenges.

Limitations. Several limitations remain. First, our evaluation is restricted to a 1.5B-parameter model and a limited subset of structured sequential decision-making tasks with discrete, low-cardinality action spaces. Accordingly, it remains unclear whether similar process-alignment dynamics will hold for larger models or richer interactive environments. In addition, our human data were drawn from a relatively small participant sample from a single pool. We therefore operationalize "human-like" process relative to the empirical behavioral distribution of this sampled population under the chosen feature representation, rather than as a claim about any canonical form of human cognition. We hope that these limitations can be addressed in future work.

References

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. Taxonomy of Risks posed by Language Models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229, 2022.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing*, 4(2):100211, 2024.

Peter S Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. AI deception: A survey of examples, risks, and potential solutions. *Patterns*, 5(5), 2024.

Alan M. Turing. Computing Machinery and Intelligence. *Mind*, 59(236):433–460, 1950.

Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. CAPTCHA: Using hard AI problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–311. Springer, 2003.

Cameron R Jones and Benjamin K Bergen. Large Language Models Pass the Turing Test. *arXiv preprint arXiv:2503.23674*, 2025.

Andreas Plesner, Tobias Vontobel, and Roger Wattenhofer. Breaking reCAPTCHAv2. In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1047–1056. IEEE, 2024.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134 (1-2):57–83, 2002.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, 2019.

Ned Block. Psychologism and Behaviorism. *The Philosophical Review*, 90(1):5–43, 1981.

Kenneth Wingate Regan, Tamal Biswas, and Jason Zhou. Human and Computer Preferences at Chess. In *MPREF@AAAI*, 2014.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114, 2001.

Anne GE Collins and Michael J Frank. How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, 35(7):1024–1035, 2012.

Patrick M. A. Rabbitt. Errors and error correction in choice-response tasks. *Journal of Experimental Psychology*, 71(2):264–272, 1966.

Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364(6432):56–58, 1993.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K Eckstein, Noémi Éltető, et al. A foundation model to predict and capture human cognition. *Nature*, 644(8078):1002–1009, 2025.

Antoine Bechara. Neurobiology of decision-making: risk and reward. In *Seminars in Clinical Neuropsychiatry*, volume 6, pages 205–216, 2001.

Darrell A Worthy, Bo Pang, and Kaileigh A Byrne. Decomposing the roles of perseveration and expected value representation in models of the Iowa gambling task. *Frontiers in Psychology*, 4: 640, 2013.

Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, Weizhu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*, 2021.

Kimron L Shapiro, Jane E Raymond, and Karen M Arnell. The attentional blink. *Trends in Cognitive Sciences*, 1(8):291–296, 1997.

Nikolaus F Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of Vision*, 2(5):2–2, 2002.

Steven J Luck and Edward K Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, 390(6657):279–281, 1997.

Philip Michael Corsi. Human Memory and the Medial Temporal Region of the Brain. 1972.

James E Mazur. An Adjusting Procedure for Studying Delayed Reinforcement. In *The Effect of Delay and of Intervening Events on Reinforcement Value*, pages 55–73. Psychology Press, 2013.

David Wechsler. Wechsler Adult Intelligence Scale. *Archives of Clinical Neuropsychology*, 1955.

Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381, 1954.

George Jewell and Mark E McCourt. Pseudoneglect: a review and meta-analysis of performance factors in line bisection tasks. *Neuropsychologia*, 38(1):93–110, 2000.

Roger N Shepard and Jacqueline Metzler. Mental Rotation of Three-Dimensional Objects. *Science*, 171(3972):701–703, 1971.

Zenon W Pylyshyn and Ron W Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, 3(3):179–197, 1988.

Frederick Callaway, Falk Lieder, Paul M Krueger, and Thomas L Griffiths. Mouselab-MDP: A new paradigm for tracing how people plan. In *The 3rd Multidisciplinary Conference on Reinforcement Learning and Decision Making, Ann Arbor, MI*, 2017.

David Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9(3):353–383, 1977.

Adrian M Owen, Kathryn M McMillan, Angela R Laird, and Ed Bullmore. N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies. *Human Brain Mapping*, 25(1):46–59, 2005.

Samuel J Cheyette and Steven T Piantadosi. A unified account of numerosity perception. *Nature Human Behaviour*, 4(12):1265–1272, 2020.

Michael I Posner. Orienting of attention: Then and now. *The Quarterly Journal of Experimental Psychology*, 69(10):1864–1875, 2016.

Stefan Künzell, Dominicus Sießmeir, and Harald Ewolds. Validation of the Continuous Tracking Paradigm for Studying Implicit Motor Learning. *Experimental Psychology*, 2017.

Bruno H Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):969–992, 2005.

Luke Clark, Trevor W Robbins, Karen D Ersche, and Barbara J Sahakian. Reflection Impulsivity in Current and Former Substance Users. *Biological Psychiatry*, 60(5):515–522, 2006.

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

Julie M Bugg, Larry L Jacoby, and Jeffrey P Toth. Multiple levels of control in the Stroop task. *Memory & Cognition*, 36(8):1484–1494, 2008.

Nicholas P Maxwell, Mark J Huff, and Jacob M Namias. Predictive alternating runs and random task-switching sequences produce dissociative switch costs in the Consonant–Vowel/Odd–Even task. *Cognitive Processing*, 26(1):157–170, 2025.

Mehrdad Jazayeri and Michael N Shadlen. Temporal context calibrates interval timing. *Nature Neuroscience*, 13(8):1020–1026, 2010.

Martin A Nowak, Karen M Page, and Karl Sigmund. Fairness Versus Reason in the Ultimatum Game. *Science*, 289(5485):1773–1775, 2000.

Miguel P Eckstein. Visual search: A retrospective. *Journal of Vision*, 11(5):14–14, 2011.

Jason Rajsic and Daryl E Wilson. Asymmetrical access to color and location in visual working memory. *Attention, Perception, & Psychophysics*, 76(7):1902–1913, 2014.

Stanislas Dehaene and Jean-Pierre Changeux. The Wisconsin Card Sorting Test: Theoretical Analysis and Modeling in a Neuronal Network. *Cerebral Cortex*, 1(1):62–79, 1991.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024.

A Appendix

A.1 CogCaptcha30 Catalog

Table 3: Full CogCaptcha30 process-feature catalog with task descriptions. CAPTCHA image task is split into two rows for Classic and Cross-Tile variants.

Task	Task description	Task source	Task-specific features
Attentional blink	Rapid serial visual presentation; detect two targets with varying temporal lag	Shapiro et al. [1997]	`blink_magnitude`, `t2_short_lag_acc`, `t2_long_lag_acc`, `t2_given_t1_acc`, `conditional_blink`
Biological motion	Point-light walker animation; judge walking direction	Troje [2002]	`direction_bias`, `acc_left`, `acc_right`, `acc_variability`
Classic CAPTCHA	Select all tiles matching a target object	Von Ahn et al. [2003]	`overselection_ratio`, `sequential_score`, `direction_change_rate`
Cross-Tile	Objects span tile boundaries, requiring spatial grouping	Von Ahn et al. [2003]	`overselection_ratio`, `sequential_score`, `direction_change_rate`
Change detection	Detect changes between alternating images	Luck and Vogel [1997]	`mean_flicker_cycles`, `flicker_cv`

Task	Task Description	Task Source	Task-Specific Features
Corsi	Spatial span task; reproduce sequence of block locations	Corsi [1972]	`acc_slope_span`, `transposition_rate`, `intrusion_rate`, `serial_first_acc`, `serial_middle_acc`, `serial_last_acc`, `primacy_recency`
Delay discounting	Choose between smaller immediate vs larger delayed rewards	Mazur [2013]	`discount_slope`, `choice_consistency`, `log_discount_k`
Digit span	Verbal working memory; recall digit sequences in order	Wechsler [1955]	`acc_slope_span`, `transposition_rate`, `intrusion_rate`, `serial_first_acc`, `serial_middle_acc`, `serial_last_acc`, `primacy_recency`
Fitts	Pointing task; click targets of varying size and distance	Fitts [1954]	`fitts_slope`, `fitts_intercept`, `click_precision_mean`, `click_precision_std`, `click_distance_px_mean`, `click_distance_px_std`
Iowa Gambling Task	Sequential card choices from decks with different reward-risk profiles	Bechara [2001]	`early_exploration`, `learning_slope`, `win_stay`, `lose_shift`, `stickiness`, `deck_entropy`
Line bisection	Mark midpoint of a line (spatial bias)	Jewell and McCourt [2000]	`mean_abs_error`, `error_slope_length`
Mental rotation	Judge whether rotated shapes are same or mirrored	Shepard and Metzler [1971]	`rotation_slope`, `angle_acc_delta`, `acc_angle_small`, `acc_angle_large`, `mirror_acc_effect`
MOT	Track moving targets among distractors	Pylyshyn and Storm [1988]	`load_slope`
Mouselab MDP	Explore states via mouse movements to maximize reward	Callaway et al. [2017]	`edge_fraction`, `mean_dist_to_edge`, `turn_rate`, `edge_hovers_mean`, `state_hovers_mean`, `unique_edges_explored`, `exploration_ratio`, `pre_move_hovers`
Navon	Global-local letters; identify large vs small letter under congruency manipulations	Navon [1977]	`navon_congruency_effect`, `global_advantage`, `navon_global_cong_acc`, `navon_global_incong_acc`, `navon_local_cong_acc`, `navon_local_incong_acc`, `navon_global_congruency_acc`, `navon_local_congruency_acc`, `global_acc_advantage`
N-back	Working memory task; detect repeats from N trials back	Owen et al. [2005]	`criterion`, `hit_rate`, `fa_rate`
Numerosity	Estimate which of two sets has more items	Cheyette and Piantadosi [2020]	`difficulty_effect`, `response_bias`, `distance_acc_slope`
Posner	Cueing task; respond to targets with valid/invalid spatial cues	Posner [2016]	`validity_acc_effect`, `valid_acc`, `invalid_acc`

Task	Task Description	Task Source	Task-Specific Features
Target pursuit	Track moving target with cursor; continuous control task	Künzell et al. [2017]	cursor_smoothness, directional_accuracy, correction_rate, anticipation_lag_ms
Rhythm tapping	Tap in time with rhythm; measures timing variability and drift	Repp [2005]	tap_error_mean, tap_error_std, tap_drift, interval_cv
Sampling	Sequential information sampling before making a decision	Clark et al. [2006]	mean_total_samples, var_total_samples, mean_sample_bias, samples_easy, samples_medium, samples_hard, neutral_first_rate, effort_accuracy_corr
Sequence prediction	Predict next element in a sequence with varying complexity	[N/A] source	difficulty_slope, seq_acc_d1, seq_acc_d2, seq_acc_d3, seq_acc_d4, seq_acc_d5
Slingshot	Aim and launch projectile to hit targets; physics-based control task	Battaglia et al. [2013]	mean_aim_time, aim_time_cv, distance_calibration_r, mean_angle_delta_45, angle_variability, undershoot_rate, overshoot_rate
Stroop	Name ink color of words; congruent vs incongruent trials	Bugg et al. [2008]	post_error_slowing, congruency_acc_effect, cong_acc, incong_acc
Task switching	Alternate between task rules; measures switching cost	Maxwell et al. [2025]	switch_cost_acc, repeat_acc, switch_acc
Temporal reproduction	Reproduce time intervals; measures temporal bias and variability	Jazayeri and Shadlen [2010]	temporal_bias, mean_ratio, regression_to_mean
Tower physics	Predict stability of stacked objects (intuitive physics)	Battaglia et al. [2013]	complexity_slope
Ultimatum game	Fairness decision-making	Nowak et al. [2000]	responder_accept_rate, reject_unfair_rate
Visual search	Find target among distractors	Eckstein [2011]	search_slope, search_click_precision, click_dist_setsize_slope
Visual Working Memory	Recall location/color under varying set sizes	Rajsic and Wilson [2014]	error_slope_setsize, bias_x_mean, bias_y_mean, bias_radius_mean, bias_anisotropy, large_error_frac, swap_error_rate
WCST	Infer and update sorting rule from feedback	Dehaene and Changeux [1991]	learning_slope, post_error_slowing, shift_error_rate, perseveration_cost, rule_acc_variability

A.2 P-SFT: implementation details

All fine-tuning was performed on a single NVIDIA H200 SXM GPU (141 GB VRAM). We fine-tuned Qwen2.5-1.5B-Instruct [Team, 2024] via LoRA [Hu et al., 2021] (rank 8, alpha 16, dropout 0.05) applied to all attention and MLP projection layers. Training used AdamW with learning rate 10^{-5}, gradient clipping at norm 1, and gradient checkpointing to fit the per-rollout state expansion in memory. A-SFT trained for 300 steps of cross-entropy on human actions. P-SFT added 30 steps of cross-entropy warmup followed by up to 150 steps of joint optimization, with early stopping when

Table 4: Energy distance—a multivariate distributional metric defined as $2\,\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|$, computed on per-feature human-standardized features—captures both location and shape differences between a method's and humans' joint feature distributions; a value of zero indicates identical distributions.

Method	Sampling	IGT	WCST	Average
Base Qwen (1.5B)	1.50	1.27	1.04	1.27
Claude Sonnet	1.33	11.14	0.61	4.36
GPT-5	0.29	5.84	2.28	2.81
Gemini 2.5 Pro	2.06	1.30	1.42	1.59
Centaur (70B)	0.03	0.52	0.56	0.37
A-SFT (observed features)	0.55	0.53	0.97	0.68
P-SFT (observed features)	**0.11**	**0.06**	**0.06**	**0.07**
Centaur (left-out features)	0.15	0.29	0.16	0.20
A-SFT (left-out features)	1.24	0.26	0.60	0.70
P-SFT (left-out features)	**0.41**	**0.33**	**0.04**	**0.26**
A-SFT (cross-task)	1.05	0.39	0.83	0.76
P-SFT (cross-task)	**1.28**	**0.44**	**0.59**	**0.77**

the feature-matching loss plateaued (patience of 20–50 steps depending on task). At each joint step, we sampled a fresh stimulus sequence from a random training participant's run, preventing the model from overfitting to a single sequence of task stimuli.

Evaluation. We use 2-fold cross-validation: 97 human participants are split into two folds (47 and 50). Each fold's model is trained on one half and evaluated against held-out humans from the other half. Per-fold rollouts (100 per fold, 200 total) are compared to per-fold held-out humans, and results are pooled. This ensures no overlap between training and evaluation participants.

Information Sampling. This task involves multiple decisions per trial: the agent chooses to reveal a tile from option A, reveal a tile from option B, or stop and choose an option. The run-level features are mean and variance of the per-trial total tile count. Because each trial involves a variable-length sequence of decisions, we compute the expected tile count differentiably by expanding the set of reachable game states at each within-trial step, weighted by the policy's action probabilities. States with identical observable histories (e.g., two A-tiles revealed via different orderings) are merged. The expected stopping time is then

$$\mathbb{E}[T] = \sum_{t=0}^{T_{\max}} t \cdot P(\text{stop at step } t), \tag{2}$$

computed exactly by accumulating $P(\text{stop} \mid \text{state}_t) \cdot P(\text{state}_t)$ over the expansion. We cap the frontier at 16 active states per step, which exceeds the theoretical maximum of 11 distinct observable states after merging (since with 5 tiles per option, the merged state space at step t is the set of (n_A,n_B) pairs with $n_A + n_B = t$. At trial boundaries, we select the most-likely outcome from the frontier (greedy stop time and choice) to construct the prompt continuation for subsequent trials, matching the prompt format used at evaluation time.

Iowa Gambling Task. Each trial is a single 4-way choice over decks A, B, C, D; no within-trial state expansion is needed. The six run features, all differentiable in the model's action probabilities p_t, are:

- *learning_slope*: mean good-deck probability in the second half minus the first half, $\frac{1}{\lfloor T/2 \rfloor} \sum_{t \geq \lceil T/2 \rceil} p_t^{\text{good}} - \frac{1}{\lceil T/2 \rceil} \sum_{t < \lceil T/2 \rceil} p_t^{\text{good}}$, where $p_t^{\text{good}} = p_t(C) + p_t(D)$
- *stickiness*: $\frac{1}{T-1} \sum_{t=2}^{T} p_t(a_{t-1})$, the mean probability assigned to the previous trial's sampled action (the sampled action a_{t-1} is treated as fixed; gradients flow through p_t only)
- *deck_entropy*: $H(\bar{p}) = -\sum_d \bar{p}_d \log \bar{p}_d$, where $\bar{p} = \frac{1}{T} \sum_t p_t$ is the mean action distribution across trials

- *win_stay*: mean probability of repeating the previous action on trials following a positive outcome, $\frac{1}{|\mathcal{W}|}\sum_{t\in\mathcal{W}} p_t(a_{t-1})$, where $\mathcal{W} = \{t : \text{net}_{t-1} > 0\}$
- *lose_shift*: mean probability of switching away from the previous action on trials following a non-positive outcome, $\frac{1}{|\mathcal{L}|}\sum_{t\in\mathcal{L}}(1 - p_t(a_{t-1}))$, where $\mathcal{L} = \{t : \text{net}_{t-1} \leq 0\}$
- *good_deck_rate*: $\frac{1}{T}\sum_t (p_t(C) + p_t(D))$

Win/loss classification at each trial is determined by the sampled action and the deterministic deck outcomes, and is treated as fixed for gradient computation. Gradients flow only through the probability vectors p_t.

Wisconsin Card Sorting Task. Each trial is a single 4-way choice (match to reference card 1, 2, 3, or 4). Whether the choice is correct depends on the current hidden rule and the test card. The two run features are:

- *perseveration_cost*: mean $p(\text{correct})$ on non-shift trials minus mean $p(\text{correct})$ on shift trials, where $p(\text{correct}) = p_t(\text{correct index})$ is the probability the model assigns to the correct card on trial t
- *learning_slope*: mean $p(\text{correct})$ in the second half of the run minus the first half

Both are direct functions of the model's per-trial softmax probabilities evaluated at the correct action index, which is known from the stimulus sequence.

Training stability. Under the closed-form feature estimator, the feature-matching loss decreases monotonically during phase-2 training across all three tasks, typically reaching a plateau around step 90-100. We use AdamW with learning rate 10^{-5}, gradient clipping at norm 1, and gradient checkpointing to fit the per-rollout computation in memory.

A.3 Task Instructions and Agent Prompts

A.3.1 Information Sampling

Agent Prompt (Sampling Phase)

There are two rows of tiles:
- Option A (top/blue): 5 tiles
- Option B (bottom/red): 5 tiles

Hidden tiles show "?". Revealed tiles show numbers (0–100). Your goal is to determine which option has the higher average.
Decide whether to sample another tile or stop and choose.
If sampling, respond with: `{"action": "sample", "option": "A", "tile": <0-4>}`
or
`{"action": "sample", "option": "B", "tile": <0-4>}`
If ready to choose: `{"action": "choose"}`

Agent Prompt (Choice Phase)

Based on the revealed tile values, which option has the higher average?
Respond with ONLY:
`{"choice": "A"}` or `{"choice": "B"}`

Participant Instructions

You will see two rows of 5 hidden tiles: Option A (top/blue) and Option B (bottom/red).
Each tile hides a number. One option tends to have higher values than the other.
Click a tile to reveal its value. Each flip costs points.
Choosing correctly earns bonus points.
When you are confident, click "I'm ready to choose."
Your goal is to determine which option has higher values overall while minimizing flip costs.

A.3.2 Visual Working Memory

Agent Prompt

Image 1 shows colored squares during encoding. Memorize each color's position.
Image 2 shows an empty canvas and a target color labeled "Click where this color was."
Find where the target color was located and report its center coordinates.
Respond with ONLY:
`{"x": <number>, "y": <number>}`

Participant Instructions

You will see several colored squares flash on the screen.
After they disappear, one of the colors will appear as the target.
Click where on the screen that color was previously shown.

A.3.3 Visual Search

Agent Prompt

Find the RED CIRCLE and click on it.
It will be hidden among red squares and blue circles.
Respond with ONLY:
`{"x": <number>, "y": <number>}`

Participant Instructions

Find the RED CIRCLE and click on it.
It will be hidden among red squares and blue circles.

A.3.4 Multiple Object Tracking

Agent Prompt (Highlight Phase)

Some dots are highlighted as targets.
Report the coordinates of each highlighted target.
Respond with ONLY:
`{"targets": [{"x": N, "y": N}, ...]}`

Agent Prompt (Tracking Phase)

Track the previously highlighted dots as they move.
Update each target's position to the nearest dot.
Respond with ONLY:
`{"targets": [{"x": N, "y": N}, ...]}`

Participant Instructions

Some dots will flash yellow—these are your targets.
All dots will then begin moving. Track the targets.
When the dots stop, click the dots you believe were the targets.

A.3.5 Delay Discounting

Agent Prompt

Look at the two monetary options on screen: A smaller amount now or A larger amount later
Determine the amounts and delay, reason about the tradeoff, and report your choice.
Respond as JSON:
`{"now_amount": N, "later_amount": N, "delay": "X", "choice": "now" or "later"}`

Participant Instructions

On each trial you will choose between:

- A smaller amount now
- A larger amount later

There are no right or wrong answers—choose whichever you prefer.