LEARNING A BEHAVIORAL SKILL BASIS FOR AI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Populations of AI agents are usually summarized either by aggregate benchmark scores, which say nothing about an individual task, or by latent embeddings whose coordinates carry no fixed meaning. Neither gives a reusable representation of what a task requires and what an agent can do. We learn a compact *behavioral skill basis* directly from item-level outcomes, using binary success records and frozen task embeddings alone, with no skill taxonomy, no rationales, and no per-task parameters. Each task is a sparse nonnegative combination of demands on the basis, each agent carries abilities on the same coordinates, and the predicted log-odds of success is the agent's global ability plus a demand-weighted gap between ability and difficulty on each demanded skill. We then ask whether the coordinates behave as measurements, first showing that a task set determines only the profile differences its demands vary, so a task set that covers every skill equally can still leave a relative skill difference unmeasurable. Passing every prediction through 8 to 32 shared coordinates makes it exactly decomposable by skill and costs at most 0.3 accuracy points against dense amortized item response theory on EmbedLLM and RouterBench. On a controlled pool of 16 agents, the learned profiles align strongly with withheld metadata about domain-specific training differences, which were never used during learning, reaching a selectivity AUC of 0.88 ± 0.06 over six retrainings against a shuffled null of 0.50. An unseen agent is placed in the frozen basis by fitting one number per skill, reaching 73.3% accuracy from 341 responses with all-mpnet-base-v2, against 73.7% when it is included in training. Refitting an agent before and after further training on a known mixture moves its profile toward that mixture in 29 of 30 adaptations. Agent populations therefore contain reusable low-dimensional behavioral structure that can be learned from outcomes and used to measure how agents differ and how they change.

1 INTRODUCTION

AI systems are increasingly deployed as populations of agents with different strengths. In a large agent ecosystem, it is useful to have a common representation of what tasks require and what agents can do. Such a representation could support explaining why one agent is preferred on a task, placing a new agent in an existing system, and measuring how an agent changes after retraining.

Existing summaries do not provide this object. Aggregate benchmark scores lose task-level differences (Burnell et al., 2023b). Rubric-based general scales (Zhou et al., 2026b) provide demand and ability profiles, but on dimensions fixed and annotated in advance. Dense latent models (Reckase, 2009; Zhuang et al., 2024; Chen et al., 2025; Yao et al., 2026) predict task-level outcomes, but their coordinates need not have stable meaning: an invertible change of coordinates can transform task and agent representations together without changing any prediction, so good prediction alone does not give a reusable representation of skill.

We ask whether such a representation can instead be learned directly from behavior. Given item-level successes and failures and frozen task embeddings, our method, *SkillBasis*, learns a small set of shared behavioral coordinates. Each task places sparse nonnegative demands on these coordinates, each agent has abilities on the same coordinates, and success depends on the demand-weighted gaps between ability and difficulty. The basis is learned without a skill taxonomy, rationales, or per-task fitted parameters. Sparse nonnegative demands on shared unit-norm directions that must also

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.

reconstruct the task embedding rule out such invertible changes: apart from relabeling the skills, the only exact freedom left in a fitted basis is adding the same vector to every profile and to the difficulty thresholds, so profiles are compared only with each other (Section 4).

Using the coordinates as measurements means reading an agent's position from its responses to some set of tasks, and two questions follow. First, what can a given task set reveal? Section 4 shows that a task set that covers every skill equally can still leave a relative skill difference invisible, and more responses to the same task combinations cannot recover it. Second, do the readings track how agents actually differ and change (Section 5)? Our contributions are:

- We introduce a behavioral skill basis learned from item-level outcomes, in which sparse task demands and dense agent abilities share the same learned coordinates, together with a compact estimator for placing a new agent in a frozen basis (Section 3).
- We analyze what a profile fit can recover from a given task set. After global ability is removed, only differences that the task demands vary are determined, so balanced skill coverage can leave a relative skill contrast unmeasurable (Section 4).
- We test four properties of the learned coordinates: *predictive sufficiency* (near dense predictors), *alignment with known specialization*, *efficient measurement* (unseen agents from few responses), and *intervention consistency* (29 of 30 controlled adaptations move toward the retraining data; Section 5).

Scope. SkillBasis is a representation, not a bid for state-of-the-art outcome prediction: it gives up a small amount of accuracy relative to the best dense predictors in exchange for shared, sparse, additive coordinates that decompose every prediction exactly by skill (Rudin, 2019). A skill is a behavioral construct, a direction in a frozen task-embedding space whose demand predicts differences in success, not a claim about mechanisms inside an agent. Skill names are LLM-generated for readability, and none of the behavioral tests uses them. Our agents answer single-turn prompts, and our controlled interventions change subject-domain training data; tool-using and long-horizon agents (Liu et al., 2024; Jimenez et al., 2024) and other kinds of intervention are left to future work.

2 Related work

Latent skills from aggregate performance. Sloth (Maia Polo et al., 2025), capability factors of benchmark scores (Burnell et al., 2023a; Ruan et al., 2024), and the low-rank analysis of Maimon et al. (2025) infer latent skills from model-by-benchmark performance, and the latter uses the resulting space to profile new models and select a model by its skill profile, while Taskonomy (Zamir et al., 2018) and Task2Vec (Achille et al., 2019) embed tasks to predict transfer. The unit of observation in these analyses is a model paired with a whole benchmark, so two tasks from one benchmark receive the same skill description; our unit is a single agent-task outcome.

Instance-level skills from rationales or rubrics. Moayeri et al. (2025) parse model-generated rationales to recover the skills relevant to each evaluation instance and group instances into skill slices, a discrete assignment of instances to named skills. Zhou et al. (2026b) are closest to our aim: they define general scales with human-designed rubrics, have an LLM annotate each instance's demand level on every scale, and place model abilities on the same scales to explain and predict performance. In these works, the skill vocabulary comes from rationales or rubrics, not from outcomes. Our coordinates are learned from cross-agent outcomes and task embeddings, with no rationale, rubric, or per-instance annotation.

Amortized IRT and cognitive diagnosis. IRT models a response through latent ability and item parameters (Rasch, 1960) and has been used to audit NLP and LLM benchmarks, evaluate models from few items, and predict model outcomes (Rodriguez et al., 2021; Maia Polo et al., 2024; Kipnis et al., 2025; Truong et al., 2025; Hofmann et al., 2025; Zhou et al., 2026a; Liu et al., 2025). Embedding-conditioned variants extend it to unseen tasks, including IrtNet (Chen et al., 2025), JE-IRT (Yao et al., 2026), EmbedLLM matrix factorization (Zhuang et al., 2024), and IRT-Router (Song et al., 2025), and JE-IRT in particular learns a shared dense geometry for queries and models and can add a new model from its responses. Our response model has the same predictive form but represents each task's demands as a sparse combination of a few shared skill directions rather than as a

free point in a dense latent space. LLM routers (Chen et al., 2024; Ong et al., 2025; Song et al., 2025; Hu et al., 2024) learn which agent to call for each query and are optimized for selection accuracy and cost; our goal is a shared interpretable representation, which can explain routing decisions but is not optimized for them. In cognitive diagnosis a Q-matrix links items to skills; NeuralCD extends the response function and uses exercise text to reduce reliance on a hand-specified Q-matrix (Wang et al., 2020), and recent work applies an embedding-informed Q-matrix to LLM evaluation while jointly estimating mastery profiles and item attributes (Liu et al., 2026). We share the goal but learn continuous skill directions inside the task representation, so demands are continuous mixtures rather than an item-attribute assignment and abilities are defined on the same directions by projection.

Sparse dictionaries and concept bottlenecks. We build on standard sparse-dictionary components, while in our approach the same learned directions carry task demands and agent abilities and are shaped by agent outcomes. Sparse dictionaries (Olshausen & Field, 1996) underlie sparse autoencoders (Bricken et al., 2023; Gao et al., 2024; Bussmann et al., 2024), including for text embeddings (O'Neill et al., 2024), and can be optimized jointly with a supervised objective (Mairal et al., 2012) or reserve a partition for ontology concepts (Yang et al., 2026). Concept bottleneck models pass prediction through nameable variables supplied in advance (Koh et al., 2020; Oikarinen et al., 2023). Learned concepts can carry unintended information (Mahinpei et al., 2021), text dictionaries can recover topic structure (Girrbach & Akata, 2026), and dictionaries can be checked against known features (Karvonen et al., 2024; Makelov et al., 2025), which motivates the controls in Section 5.3.

3 THE BEHAVIORAL SKILL BASIS

Setting. We observe triples $\mathcal{D} = \{(M_i, T_j, y_{i,j})\}$ over tasks $\mathcal{T}$ and agents $\mathcal{M}$, with $y_{i,j} \in \{0, 1\}$ recording whether agent i answered task j correctly, and a frozen encoder ϕ maps each task to $\mathbf{x}_j = \phi(T_j) \in \mathbb{R}^d$. No skill taxonomy, task label, or item-to-skill assignment is provided. We use *agent* for any system whose interaction with a task produces a scored outcome: language models answering single-turn prompts in Section 5, and small MLPs in a procedural environment in Section 4.

Skills as shared directions. We represent each embedding with $K = C + R$ unit-norm directions, split into C skill directions $\{\hat{\mathbf{a}}_k\}$ and R auxiliary reconstruction directions $\{\hat{\mathbf{r}}_p\}$,

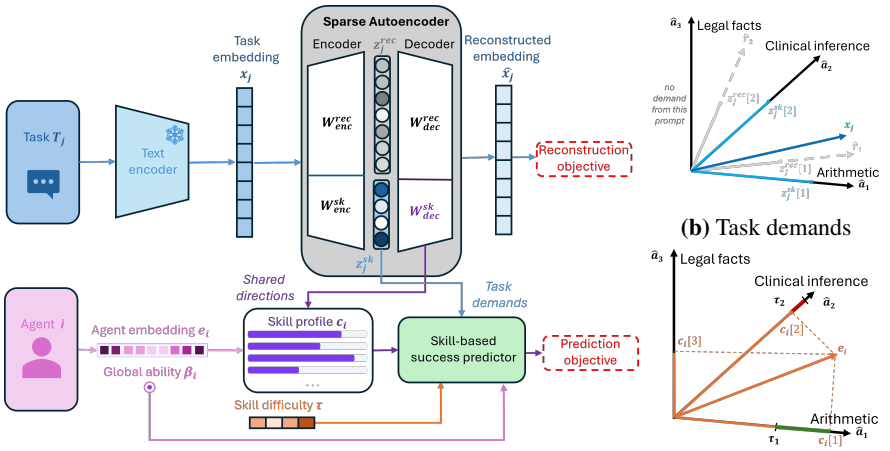
$$\mathbf{x}_j \;\approx\; \sum_{k=1}^{C} z_j^{\mathrm{sk}}[k]\,\hat{\mathbf{a}}_k + \sum_{p=1}^{R} z_j^{\mathrm{rec}}[p]\,\hat{\mathbf{r}}_p, \tag{1}$$

with $z_j^{\mathrm{sk}}[k] \geq 0$ read as the *demand* task j places on skill k. Only the skill coefficients enter the outcome model, and the reconstruction directions absorb embedding variation that need not support prediction. Each skill carries a difficulty threshold τ_k, and every agent has a trainable embedding $\mathbf{e}_i$ whose projection $c_i[k] = \hat{\mathbf{a}}_k^\top \mathbf{e}_i$ is its *ability* on skill k. Using the same directions on both sides is what makes demands and abilities comparable, and the agent embedding is fitted from observed outcomes rather than taken from anything inside the agent. The log-odds of success is

$$\eta_{i,j} = \beta_i + \sum_{k=1}^{C} m_{i,j}[k], \qquad m_{i,j}[k] = z_j^{\mathrm{sk}}[k]\big(c_i[k] - \tau_k\big), \qquad P(y_{i,j} = 1) = \sigma(\eta_{i,j}), \tag{2}$$

where β_i is a global ability intercept and $m_{i,j}[k]$ is the *margin* on skill k. A margin is positive when the task demands the skill and the agent clears its threshold, negative when the agent falls short, and zero when the task places no demand, so every prediction is an exact sum of a global term and a few per-skill terms. Relative to multidimensional IRT, $\sigma(\mathbf{a}_j^\top \boldsymbol{\theta}_i - b_j)$ (Reckase, 2009), the sparse demand vector plays the role of discrimination, $\mathbf{c}_i$ of ability, and $(\mathbf{z}_j^{\mathrm{sk}})^\top \boldsymbol{\tau}$ of an effective difficulty assembled from the skills a task demands; since every task-side quantity is computed from $\mathbf{x}_j$, a new task is scored without fitting a new parameter (Figure 1).

Architecture. The encoder centers each embedding, applies an affine map, and takes a ReLU (Bricken et al., 2023), with the offset fixed to the training-set mean and added back during decoding, which keeps the large shared mean component of frozen embeddings (Mu & Viswanath, 2018) out



(a) Model and training framework

(b) Task demands

(c) Agent abilities

Figure 1: **Agent abilities and task demands share one set of directions.** **(a)** A frozen encoder embeds task T_j as $\mathbf{x}_j$. Sparse skill and reconstruction features jointly reconstruct it, only skill demands enter the outcome head, and abilities are projections of $\mathbf{e}_i$ onto the skill decoder directions. **(b)** A task places sparse demands on skill directions (blue); auxiliary directions (gray) absorb the remaining variation. **(c)** Abilities lie on the same directions (orange), and demand-weighted gaps to $\boldsymbol{\tau}$ give the margins of Eq. 2.

of the skill partition. The weights split into skill and reconstruction blocks, following the partitioned design of AlignSAE (Yang et al., 2026) with agent outcomes in place of a concept ontology, and BatchTopK (Bussmann et al., 2024) is applied separately to the two so that neither consumes the other's activation budget. Unit-norm decoder columns remove the rescaling freedom between a direction and its activation (Mairal et al., 2012), and at evaluation a fixed per-partition threshold is used (Bussmann et al., 2024), making demands deterministic (Appendix A).

Coupled prediction head. Reading abilities off the decoder does not shrink the set of expressible profiles: with $\mathbf{A} = \mathbf{W}_{\text{dec}}^{\text{sk}}$ of full column rank, as in all six reported skill blocks, $\mathbf{e} \mapsto \mathbf{A}^\top \mathbf{e}$ is onto $\mathbb{R}^C$, so coupling only changes how profiles are regularized (Appendix A.4). It does change what the outcome loss can reach, since under a free head the prediction gradient with respect to a decoder column is zero, whereas under coupling each column rotates toward the embeddings of agents that succeed when its skill is demanded (Appendix A.3).

Training. We optimize the encoder, decoder, agent embeddings, intercepts, and thresholds jointly against reconstruction error, binary cross-entropy on $\sigma(\eta_{i,j})$, an auxiliary term that revives dead directions (Gao et al., 2024), a cross-partition term that keeps the reconstruction partition from re-encoding skill structure (Yang et al., 2026), and L_2 penalties on β_i and $\mathbf{e}_i$. Training starts with a reconstruction-only warmup, and each β_i is initialized at the logit of agent i's training accuracy so that global ability starts in its intended term (Appendix A).

Model selection. Validation loss alone does not fix the basis. On the controlled pool, removing the encoder's centering keeps validation cross-entropy within 0.003, but three skills go dead and two fire on nearly every task. For each dataset and encoder we sweep about 40 configurations and keep the lowest-loss one that passes three checks fixed in advance, none of which uses domain labels: $\text{corr}(\beta_i, \text{acc}_i) > 0.95$, no skill firing on over 90% of tasks, and mean profile norm above 0.5. Appendices I and J give a small audit of the checks, a sweep over C, and the grids.

Placing a new agent. Freezing the encoder, the directions, and $\boldsymbol{\tau}$ makes Eq. 2 linear in the new intercept β and profile $\mathbf{c}$, with $\mathbf{z}_j^{\text{sk}}$ as the design vector and $-(\mathbf{z}_j^{\text{sk}})^\top \boldsymbol{\tau}$ as a known offset. A new agent is fitted from its observed responses S by an L_2-regularized logistic regression in $C+1$ parameters,

$$(\hat{\beta}, \hat{\mathbf{c}}) = \arg\min_{\beta, \mathbf{c}} \sum_{j \in S} \text{BCE}\Big(\sigma\big(\beta + \textstyle\sum_k z_j^{\text{sk}}[k](c[k] - \tau_k)\big), y_j\Big) + \tfrac{\lambda}{2}\|\mathbf{c}\|_2^2, \tag{3}$$

solved with L-BFGS or iteratively reweighted least squares (McCullagh & Nelder, 1989). The same fit applied at two points in an agent's training history gives a profile change in fixed coordinates, which is the estimator used in Section 5.5.

4 What a profile fit can recover

Section 5 reads agent profiles from observed responses with Eq. 3, both to place new agents and to track an agent through retraining. We first ask which agent differences a task set can reveal.

What responses can determine In this analysis, we fix the learned basis, so the demands of every task and the thresholds $\boldsymbol{\tau}$ are known. Eq. 2 depends on abilities only through the gaps $\mathbf{g}_i = \mathbf{c}_i - \boldsymbol{\tau}$, so adding the same vector to every profile and to the thresholds changes no prediction. Profiles can therefore only be compared with each other, including one agent's profile before and after retraining. For a set S of m tasks with demand matrix Z_S, the logits of agent i are $\boldsymbol{\eta}_{i,S} = \beta_i \mathbf{1} + Z_S \mathbf{g}_i$. A profile change Δ that raises the logit of every task by the same amount has the same effect as a larger β_i, so the responses cannot reveal it. In matrix form, with column-centered demands $Z_S^\circ = H_S Z_S$ and $H_S = I_m - \mathbf{1}\mathbf{1}^\top/m$, a change is invisible exactly when $Z_S^\circ \Delta = 0$, and a direction with a small singular value of Z_S° is resolved only weakly (Appendix B.1). Along such a direction, the estimate from Eq. 3 is set by the regularizer rather than by the responses.

Balanced task sets can hide a skill difference. Covering every skill equally often does not ensure that every skill difference is measured. Suppose each task demands one skill from a group U and one skill from a group V. An agent that is slightly stronger on every skill in U and slightly weaker on every skill in V gains and loses the same amount on each task, so none of its predictions change. Adding a few tasks that pair two skills from the same group removes the ambiguity. More generally, suppose every task demands two skills and the graph that links co-demanded skills is connected. Then a contrast beyond the common shift is hidden exactly when the graph is bipartite, which is a standard property of incidence matrices (Appendix B.2).

Experimental validation. To see the effect on real responses, we need a setting where we know which skills each task demands and can choose how the skills are paired. The language-model pools of Section 5 do not allow this, because their demands are learned and continuous, so we can neither build a task set with an exactly hidden direction nor know the true profile to measure error against. We therefore use a procedural world with eight operations, in which each task is a short program and the operations it uses are its demands (Appendix B.3). The agents are 50 MLPs trained by SGD on different mixtures of operations, so their strengths are learned rather than specified by us, and they vary and overlap as in a real agent pool. Because every agent can be evaluated on all 28 operation pairs, we fit a reference profile for each agent and measure how far a fit from a smaller task set falls from it. We compare two designs that each have 12,000 tasks and 3,000 appearances of every operation, and that differ only in which operations are paired. Under the bipartite design, the error along the hidden direction stays at 0.913 as the number of responses grows from 32 to 1,024, while the error along the other directions falls by a factor of 9.6 (Figure 2). Fitting the agents' actual correct and incorrect answers gives the same result, with the hidden direction flat at 0.913 and the other directions falling by a factor of 5.3. Under the non-bipartite design, the same direction is measured, and its error falls to 0.302. As few as 20 same-group tasks among the 12,000 also make it measurable, although it is then estimated with high variance (Appendix B.3). Because these fits use the estimator of Sections 5.4 and 5.5, its accuracy along a direction depends on the task design, not only on the number of responses.

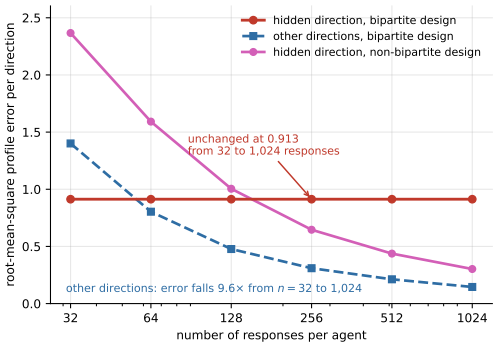


Figure 2: **More responses do not measure a skill difference that the tasks hide.** Profile error averaged over 50 trained agents. In the bipartite design, the error along the hidden direction (red) never changes, while the other directions (blue) improve. Tasks that pair two skills from the same group make the hidden direction measurable (pink). Appendix B.3 gives details.

Consequences for the experiments. The analysis shapes how Section 5 reads profiles. First, because profiles can only be compared with each other, we compare agents within a skill (Section 5.3) and measure retraining as the change in one agent's profile (Section 5.5). Second, the regularizer sets the estimate along weakly resolved directions, so we refit both endpoints of an adaptation with the same estimator (Appendix G). A measured change then reflects the agent rather than a difference between fitting procedures. Third, a support set must vary its demands. The learned demands do: on every dataset and encoder, no direction among the live skills is exactly hidden, and random supports of a few hundred tasks worked as well as structured selection (Appendix F).

5 Do the coordinates behave as measurements?

We test the four properties of Section 1 on real language-model agent populations.

5.1 Experimental Setup

Data. EmbedLLM (Zhuang et al., 2024) records binary correctness for 112 agents on 35,673 prompts from ten public benchmarks, and RouterBench (Hu et al., 2024) contributes 11 agents on 36,497 prompts from 14 task families with graded scores thresholded at 0.5. The controlled pool is built from MMLU-Pro (Wang et al., 2024) and contains 16 agents derived from Qwen2.5-1.5B-Instruct (Qwen Team, 2025), namely ten LoRA specialists (Hu et al., 2021) each trained on a known mixture of two or three of the 14 domains, the unadapted base model, and five chain-of-thought variants. Every domain is covered by exactly two specialists and no two share a mixture, so domain identity alone cannot separate the two experts covering a domain from the eight that do not, and the mixtures are never given as labels (Appendix C). All splits are prompt-level.

Encoders and baselines. We use the two frozen encoders of Yao et al. (2026), all-mpnet-base-v2 (Reimers & Gurevych, 2019) and ModernBERT-Large (Warner et al., 2024), and rerun every baseline on identical splits and embeddings. The untrained controls are per-agent and per-domain majority rules. The learned baselines are EmbedLLM matrix factorization and kNN (Zhuang et al., 2024), IrtNet (Chen et al., 2025), JE-IRT (Yao et al., 2026), correctness-only IRT-Router (Song et al., 2025), and an unconstrained linear probe. Our JE-IRT reproduction lands within 0.2 points of the published EmbedLLM number and our kNN and IrtNet ports exceed theirs (Appendix D).

5.2 Predictive sufficiency

SkillBasis is not built to be the most accurate predictor, so the question is what its interpretable constraint costs. On EmbedLLM and RouterBench SkillBasis is within about 0.3 points of the best learned method under both encoders and above matrix factorization, kNN, and the linear probe in each of these four columns (Table 1), so passing every prediction through 8 to 32 shared coordinates, with only 2 to 8 active per task, costs almost nothing there. On the controlled pool, which is constructed by domain, the gap is 1.1 points with ModernBERT-Large and 2.1 with all-mpnet-base-v2; Section 5.5 shows that learned demands add information beyond domains.

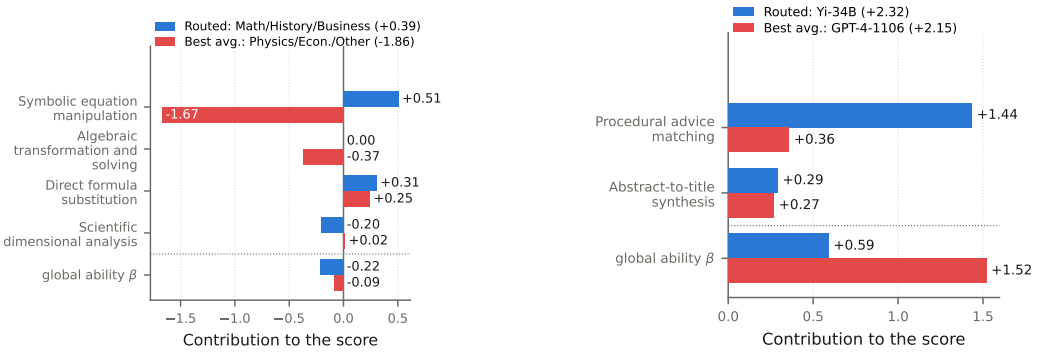
Consequence: explaining agent differences. This is what the small accuracy cost buys: because Eq. 2 is an exact identity, the score gap between two fitted agents on a task decomposes exactly into a global-ability difference and a few skill margins. Selecting the agent with the highest predicted success for each prompt, and keeping the prompts where that selection differs from the strongest agent overall, skill margins carry 79% of the score-gap magnitude on the controlled pool and 71% on EmbedLLM, with an average of 4.1 and 2.0 nonzero skill terms.

Figure 3 shows the two cases that arise: one skill separating two agents of nearly equal global ability ($+0.51$ against -1.67 on symbolic equation manipulation), and skill matching overturning a 0.93 global-ability deficit (2.32 against 2.15). Because the components sum exactly to the score, the account is the model's own arithmetic rather than a post-hoc rationale. Skill names are LLM-generated labels (Section 5.3).

Table 1: **A prediction constrained to pass through a small shared skill basis stays close to dense predictors.** Test accuracy at a 0.5 probability threshold. Subscripts are standard deviations over three seeds (e.g., $0.746_{\pm 2} = 0.746 \pm 0.002$). Deterministic controls and kNN are reported once. Bold marks the best learned method in each column.

	EmbedLLM		RouterBench		MMLU-Pro experts	
Method	mpnet	M-BERT	mpnet	M-BERT	mpnet	M-BERT
Majority (*per agent*)	0.636		0.681		0.632	
Majority (*per domain*)	0.711		0.607		0.639	
kNN (Zhuang et al., 2024)	0.742	0.741	0.712	0.718	0.698	0.678
MF (Zhuang et al., 2024)	$0.734_{\pm 0}$	$0.732_{\pm 1}$	$0.707_{\pm 0}$	$0.712_{\pm 1}$	$0.688_{\pm 1}$	$\mathbf{0.697}_{\pm \mathbf{0}}$
IrtNet (Chen et al., 2025)	$\mathbf{0.746}_{\pm \mathbf{0}}$	$0.748_{\pm 1}$	$0.717_{\pm 0}$	$\mathbf{0.724}_{\pm \mathbf{0}}$	$\mathbf{0.699}_{\pm \mathbf{2}}$	$0.690_{\pm 1}$
IRT-Router (Song et al., 2025)	$0.740_{\pm 1}$	$0.746_{\pm 1}$	$0.716_{\pm 0}$	$\mathbf{0.724}_{\pm \mathbf{0}}$	—†	
Linear probe	$0.729_{\pm 0}$	$0.737_{\pm 1}$	$0.703_{\pm 0}$	$0.715_{\pm 0}$	$0.669_{\pm 1}$	$0.680_{\pm 1}$
JE-IRT (Yao et al., 2026)	$0.745_{\pm 0}$	$\mathbf{0.749}_{\pm \mathbf{0}}$	$\mathbf{0.718}_{\pm \mathbf{0}}$	$0.717_{\pm 1}$	$0.689_{\pm 2}$	$0.690_{\pm 2}$
SkillBasis (ours)	$0.744_{\pm 1}$	$0.746_{\pm 1}$	$0.716_{\pm 1}$	$0.722_{\pm 2}$	$0.678_{\pm 2}$	$0.686_{\pm 2}$

†IRT-Router embeds each agent's name, and the MMLU-Pro expert names encode their training mixtures, so we report no number rather than a leaked one.



(a) Similar global ability

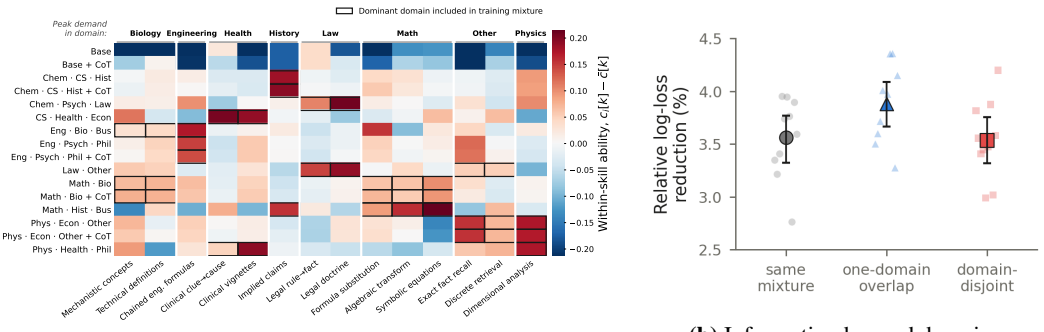
(b) Skill match overrides global ability

Figure 3: **Skill margins can separate near-tied agents or overturn a global-ability disadvantage.** Each panel decomposes both agents' scores into the global-ability term and the skill margins that differ. **(a)** On the controlled pool, two agents with similar global ability are separated mainly by symbolic equation manipulation. **(b)** On RouterBench, procedural advice matching overcomes a global-ability disadvantage. Every displayed component sums exactly to the corresponding score.

5.3 Alignment with known specialization

Agents that were deliberately trained differently should occupy correspondingly different locations. The controlled pool provides independently known domain-specific training histories that are never supplied to the model. We use these domains only after fitting as external pseudolabels for validating the learned profiles; we do not assume that a learned skill is a domain, that every domain has its own skill, or that skills and domains correspond one-to-one. To define a reproducible validation statistic, we associate each skill with the domain on which its mean demand is highest. *Selectivity AUC* is then the probability that a specialist trained on this associated domain scores higher on the skill than one that was not, averaged within skills so that no comparison crosses a skill.

The checkpoints selected in Section 3 reach selectivity 0.943 with ModernBERT-Large and 0.950 with all-mpnet-base-v2 (appendix Table 5), against null means of 0.499 and 0.495 from reassigning the ten planted mixtures among the specialists with profiles fixed; none of 2,000 shuffles reaches either value. Six ModernBERT-Large retrainings average 0.878 ± 0.059, with the weakest at 0.774. All 14 ModernBERT-Large skills and 14 of the 15 live all-mpnet skills exceed chance on their own. Figure 4a shows this correspondence agent by agent. The task side agrees as well, since domain explains 33.3% and 40.7% of the variation in demand under the two encoders against about 0.1% after shuffling the labels.



(a) Specialization alignment

(b) Information beyond domains

Figure 4: **Profiles align with specialization while learned demands add predictive information beyond domains.** **(a)** Original controlled pool: profiles are centered across 16 agents with ModernBERT-Large. Outlined cells mark agents trained on that domain, which tend to have larger within-skill scores. **(b)** The 30 adapted endpoints of Section 5.5 in the same frozen basis: each dot is the out-of-fold log-loss reduction from adding skills to domain effects with a shared frozen offset. All improve. Large markers and bars show condition means and 95% confidence intervals.

Each coordinate is used where its demands say it is used. Zeroing any skill's demand $z_j^{\mathrm{sk}}[k]$ raises held-out cross-entropy wherever that skill fires, and the mean effect on tasks where it is the dominant demand is 9.6 times that elsewhere (Figure 9, Appendix I); because the logit is exactly additive, this test needs no generated label.

Controls, and what the pool does not test. Frozen task embeddings already carry part of the planted structure: a dictionary frozen after warmup reaches 0.733 ± 0.041 under ModernBERT-Large against the six-run supervised 0.878 ± 0.059 (all-mpnet: 0.864 ± 0.025 against one supervised run at 0.950), and supervision also improves prediction on the pool by about 3.5 points, so outcomes add structure that the embeddings alone do not carry. A k-means topic anchor organizes tasks by domain less well than our demands under ModernBERT-Large and better under all-mpnet, and same-domain skills rank agents differently (Appendix E, Table 6).

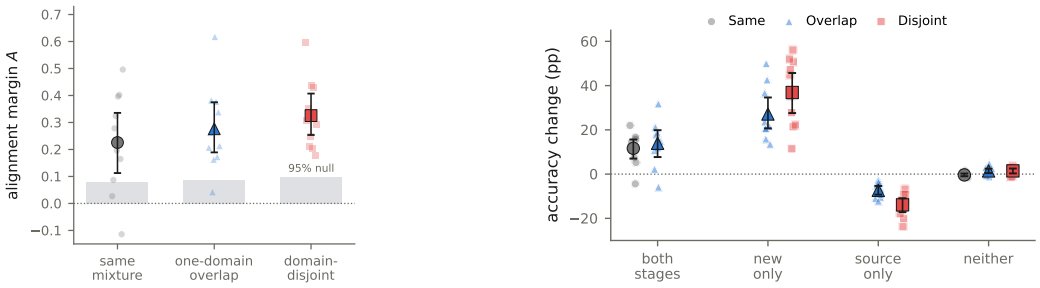
Following established protocols (O'Neill et al., 2024; Bills et al., 2023), an LLM names each skill from its highest-demand tasks, and a held-out judge seeing only that label identifies activating tasks with mean accuracy 0.830, which we treat as supporting evidence with the behavioral tests primary (Heap et al., 2026). Because the available controlled interventions are domain-specific, this experiment tests agreement with independently known training differences; Section 5.5 shows that the learned demands also carry information beyond domain identity.

5.4 Efficient measurement

Placing a new agent costs $C + 1$ parameters and no retraining of the task side; the learned demands form the design of Eq. 3 analyzed in Section 4. Following the onboarding protocols of Yao et al. (2026), we first remove ten EmbedLLM agents before the basis is learned and profile them in the frozen basis. With all-mpnet-base-v2, SkillBasis reaches 73.3% accuracy from 1% support (341 responses) and 73.6% from 10%, against 73.7% when the same agents are included during training; IrtNet is stronger in absolute terms at 74.1% from the same support. For six recent agents that contribute no outcomes when the basis is learned, SkillBasis is within about one point of matrix factorization, IrtNet, and JE-IRT (232 to 256 parameters) at 1% support but saturates at 57.5% where JE-IRT reaches 62.4% with full support (Table 8), so one number per skill is competitive when responses are scarce but not when they are plentiful.

5.5 Intervention consistency

The strongest test of a measurement is whether it responds correctly when the measured system is changed. Each of the ten specialists receives each of three further training mixtures, namely its own domains, a mixture sharing exactly one domain, or a disjoint mixture, for 30 balanced adaptations (Appendix G gives the protocol and full results). Holding the basis fixed, we refit the agent at



(a) Profile alignment **(b)** Domain accuracy

Figure 5: **The profile shifts toward the domains an agent was actually retrained on, and accuracy changes track the same domains. (a)** Cosine of each profile change with the actual adaptation mixture minus its mean cosine with the nine alternatives; shading marks the shuffled-mixture reference. **(b)** Accuracy change by domain group. Error bars are 95% confidence intervals.

both endpoints with the same estimator of Eq. 3 (Section 4). We compare each profile change $\Delta c_{i,D}$ with the mean demand of the mixture used for adaptation and with the nine alternatives. A positive margin means that the profile moved more toward the data the agent was actually trained on than toward the average alternative. The margin is positive in 29 of 30 adaptations (the exception also lost 4.4 points on its retraining domains; Table 10), with condition means of 0.226, 0.277, and 0.326 for same, overlapping, and disjoint mixtures, each above its shuffled-mixture reference (Figure 5a), and the ordering is the expected one, since a disjoint mixture should move a profile further than a repetition of its own domains. Measured behavior moves the same way (Figure 5b), since newly introduced domains gain 27.2 points under one-domain overlap and 36.9 points under disjoint adaptation, omitted source domains lose 7.3 and 13.9 points, and domains in neither stage change by -0.4 to $+1.4$ points.

Predictive information beyond domains. The adapted agents also show that learned demands add predictive information beyond domain identity. With the basis frozen, we fit each of the 30 endpoints with domain effects and a shared frozen difficulty offset, with and without the learned skill demands. Profile fitting is cross-validated over five shared, domain-balanced folds of the 12,032 questions, with duplicate clusters kept together. Adding skills reduces out-of-fold log loss by 3.7% on average and improves all 30 endpoints (Figure 4b; Appendix H).

6 Discussion

Item-level outcomes and frozen task embeddings suffice to learn a shared basis for what tasks demand and what agents can do. Every prediction of SkillBasis is an exact sum of a global term and a few skill margins, at a cost of at most 0.3 accuracy points against the best dense predictors on EmbedLLM and RouterBench. A profile fit recovers only the differences that a task set's demands vary, so balanced skill coverage can leave a skill difference unmeasured. In our tests, the coordinates behave as measurements: they align with training differences never shown to the model, place a new agent from a few hundred responses, and move toward the data an agent is retrained on in 29 of 30 adaptations. They also explain why an agent is preferred on a task and track what retraining changed. More broadly, learned representations of agents should be judged by whether they behave as measurements, not by predictive fit alone, and benchmarks by which skill contrasts they measure, not by how evenly they cover skills.

Limitations. Beyond the scope stated in Section 1, the basis depends on the frozen encoder and the task population. Selectivity stays well above its null across retrainings, but skills matched across seeds agree only partly (Table 6), so the basis is reproducible in aggregate rather than coordinate by coordinate, and we compare agents only within one frozen basis. One number per skill saturates below a dense embedding when responses are plentiful (Section 5.4), graded scores are binarized, and we report uncertainty across agents and seeds rather than for individual profiles.

AI USE STATEMENT

LLM tools were used to improve the clarity and presentation of the manuscript and to assist with reviewing, debugging, and testing experimental code. All suggested changes were inspected by the authors, and all reported results were produced and verified using the authors' experimental pipelines. The authors take full responsibility for the content of the paper, the correctness of the implementation, and the interpretation of the results. Within the method, an LLM names learned skills and an LLM judge tests those names (Section 5.3, Appendix E).

BROADER IMPACT STATEMENT

Named coordinates make agent evaluation and procurement easier to audit by exposing the factors behind a score. The same profiles can characterize proprietary systems from observed outputs, and generated labels can carry the biases of the judge model. Any report built on the method should state the evaluated task distribution, the uncertainty, and the annotation procedure, and should support rather than replace human oversight.

REPRODUCIBILITY STATEMENT

Appendix A gives the full training objective, schedule, and inference rule, and Appendix B gives the measurement proofs, the procedural environment, and the controlled measurement experiment. Appendix C describes how each success matrix was built, including the construction of the controlled expert pool, and Appendix D gives baseline settings and reproduction checks against published numbers. Appendices E–I give the protocols for the alignment metrics, onboarding, adaptation, and the beyond-domain test, together with ablations and selection diagnostics, and Appendix J lists every search range and the selected configuration for each reported model. Code and the controlled expert pool will be released.

REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. OpenAI, 2023. URL https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

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

Ryan Burnell, Han Hao, Andrew R. A. Conway, and José Hernández-Orallo. Revealing the structure of language model capabilities. *arXiv preprint arXiv:2306.10062*, 2023a.

Ryan Burnell, Wout Schellaert, John Burden, Tomer D. Ullman, Fernando Martínez-Plumed, Joshua B. Tenenbaum, Danaja Rutar, Lucy G. Cheke, Jascha Sohl-Dickstein, Melanie Mitchell, Douwe Kiela, Murray Shanahan, Ellen M. Voorhees, Anthony G. Cohn, Joel Z. Leibo, and José Hernández-Orallo. Rethink reporting of evaluation results in AI. *Science*, 380(6641):136–138, 2023b. doi: 10.1126/science.adf6369.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Jianhao Chen, Chenxu Wang, Gengrui Zhang, Peng Ye, Lei Bai, Wei Hu, Yuzhong Qu, and Shuyue Hu. Learning compact representations of LLM abilities via item response theory. *arXiv preprint arXiv:2510.00844*, 2025.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Leander Girrbach and Zeynep Akata. Sparse autoencoders are topic models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2511.16309.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Automated interpretability metrics do not distinguish trained and random transformers. *arXiv preprint arXiv:2501.17727*, 2026.

Valentin Hofmann, David Heineman, Ian Magnusson, Kyle Lo, Jesse Dodge, Maarten Sap, Pang Wei Koh, Chun Wang, Hannaneh Hajishirzi, and Noah A. Smith. Fluid language model benchmarking. In *Conference on Language Modeling (COLM)*, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. Measuring progress in dictionary learning for language model interpretability with board game models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2408.00113.

Alex Kipnis, Konstantinos Voudouris, Luca M. Schulze Buschoff, and Eric Schulz. metabench – a sparse benchmark of reasoning and knowledge in large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348, 2020.

Jia Liu, Zhiyu Xu, and Yuqi Gu. Scalable text-embedding-informed cognitive diagnosis of large language models. *arXiv preprint arXiv:2603.14676*, 2026. doi: 10.48550/arXiv.2603.14676.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024.

Yunting Liu, Shreya Bhandari, and Zachary A Pardos. Leveraging llm respondents for item evaluation: A psychometric analysis. *British Journal of Educational Technology*, 56(3):1028–1052, 2025.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *arXiv preprint arXiv:2106.13314*, 2021.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326, 2024.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. Sloth: scaling laws for LLM skills to predict multi-benchmark performance across families. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2412.06540.

Aviya Maimon, Amir DN Cohen, Gal Vishne, Shauli Ravfogel, and Reut Tsarfaty. From benchmarks to skills: Low-rank factors for LLM evaluation. *arXiv preprint arXiv:2507.20208*, 2025.

Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012.

Aleksandar Makelov, George Lange, and Neel Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2405.08366.

Peter McCullagh and John A. Nelder. *Generalized Linear Models*, volume 37 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London, 2 edition, 1989. doi: 10.1007/978-1-4899-3242-6.

Mazda Moayeri, Vidhisha Balachandran, Varun Chandrasekaran, Safoora Yousefi, Thomas Fel, Soheil Feizi, Besmira Nushi, Neel Joshi, and Vibhav Vineet. Unearthing skill-level insights for understanding trade-offs of foundation models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.13826.

Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations (ICLR)*, 2018.

Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2023.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. doi: 10.1038/381607a0.

Charles O'Neill, Christine Ye, Kartheik Iyer, and John F. Wu. Disentangling dense embeddings with sparse autoencoders. *arXiv preprint arXiv:2408.00657*, 2024.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations (ICLR)*, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Georg Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danish Institute for Educational Research, Copenhagen, 1960.

Mark D. Reckase. *Multidimensional Item Response Theory*. Statistics for Social and Behavioral Sciences. Springer, New York, NY, 2009. doi: 10.1007/978-0-387-89976-3.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pp. 4486–4503, 2021. doi: 10.18653/v1/2021.acl-long.346.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 15841–15892, 2024.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. doi: 10.1038/s42256-019-0048-x.

Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. arXiv:2506.01048.

Sang T. Truong, Yuheng Tu, Percy Liang, Bo Li, and Sanmi Koyejo. Reliable and efficient amortized model-based evaluation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 60238–60265, 2025.

Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6153–6161, 2020. doi: 10.1609/aaai.v34i04.6080.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. arXiv:2406.01574.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*, 2024.

Gongjun Xu and Stephanie Zhang. Identifiability of diagnostic classification models. *Psychometrika*, 81(3):625–649, 2016. doi: 10.1007/s11336-015-9471-z.

Minglai Yang, Xinyu Guo, Zhengliang Shi, Jinhe Bi, Steven Bethard, Mihai Surdeanu, and Liangming Pan. AlignSAE: Concept-aligned sparse autoencoders. *arXiv preprint arXiv:2512.02004*, 2026.

Louie Hong Yao, Nicholas Jarvis, Tiffany Zhan, Saptarshi Ghosh, Linfeng Liu, and Tianyu Jiang. JE-IRT: A geometric lens on LLM abilities through joint embedding item response theory. *Transactions on Machine Learning Research*, 2026.

Amir R. Zamir, Alexander Sax, Bokui Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Hongli Zhou, Hui Huang, Ziqing Zhao, Lvyuan Han, Huicheng Wang, Kehai Chen, Muyun Yang, Wei Bao, Jian Dong, Bing Xu, Conghui Zhu, Hailong Cao, and Tiejun Zhao. Lost in benchmarks? rethinking large language model benchmarking with item response theory. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026a. arXiv:2505.15055.

L. Zhou, L. Pacchiardi, F. Martínez-Plumed, et al. General scales unlock ai evaluation with explanatory and predictive power. *Nature*, 652:58–67, 2026b. doi: 10.1038/s41586-026-10303-2. URL https://doi.org/10.1038/s41586-026-10303-2.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. *arXiv preprint arXiv:2410.02223*, 2024.

A Model and training details

A.1 Loss components

The components of the joint objective are

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_j\left[\|\mathbf{x}_j - \hat{\mathbf{x}}_j\|_2^2\right], \tag{4}$$

$$\mathcal{L}_{\text{pred}} = \mathbb{E}_{(i,j)}[-y_{i,j}\log\hat{y}_{i,j} - (1-y_{i,j})\log(1-\hat{y}_{i,j})], \quad \hat{y}_{i,j} = \sigma(\eta_{i,j}), \tag{5}$$

$$\mathcal{L}_{\text{reg}} = \lambda_\beta \sum_i \beta_i^2 + \lambda_e \sum_i \|\mathbf{e}_i\|_2^2. \tag{6}$$

The auxiliary term is the TopK-aux loss of Gao et al. (2024). At each step the top-k_{aux} pre-activations among directions that have not fired for a fixed number of steps reconstruct the current residual $\mathbf{x}_j - \hat{\mathbf{x}}_j$ through their decoder columns alone, and $\mathcal{L}_{\text{aux}}$ is the mean squared error of that reconstruction. The cross-partition term penalizes linear dependence between the partitions. Writing $\tilde{z}_k^{\text{sk}}$ and $\tilde{z}_p^{\text{rec}}$ for the batch-centered unit-normalized activation columns,

$$\mathcal{L}_{\text{orth}} = \frac{1}{CR}\sum_{k=1}^{C}\sum_{p=1}^{R}\left(\tilde{z}_k^{\text{sk}} \cdot \tilde{z}_p^{\text{rec}}\right)^2, \tag{7}$$

following AlignSAE (Yang et al., 2026). A direction with zero batch variance contributes zero. The penalties in $\mathcal{L}_{\text{reg}}$ stop the model from concentrating all predictive signal in β_i, which would reduce it to a scalar ranking baseline, or in large agent embedding norms, and the λ_e term additionally absorbs the scale freedom between agent embeddings and skill directions.

A.2 Schedule and deterministic inference

Training proceeds in two stages. First, for E_{rec} epochs the prediction loss is disabled and only the reconstruction objective is optimized, which lets the dictionary stabilize before any outcome pressure is applied. Second, all parameters are optimized jointly. Intercepts are seeded as $\beta_i = \text{logit}(\min\{0.98, \max\{0.02, \bar{y}_i\}\})$ with $\bar{y}_i$ the mean training accuracy of agent i, where the bound prevents an infinite logit for an agent with an empirical rate of zero or one.

BatchTopK couples a task's support to the other tasks in its mini-batch. At evaluation we follow Bussmann et al. (2024) and replace it with fixed per-partition JumpReLU thresholds, calibrated on validation data from the minimum activations retained under BatchTopK so that the expected per-task L_0 budget is matched. Demands are then deterministic and independent of batch composition.

A.3 Gradient flow under decoder coupling

Write $\delta_{i,j} = \sigma(\eta_{i,j}) - y_{i,j}$ for the prediction residual of one pair. The logit depends on trainable parameters along two paths.

Ability path. Because $c_i[k] = \hat{\mathbf{a}}_k^\top \mathbf{e}_i$ and every column has unit norm,

$$\frac{\partial \mathcal{L}_{\text{pred}}}{\partial \mathbf{W}_{\text{dec}}^{\text{sk}}[:,k]} = \delta_{i,j}\, z_j^{\text{sk}}[k]\left(\mathbf{I} - \hat{\mathbf{a}}_k\hat{\mathbf{a}}_k^\top\right)\mathbf{e}_i, \qquad \frac{\partial \mathcal{L}_{\text{pred}}}{\partial \mathbf{e}_i} = \delta_{i,j}\, \mathbf{W}_{\text{dec}}^{\text{sk}}\, \mathbf{z}_j^{\text{sk}}. \tag{8}$$

The update to a direction is proportional to the task's demand for it, the tangent-space projection keeps the update a rotation, and the direction rotates toward $\mathbf{e}_i$ on success and away on failure. The agent embedding moves toward the demand-weighted directions of the tasks it solves. A free head gives zero gradient into the decoder column, so no outcome signal reaches the directions at all. The two updates are mutually attracting, which raises the possibility of a joint drift that changes no logit, and two asymmetries prevent it. The decoder columns are pinned to the unit sphere and anchored by $\mathcal{L}_{\text{rec}}$ to the fixed geometry of ϕ, while the agent embeddings are free in direction and shrunk by λ_e, so the directions serve as the reference frame and the agents position themselves inside it.

Demand path. Demands enter the logit linearly, so for each retained skill

$$\frac{\partial \mathcal{L}_{\text{pred}}}{\partial z_j^{\text{sk}}[k]} = \delta_{i,j}\left(c_i[k] - \tau_k\right), \qquad \frac{\partial \mathcal{L}_{\text{pred}}}{\partial \mathbf{W}_{\text{enc}}^{\text{sk}}[k,:]} = \delta_{i,j}\left(c_i[k] - \tau_k\right)(\mathbf{x}_j - \boldsymbol{\mu})^\top. \tag{9}$$

The encoder row for a skill is adjusted in proportion to how decisive the skill is for the observed outcome, so the encoder learns to assign demand where demand changes predictions and not only where it aids reconstruction. The reconstruction rows receive prediction gradient only indirectly.

A.4 Rank and induced regularization

Let $\mathbf{A} = \mathbf{W}_{\mathrm{dec}}^{\mathrm{sk}}$. All six reported skill blocks are full column rank, with smallest singular values from 0.023 to 0.577 and condition numbers from 3.7 to 121.7, the poorer conditioning coming from all-mpnet-base-v2 as C grows. Full rank means every profile is realizable by at least one agent embedding when realization is required, and the direct fit of Eq. 3 estimates $\mathbf{c}$ itself and does not need the condition. For a realizable profile the minimum-norm embedding is $\mathbf{e}^\star = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{c}$ with $\|\mathbf{e}^\star\|_2^2 = \mathbf{c}^\top(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{c}$, so the training-time penalty on agent embeddings induces a geometry-dependent rather than isotropic penalty on profiles.

B Measurement geometry

Which item-to-skill structures make skill profiles identifiable is a classical question in cognitive diagnosis (Xu & Zhang, 2016); here the demands are fixed and continuous, and we ask which profile differences a given task set determines.

B.1 Profile distinguishability

For a frozen basis, write $g = c - \tau$. For a set S of m tasks, let Z_S contain their demand vectors and define

$$H_S = I_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top, \qquad Z_S^\circ = H_S Z_S.$$

The logit vector is

$$\eta_S(\beta, g) = \beta\mathbf{1} + Z_S g.$$

A profile change Δ is exactly invisible after refitting the intercept if and only if

$$Z_S^\circ \Delta = 0.$$

Indeed, this condition is equivalent to $Z_S\Delta$ being constant across tasks, in which case the change can be absorbed by β.

The same statement has a quantitative form away from the null space.

Lemma B.1. *For independent Bernoulli responses under the logistic model,*

$$\min_b D_{\mathrm{KL}}\big(P^S_{\beta,g} \,\|\, P^S_{\beta+b,g+\Delta}\big) \le \frac{1}{8}\|Z_S^\circ\Delta\|_2^2.$$

Proof. Choose

$$b^\star = -\frac{1}{m}\mathbf{1}^\top Z_S\Delta.$$

The resulting change in the logit vector is

$$b^\star\mathbf{1} + Z_S\Delta = H_S Z_S\Delta = Z_S^\circ\Delta.$$

For one response, the Bernoulli logistic model is an exponential family with log-partition $A(\eta) = \log(1+\exp\eta)$. Since

$$A''(\eta) = \sigma(\eta)(1-\sigma(\eta)) \le \frac{1}{4},$$

the Bregman divergence between logits η and $\eta+\delta$ is at most $\delta^2/8$. Summing over independent tasks gives the result. □

If v is a unit right singular vector of Z_S° with singular value σ, then

$$\min_b D_{\mathrm{KL}}\big(P^S_{\beta,g} \,\|\, P^S_{\beta+b,g+tv}\big) \le \frac{t^2\sigma^2}{8}.$$

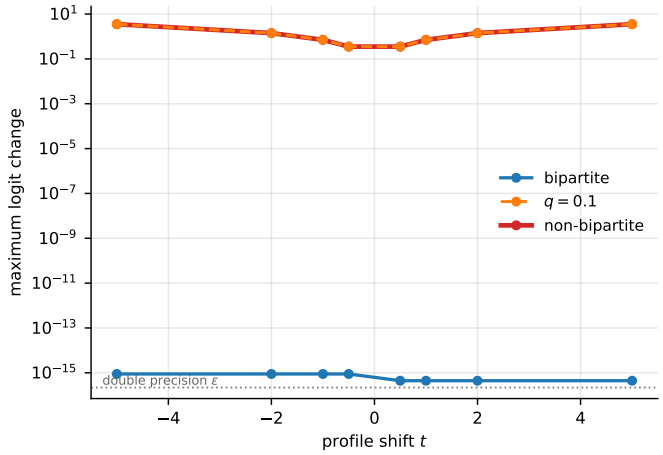


Figure 6: A profile difference can be completely invisible to the observed tasks. We shift 50 agent profiles by $g \mapsto g + t\bar{s}$, with $\bar{s} = s/\sqrt{8}$ the unit bipartition vector, and report the largest change in any logit over 12,000 tasks. In the bipartite design ($q = 0$), the change remains at numerical precision for every tested value of t. For any $q > 0$, at least one task breaks the additional ambiguity, so the maximum change is nonzero. The $q = 0.1$ and $q = 1$ curves overlap because the maximum is attained on a single ambiguity-breaking task and equals $2|t|/\sqrt{8}$ whenever such a task is present.

Thus the null space determines which profile differences cannot be measured, while the singular spectrum determines how strongly the observed tasks distinguish nearby profiles.

The lemma has a simple interpretation. The responses identify a profile only up to directions in $\ker Z_S^\circ$. Regularization in Eq. 3 may choose one representative along such a direction, but that choice is supplied by the regularizer rather than by information in the observed responses.

B.2 Pairwise demands

Suppose every demand is $z_{uv} = e_u + e_v$. The demand matrix is then the unsigned edge-vertex incidence matrix of the skill co-demand graph. Since

$$Z\mathbf{1} = 2\mathbf{1},$$

a common shift of all skill coordinates is absorbed by the global ability intercept for every pairwise design.

Now suppose $Z^\circ x = 0$. Then $Zx = a\mathbf{1}$ for some scalar a. Define

$$y = x - \frac{a}{2}\mathbf{1}.$$

For every edge (u, v),

$$y_u + y_v = 0.$$

If the connected graph is non-bipartite, these equalities imply $y = 0$, so the only invisible profile change is the common shift $\mathbf{1}$ already absorbed by the intercept. If the graph is bipartite with parts U, V, they instead allow

$$y = ts, \qquad s_k = \begin{cases} +1, & k \in U, \\ -1, & k \in V, \end{cases}$$

giving one additional invisible relative-skill contrast. This gives the pairwise statement of Section 4.

The exact ambiguity can also be seen directly in the predicted logits. Figure 6 shifts profiles along this additional null direction while leaving the task set fixed.

Thus bipartiteness creates one additional ambiguity beyond the common shift already absorbed by the intercept. Figure 2 shows the consequence for profile estimation in the controlled experiment of Appendix B.3.

B.3 CONTROLLED MEASUREMENT EXPERIMENT

Procedural environment and agents. The controlled world contains eight executable operations: ADD, MUL, MAX, MIN, PARITY, THRESHOLD, LOOKUP, and BOOL. A task is a short program that composes one to three operations over discrete inputs. Arithmetic outputs lie in $\{0, \ldots, 11\}$ and Boolean outputs use two additional classes, giving a 14-class answer space. Tasks also contain unused operand and parameter slots, so the operation sequence cannot be inferred from a trivial surface field.

The agents are 50 independently trained MLP classifiers, not hand-specified ability vectors. Each receives supervised input–answer examples generated by the same procedural world and is trained by SGD. The population contains 24 primitive specialists, 10 generalists, eight composition specialists, and eight perturbed agents. Specialists place 70% of their training mass on tasks using their preferred operations, 20% on generic single-operation tasks, and 10% on the full task distribution. Generalists use balanced task mixtures and mainly differ in training-set size. Composition specialists preferentially see particular pairs or triples of operations. Perturbed agents vary training-set size, number of epochs, model width, and label noise. Their held-out accuracies range from 0.253 to 0.659, and their mean pairwise behavioral correlation is 0.377, so the population contains graded and overlapping differences rather than disjoint hand-coded experts. Every binary outcome used below is obtained by running a trained agent and scoring whether its predicted answer is correct.

Measurement design. The measurement experiment reuses this trained population unchanged and varies only the evaluation tasks. Here every evaluation task contains exactly two operations. The bipartite condition uses the 3-cube over the eight operations, while the matched non-bipartite condition uses a Möbius ladder. Both endpoint designs contain 12 operation pairs, 12,000 tasks, and exactly 3,000 appearances of every operation.

We interpolate between the designs with $q \in \{0, .01, .02, .05, .1, .25, .5, 1\}$ while keeping the total task count and marginal operation frequencies fixed. At $q = 0$, the raw demand matrix Z has rank 7 and the analytic bipartition vector satisfies $\|Zs\|_2 = 0$. For every $q > 0$, the raw matrix has rank 8. Since every pairwise task also admits the common profile shift absorbed by β, the relevant distinction is that $q = 0$ has one additional relative-profile null direction, whereas $q > 0$ does not. Only 20 of 12,000 tasks are contrast breaking at $q = .01$.

Finite-sample protocol. We first fit an eight-dimensional reference profile for each of the 50 trained agents from a separate evaluation pool containing all 28 operation pairs. To isolate the statistical consequence of the measurement design under a correctly specified response model, these reference profiles define logistic success probabilities on the controlled tasks, from which we draw binary responses and refit each profile from support sets of 32 to 1,024 observations. At $q = 0$, the error along the hidden direction is 0.913 at every support size, while the error along the other directions falls from 1.401 to 0.145. With the complete 12,000-task design, the blind-direction error has zero sampling variance because the likelihood contains no information in that direction.

We also repeat the profile fitting directly on the actual correct/incorrect outputs of the trained MLP agents rather than on draws from the fitted logistic model. The same additional blind direction at $q = 0$ remains exactly unchanged across the six support budgets.

The finite-sample experiment separates the unmeasurable direction from the remaining profile directions. Figure 7 shows how both errors change as the number of observed responses increases.

At $q = 0$, increasing the support size by $32\times$ does not change the error along the blind direction, while the remaining profile error falls sharply. The limitation therefore comes from the task design rather than from a lack of responses. Once the ambiguity is broken, the remaining uncertainty decreases with the amount of information supplied by the contrast-breaking tasks.

C DATASETS

EmbedLLM. EmbedLLM (Zhuang et al., 2024) provides binary correctness for 112 models and $35{,}673$ unique prompts from ten public benchmarks and 80 fine-grained variants. Following Yao et al. (2026) we use prompt-level splits of $29{,}673$, $3{,}000$, and $3{,}000$ prompts, totaling 4.59M ob-

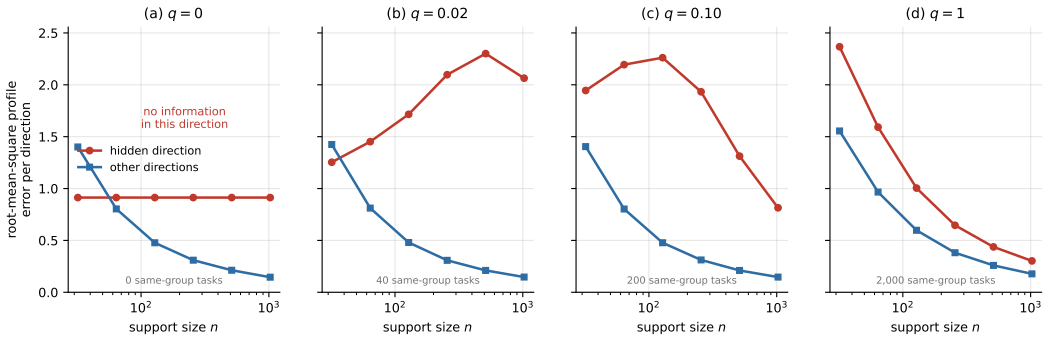


Figure 7: More observations cannot recover a profile direction that the task set does not measure. At $q = 0$, error along the additional unmeasurable direction remains 0.913 from 32 to $1{,}024$ observed responses, while error orthogonal to that direction falls by $9.6\times$. When tasks that break the ambiguity are introduced, the previously hidden direction becomes measurable and its error eventually decreases with additional observations. With only a few informative tasks, the newly identified direction can initially have high variance.

Table 2: **Dataset statistics.** Splits are prompt-level, so no prompt appears in two splits. The positive rate is the fraction of successful (agent, task) outcomes.

Dataset	Agents	Domains	Prompts (train/val/test)	Interactions	Pos. rate
EmbedLLM	112	10	29,673 / 3,000 / 3,000	4,586,288	0.370
RouterBench (0-shot)	11	14	29,197 / 3,649 / 3,651	401,467	0.551
MMLU-Pro LoRA experts	16	14	9,626 / 1,203 / 1,203	192,512	0.375

served rows. The row count exceeds the full grid because 12.9% of shipped rows rescore a question under another harness configuration, mostly for GPQA and GSM8K, and 1.9% of distinct agent-prompt cells receive conflicting repeated labels. We keep all shipped rows and give every method the identical row set. Neither a prompt nor an agent-prompt pair crosses splits. The separate recent-agent benchmark of Yao et al. (2026) provides $23{,}712$ training and $2{,}964$ test prompts per agent, so 1% support is 237 responses. In the held-out-pool protocol, 1% of an agent's training rows, rescored rows included, is 341 responses.

RouterBench. RouterBench (Hu et al., 2024) provides $36{,}497$ prompts from 14 task families and graded performance for 11 models. We convert each pair to one row with $y = \mathbf{1}[\text{performance} \geq 0.5]$ and use fixed prompt-level $80/10/10$ splits. Scores are already binary for 78.9% of rows. We preserve but do not use per-query cost, since we study the benchmark as a second success matrix.

MMLU-Pro experts. Ten LoRA experts are fine-tuned on the MMLU-Pro training questions of a mixture of two or three of the 14 domains (Table 3), with rank 32, $\alpha = 64$, dropout 0.05, learning rate 2×10^{-4}, ten epochs, effective batch size 16, and chain-of-thought supervision. The pool adds the unadapted base model and five chain-of-thought inference variants, which vary inference behavior at fixed weights. Every agent is evaluated on all $12{,}032$ prompts, giving $192{,}512$ interactions and prompt-level splits of $9{,}626$, $1{,}203$, and $1{,}203$.

D Baselines and Reproduction

Every learned baseline is trained with binary cross-entropy on the same prompt-level splits and the same frozen embeddings, so a comparison isolates the predictor rather than the representation.

Prediction rules. The majority controls predict $\mathbf{1}[\hat{p} \geq 0.5]$ from the training success rate computed per agent or per domain. A domain-by-agent lookup, which predicts from agent i's training success rate within the benchmark domain of task j and therefore consumes the benchmark label as side information, reaches 0.741, 0.719, and 0.698 on EmbedLLM, RouterBench, and MMLU-Pro; it is not label-free, so we do not list it among the methods in Table 1.

Table 3: **LoRA expert training mixtures.** Every domain appears in exactly two mixtures and every mixture is unique.

Expert	Training domains
math+bio	math, biology
math+hist+biz	math, history, business
phys+health+phil	physics, health, philosophy
phys+econ+other	physics, economics, other
chem+psych+law	chemistry, psychology, law
chem+cs+hist	chemistry, computer science, history
eng+bio+biz	engineering, biology, business
eng+psych+phil	engineering, psychology, philosophy
cs+health+econ	computer science, health, economics
law+other	law, other

The kNN control fits one classifier per agent over the frozen embeddings and predicts the majority outcome among the 131 nearest training tasks. Matrix factorization scores $\mathbf{w}^\top(\mathbf{p}_i \odot \mathbf{W}\mathbf{x}_j) + b$, and the linear probe scores $\mathbf{w}^\top[\mathbf{x}_j; \mathbf{m}_i] + b$. The three IRT baselines and our head all instantiate $\mathrm{logit}(i, j) = \mathbf{a}_j^\top \boldsymbol{\theta}_i - b_j$ with item parameters amortized from $\mathbf{x}_j$. IrtNet reads them from a sparsely routed mixture of 39 experts (Chen et al., 2025). JE-IRT constrains the discrimination to the unit sphere and sets the difficulty to the norm of the same vector (Yao et al., 2026). IRT-Router produces ability from an embedding of the agent's name string and the item parameters from linear maps of $\mathbf{x}_j$ (Song et al., 2025). Our head replaces the dense discrimination with the sparse nonnegative demand vector over shared axes, replaces the scalar item difficulty with $(\mathbf{z}_j^{\mathrm{sk}})^\top \boldsymbol{\tau}$, reads ability off the decoder, and adds a per-agent intercept. None of the three substitutions enlarges the model class.

Settings and reproduction. JE-IRT uses joint dimension 256 with the original per-encoder learning rates, dropout 0.5, weight decay 0.01, batch size 512, and 100 epochs with best-validation checkpointing, and on the controlled pool we also tried the original small joint dimension and report the better validation variant ($d = 32$). Matrix factorization uses embedding dimension 232, noise 0.05, batch size 2048, learning rate 10^{-4}, and 50 epochs. IrtNet uses the original configuration with 232-dimensional agent embeddings, 39 experts, dropout 0.5, embedding noise 0.05, and early-stopping patience 3. The linear probe concatenates the frozen prompt embedding with a learned 32-dimensional agent embedding. With the original all-mpnet-base-v2 encoder our ports reproduce the published EmbedLLM results closely, since JE-IRT reaches 0.745 against a published 0.746 and matrix factorization reaches 0.734 against 0.741, matching the gap already seen when the original code is run under best-validation checkpointing. Our kNN and IrtNet ports exceed their published numbers, at 0.742 against 0.715 and 0.746 against 0.722, which we attribute to our fixed prompt-level splits and precomputed embeddings.

Template artifacts. We rebuild EmbedLLM after stripping shared prompt templates and regenerate both encoders' embeddings. Every learned method loses at most one accuracy point and the leading methods are unchanged (Table 4), so template overlap does not explain the predictive comparison.

Table 4: **Shared prompt templates do not explain the predictive comparison.** Test accuracy on EmbedLLM and on its template-stripped rebuild. All methods drop by at most one point, and the leading methods and overall conclusion are unchanged.

	mpnet		ModernBERT-Large	
Method	original	cleaned	original	cleaned
kNN	0.742	0.737	0.741	0.735
Matrix factorization	0.734	0.724	0.732	0.731
IrtNet	0.746	0.745	0.748	0.744
Linear probe	0.729	0.721	0.737	0.730
JE-IRT	0.745	0.740	0.749	0.744
SkillBasis (ours)	0.744	0.740	0.746	0.739

Table 5: **Every recovery statistic clearly exceeds its shuffled-control null, on both encoders.** Selectivity AUC asks whether specialists trained on a skill's dominant domain receive higher activation on that skill than other agents. η^2 and normalized mutual information (NMI) instead measure how strongly each task's domain determines which skill is most active for it. AUC is compared against a null that reshuffles which domain each specialist was trained on, while η^2 and NMI are compared against a null that reshuffles each task's domain label. Nearly every skill remains active on both encoders. The domain-informed upper reference of 1.000 is the score a method with perfect knowledge of the true domain assignments would achieve here, not a general ceiling on achievable prediction accuracy.

	Observed		Shuffled null (mean / 95%)	
Metric	M-BERT	mpnet	M-BERT	mpnet
Selectivity AUC, within skill	0.943	0.950	0.499 / 0.644	0.495 / 0.645
domain-informed upper reference	1.000	1.000	—	—
skills above chance	14/14	14/15	—	—
Demand domain selectivity, mean η^2	0.333	0.407	0.001 / 0.001	0.001 / 0.001
NMI (dominant skill vs. domain)	0.301	0.407	0.003 / 0.003	0.003 / 0.004
Live skills	14/14	15/16		

Selectivity AUC is reported for the selected canonical checkpoint; across six runs of the same ModernBERT-Large configuration it is 0.88 ± 0.06.

E Recovery metrics, controls, and annotation

Selectivity AUC. Let d_k be the domain on which skill k has the highest mean demand and let $\mathcal{A}$ be the 14 agents with a known mixture, excluding the base model and its chain-of-thought variant. For each skill we form labels $y_i = \mathbf{1}[d_k \in \mathrm{mix}(i)]$ over $\mathcal{A}$, score agents by $c_i[k]$, and average the per-skill areas under the ROC curve. Averaging within skills rather than pooling all pairs keeps every comparison inside one skill, so neither domain difficulty nor the scale of $\mathbf{c}$ across skills can enter. The two forms differ here, since pooling gives 0.704 where averaging gives 0.943. Selectivity AUC is unaffected by any per-skill shift, because it compares agents strictly within a skill.

Permutation nulls. We permute which of the ten mixtures is attributed to which specialist and recompute the statistic, for $B = 2{,}000$ draws. Permuting the attribution preserves the profiles exactly, along with the number of experts per domain and the number of positives per skill, so the null isolates the correspondence between profile and mixture. Reported p-values use the bias-corrected form $(\#\{b : \mathrm{AUC}^{(b)} \geq \mathrm{AUC}^{\mathrm{obs}}\} + 1)/(B + 1)$ and are bounded below by 0.0005, so a value of 0.0005 means no permutation matched the observed statistic. The demand metrics η^2 and NMI are nulled by permuting prompt domain labels, which breaks the correspondence while preserving both marginals. Table 5 reports every statistic against its null.

Topic and resolution controls. A permutation null gives a chance reference and not a topic-only reference, so we also cluster the same $12{,}032$ unnormalized task embeddings with k-means at each model's skill count, using ten initializations and five seeds (Table 6). The anchor is weaker than the learned demands under ModernBERT-Large and stronger under all-mpnet-base-v2. For resolution, we compare Spearman correlations between the agent rankings induced by two skills that share a dominant domain, which gives 0.378 and 0.418 under the two encoders, against 0.677 and 0.646 for the same skill matched across seeds by task demand. ModernBERT-Large also assigns three distinct coordinates to mathematics that receive different labels and rank agents differently. The anchor measures topical organization rather than behavioral usefulness, and SkillBasis is not expected to maximize domain NMI.

Labeling and the held-out identification test. Each live skill is labeled by gpt-5.4-mini from ten maximally activating prompts, four silent prompts nearest the activating centroid, and three random silent prompts, where the close negatives force the label to name an operation rather than a topic. The judge is blind to agent identities and dataset domains and returns a label, a description, a discriminating criterion, and a coherence rating. A context-separated invocation of the same model then sees only the label and description and classifies sixteen held-out prompts, namely eight activating, four close negatives, and four random silent prompts. Following Heap et al. (2026) we treat

Table 6: **Recovery controls on the controlled MMLU-Pro pool.** The topic anchor is k-means on the same unnormalized task embeddings, averaged over five seeds with ten initializations each. "Within-domain" is the mean Spearman correlation between agent rankings induced by two learned skills with the same dominant domain. "Cross-seed match" compares skills matched across seeds using task demands. A lower within-domain correlation indicates that same-domain skills need not impose the same agent ordering.

Measure	ModernBERT-Large	all-mpnet-base-v2
SkillBasis NMI	0.301	0.407
Topic-anchor NMI	0.241 ± 0.001	0.526 ± 0.001
Within-domain rank correlation	0.378	0.418
Cross-seed matched correlation	0.677	0.646
Difference [95% CI]	0.299 $[0.176, 0.421]$	0.229 $[0.047, 0.416]$
Rank-sum p	$< 10^{-4}$	0.023

the result as semantic evidence rather than validation, since the same model family generates and scores the labels. Table 7 gives the inventory of the controlled model.

Table 7: **Skill inventory of the canonical MMLU-Pro model (ModernBERT-Large).** Activation frequency is the fraction of tasks on which a skill is active. Intrusion accuracy is the intrusion-test score of Appendix E. Coherence is the labeling judge's self-rated cluster coherence (1 to 5).

#	Skill label	Activation freq.	Coherence	Intrusion acc.
1	Clinical clue-to-cause inference	0.39	4	1.00
6	Scientific dimensional analysis	0.37	5	0.81
2	Inferring implied claims from texts	0.35	4	0.75
8	Chained engineering formula solving	0.34	5	0.88
4	Discrete fact retrieval	0.33	4	0.62
10	Direct formula substitution	0.33	4	0.62
9	Technical concept definition retrieval	0.32	5	1.00
13	Clinical vignette rule matching	0.28	5	1.00
11	Algebraic transformation and solving	0.27	5	0.94
7	Mechanistic concept matching	0.26	4	0.81
12	Symbolic equation manipulation	0.25	4	0.81
0	Legal rule-to-fact application	0.20	5	1.00
5	Legal doctrine application	0.19	5	0.88
3	Exact fact recall	0.18	4	0.50
	Mean		4.50	0.83

F ONBOARDING DETAILS

For a new agent with responses $\{(j, y_j)\}_{j \in S}$, the design row is $\mathbf{X}_j = [\,1,\ (\mathbf{z}_j^{\mathrm{sk}})^\top\,]$ and the offset is $\psi_j = -(\mathbf{z}_j^{\mathrm{sk}})^\top \boldsymbol{\tau}$, so $\eta_j = \mathbf{X}_j \boldsymbol{\theta} + \psi_j$ with $\boldsymbol{\theta} = [\beta; \mathbf{c}]$. The first estimator minimizes the sum of binary cross-entropies plus $(\lambda/2)\|\mathbf{c}\|_2^2$ with L-BFGS, leaving the intercept unregularized. The second solves the same problem by iteratively reweighted least squares (McCullagh & Nelder, 1989), using damped Newton steps, backtracking, and a $10^{-6}\mathbf{I}$ stabilizer, and gives nearly identical results. Both fit $\mathbf{c}$ directly and can return a profile outside the span realized by the training-agent embeddings. In the sense of Section 4, no direction among the live skills is exactly hidden by the learned demands: for every dataset and encoder, the centered demands of the training tasks have full column rank on the live skills, and so does each random 1% support we drew for the held-out EmbedLLM agents. With several hundred responses, random support selection matches the structured selection rules we tested, and demand-weighted selection can shift the support set's mean success rate, which miscalibrates β and lowers accuracy.

Table 8 reports both regimes and both encoders. In the held-out-pool regime every displayed task-side model is retrained without the same ten target agents. We use EmbedLLM for holdout experiments because its 112 agents allow six or ten to be excluded without redefining the shared task basis,

Table 8: **Onboarding with a frozen task representation.** *Recent models* are six agents from the held-out new-model benchmark of Yao et al. (2026). *Held-out pool* contains ten EmbedLLM agents excluded before learning each method's task-side representation. Both regimes report ModernBERT-Large and all-mpnet-base-v2. "All" is accuracy when the target agents are included during training, not a support fraction or an upper bound. Support-fraction entries use direct L-BFGS fitting. Values are test accuracy (%), mean±standard deviation across target agents.

Setting	Method	1%	10%	50%	100%	All
Recent models ModernBERT-Large	MF (Zhuang et al., 2024)	$57.1_{\pm 0.6}$	$60.5_{\pm 0.5}$	$61.4_{\pm 0.4}$	$61.7_{\pm 0.7}$	–
	IrtNet (Chen et al., 2025)	$57.2_{\pm 1.1}$	$59.9_{\pm 0.8}$	$61.2_{\pm 0.8}$	$61.7_{\pm 0.8}$	–
	JE-IRT (Yao et al., 2026)	$56.1_{\pm 1.1}$	$59.8_{\pm 0.7}$	$62.0_{\pm 0.5}$	$62.4_{\pm 0.6}$	$64.7_{\pm 1.0}$
	SkillBasis (ours)	$56.2_{\pm 1.0}$	$57.4_{\pm 1.1}$	$57.5_{\pm 1.2}$	$57.5_{\pm 1.2}$	$60.3_{\pm 0.9}$
Recent models all-mpnet-base-v2	MF (Zhuang et al., 2024)	$57.9_{\pm 0.6}$	$60.1_{\pm 1.0}$	$61.0_{\pm 0.8}$	$61.4_{\pm 0.8}$	–
	IrtNet (Chen et al., 2025)	$57.3_{\pm 1.1}$	$60.1_{\pm 0.8}$	$61.4_{\pm 0.6}$	$61.9_{\pm 0.5}$	–
	JE-IRT (Yao et al., 2026)	$57.2_{\pm 0.8}$	$59.8_{\pm 0.8}$	$61.1_{\pm 0.6}$	$61.5_{\pm 0.5}$	$63.0_{\pm 1.2}$
	SkillBasis (ours)	$57.1_{\pm 0.9}$	$59.4_{\pm 0.9}$	$60.1_{\pm 0.8}$	$60.2_{\pm 0.9}$	$61.9_{\pm 0.9}$
Held-out pool ModernBERT-Large	MF (Zhuang et al., 2024)	$73.1_{\pm 2.2}$	$73.6_{\pm 2.2}$	$73.7_{\pm 2.2}$	$73.8_{\pm 2.3}$	$73.0_{\pm 2.3}$
	IrtNet (Chen et al., 2025)	$74.4_{\pm 2.0}$	$74.6_{\pm 2.0}$	$74.8_{\pm 1.9}$	$74.8_{\pm 1.9}$	$74.4_{\pm 2.0}$
	JE-IRT (Yao et al., 2026)	$74.2_{\pm 2.1}$	$74.4_{\pm 2.0}$	$74.5_{\pm 2.0}$	$74.4_{\pm 2.0}$	$74.5_{\pm 2.0}$
	SkillBasis (ours)	$72.8_{\pm 2.4}$	$73.1_{\pm 2.5}$	$73.2_{\pm 2.5}$	$73.1_{\pm 2.5}$	$74.3_{\pm 2.1}$
Held-out pool all-mpnet-base-v2	MF (Zhuang et al., 2024)	$72.2_{\pm 2.3}$	$72.6_{\pm 2.4}$	$72.8_{\pm 2.3}$	$72.9_{\pm 2.2}$	$72.4_{\pm 2.3}$
	IrtNet (Chen et al., 2025)	$74.1_{\pm 2.1}$	$74.3_{\pm 2.0}$	$74.3_{\pm 2.0}$	$74.4_{\pm 2.0}$	$74.1_{\pm 2.1}$
	JE-IRT (Yao et al., 2026)	$73.6_{\pm 2.1}$	$73.8_{\pm 2.2}$	$73.8_{\pm 2.2}$	$73.8_{\pm 2.2}$	$73.9_{\pm 2.2}$
	SkillBasis (ours)	$73.3_{\pm 2.2}$	$73.6_{\pm 2.2}$	$73.6_{\pm 2.2}$	$73.6_{\pm 2.2}$	$73.7_{\pm 2.2}$

whereas RouterBench has 11 agents and each planted MMLU-Pro domain is covered by only two specialists.

G Adaptation protocol and full results

Alignment margin. Let $\bar{\mathbf{z}}_D^{\mathrm{sk}}$ be the mean demand over the new training split of mixture D and let $\Delta\mathbf{c}_{i,D} = \hat{\mathbf{c}}_{i,D}^{\mathrm{post}} - \mathbf{c}_i^{\mathrm{pre}}$ be the profile change of source expert i. Because MMLU-Pro mixtures share broad demands, a raw cosine would be positive against almost any mixture, so for the set $\mathcal{G}$ of ten mixtures we report

$$A_{i,D} = \cos\big(\Delta\mathbf{c}_{i,D}, \bar{\mathbf{z}}_D^{\mathrm{sk}}\big) - \frac{1}{|\mathcal{G}|-1} \sum_{D' \in \mathcal{G}\setminus\{D\}} \cos\big(\Delta\mathbf{c}_{i,D}, \bar{\mathbf{z}}_{D'}^{\mathrm{sk}}\big)\,. \tag{10}$$

Design and training. Every mixture serves exactly once as the new training mixture in each of the three conditions. Each adaptation initializes from the corresponding existing adapter and reuses the expert-generation configuration without change, with a deterministic $80/20$ train-validation split at seed 42, and the mean demand is computed only over the 80% split actually used. Each adapted model and its unadapted source are evaluated on all $12{,}032$ questions with the same evaluator, and both endpoints are fitted from their own response vectors with the same L-BFGS estimator at $\lambda = 1$.

Why the estimator is matched. Subtracting a stored jointly trained source profile from a freshly fitted adapted profile would confound adaptation with differences between the two estimation procedures. In particular, shrinkage induced by fitting could introduce a component proportional to $-\mathbf{c}_i^{\mathrm{pre}}$ and thereby reproduce the observed ordering of alignment margins across conditions even without adaptation. We therefore fit both endpoints using the same estimator; Table 9 retains the mixed comparison only as a sensitivity check. The re-onboarded source profiles are 3.2% larger in norm and have cosine similarity 0.976 with the stored profiles. The resulting estimator offset is nearly orthogonal to $-\mathbf{c}_i^{\mathrm{pre}}$ (cosine -0.023). In the offset-only placebo, the mean margin is -0.001 and only 13 of 30 margins are positive, indicating that estimator mismatch alone does not produce systematic positive alignment.

Behavioral groups. With $\mathcal{S}_i$ the source domains, $\mathcal{D}$ the adaptation domains, and $\mathcal{U}$ all domains, each domain falls in exactly one of $\mathcal{S}_i \cap \mathcal{D}$, $\mathcal{D} \setminus \mathcal{S}_i$, $\mathcal{S}_i \setminus \mathcal{D}$, and $\mathcal{U} \setminus (\mathcal{S}_i \cup \mathcal{D})$. Accuracy changes

Table 9: **The alignment margin holds under two different estimators but vanishes when no adaptation is applied.** The matched estimator re-fits both the pre- and post-adaptation profiles with the same L-BFGS procedure and is the primary analysis. The mixed estimator instead subtracts the stored joint-training profile from the fitted post-adaptation profile. The offset-only placebo applies the same estimator difference without any adaptation, as a control. Intervals are source-bootstrap 95% intervals, and the final row gives the 95th percentile under 20,000 mixture permutations, including the positive-margin count.

Analysis	Positive	Overall	Same	Overlap	Disjoint
Matched estimator	29/30	0.276	0.226 [0.112, 0.335]	0.277 [0.189, 0.375]	0.326 [0.254, 0.408]
Mixed estimator	30/30	0.260	0.194 [0.118, 0.272]	0.235 [0.163, 0.306]	0.349 [0.266, 0.446]
Offset-only placebo	13/30	−0.001	0.036 [0.006, 0.068]	−0.055 [−0.092, −0.020]	0.017 [−0.058, 0.118]
Permutation 95th pct.	17/30	0.049	0.080	0.087	0.097

Table 10: **The alignment margin is positive in nearly every run.** A is the primary matched-estimator alignment margin from Eq. 10, with both endpoints fitted using the same L-BFGS procedure. *Both stages*, *new only*, *source only*, and *neither* partition domains by their occurrence in the source and adaptation mixtures, and accuracy entries are macro-averaged changes in percentage points. Dashes mark structurally empty groups.

Condition	Source expert	Adaptation mixture	A	Both stages	New only	Source only	Neither
Same	eng+psych+phil	eng+psych+phil	0.027	+7.4	–	–	+0.8
	phys+econ+other	phys+econ+other	0.087	+5.2	–	–	−0.2
	math+bio	math+bio	0.278	+15.5	–	–	+0.8
	math+hist+biz	math+hist+biz	0.324	+14.4	–	–	−1.1
	phys+health+phil	phys+health+phil	0.196	+12.0	–	–	−0.8
	chem+psych+law	chem+psych+law	0.396	+15.8	–	–	−1.4
	chem+cs+hist	chem+cs+hist	0.402	+22.0	–	–	−1.7
	eng+bio+biz	eng+bio+biz	0.165	+16.7	–	–	−0.5
	cs+health+econ	cs+health+econ	−0.114	−4.4	–	–	+1.4
	law+other	law+other	0.496	+12.0	–	–	−1.0
Overlap	eng+psych+phil	eng+bio+biz	0.206	+17.8	+26.4	−5.0	+2.0
	phys+econ+other	phys+health+phil	0.261	+21.2	+50.0	−6.9	−1.2
	math+bio	math+hist+biz	0.381	+31.8	+36.7	−2.6	+2.9
	math+hist+biz	math+bio	0.042	+2.2	+15.9	−10.4	+0.7
	phys+health+phil	phys+econ+other	0.162	+10.8	+21.5	−8.1	+0.8
	chem+psych+law	law+other	0.617	+11.9	+20.8	−4.3	−0.7
	chem+cs+hist	cs+health+econ	0.376	+18.5	+42.6	−10.9	+1.2
	eng+bio+biz	eng+psych+phil	0.172	+14.9	+23.6	−3.7	+1.4
	cs+health+econ	chem+cs+hist	0.338	−5.9	+13.5	−12.2	+2.5
	law+other	chem+psych+law	0.213	+16.1	+20.6	−9.2	+4.7
Disjoint	eng+psych+phil	law+other	0.598	–	+56.1	−9.1	−1.4
	phys+econ+other	chem+psych+law	0.306	–	+21.4	−11.8	+2.0
	math+bio	phys+health+phil	0.351	–	+50.7	−6.8	+3.4
	math+hist+biz	phys+econ+other	0.213	–	+44.8	−20.2	+3.1
	phys+health+phil	math+bio	0.247	–	+11.5	−17.9	+0.3
	chem+psych+law	math+hist+biz	0.438	–	+35.3	−12.4	+0.1
	chem+cs+hist	eng+bio+biz	0.205	–	+27.8	−12.3	+1.6
	eng+bio+biz	chem+cs+hist	0.431	–	+52.0	−9.2	−1.2
	cs+health+econ	eng+psych+phil	0.176	–	+22.2	−23.8	+3.6
	law+other	cs+health+econ	0.291	–	+47.2	−15.7	+2.9

are macro-averaged over domains within each nonempty group and then over the ten source experts, so every source expert carries equal weight. Reported intervals resample the ten source experts and therefore describe variation across the pool rather than across adaptation seeds, since every adaptation uses one fixed seed.

H PREDICTIVE INFORMATION BEYOND DOMAINS

Specialization alignment alone does not establish that learned demands provide information beyond subject identity. We therefore give both predictors explicit domain effects and ask whether adding skills improves prediction further. This compares domains against domains plus skills, rather than treating them as alternative representations.

We use the 30 adapted agents from Appendix G and freeze the canonical ModernBERT-Large basis. Let h_j denote the domain of question j. With frozen demands $\mathbf{z}_j^{\mathrm{sk}}$ and shared offset $\psi_j = -(\mathbf{z}_j^{\mathrm{sk}})^\top \boldsymbol{\tau}$,

we fit

$$\eta^{D}_{i,j} = \psi_j + \beta^{D}_i + \alpha^{D}_{i,h_j},$$
$$\eta^{D+S}_{i,j} = \psi_j + \beta^{D+S}_i + \alpha^{D+S}_{i,h_j} + (\mathbf{z}^{\mathrm{sk}}_j)^\top \mathbf{c}_i.$$

Here D denotes domain effects with the shared frozen offset, and S denotes the added skill contribution. Each predictor independently fits its own agent intercept and 14 domain coefficients. $D + S$ additionally fits the agent's skill coefficients. Thus, domain effects can capture an agent's overall strength in mathematics, while learned demands can distinguish questions within that domain.

Both predictors use summed logistic loss with an L_2 penalty on domain and skill coefficients. We fix $\lambda = 1$, matching the adaptation-profile estimator, and leave intercepts unpenalized. Five shared, domain-balanced folds over all 12,032 questions keep duplicate clusters together. We fit on four folds and predict the fifth, then compute each agent's log loss from its out-of-fold predictions.

Adding skills reduces log loss by 3.7% on average, with improvement for every adapted agent. Mean relative reductions are 3.56% for same-mixture, 3.88% for overlapping, and 3.54% for disjoint adaptations (Figure 4b). These percentages are computed within each agent and then averaged. The aggregate gain remains positive with weaker and stronger regularization ($\lambda = 0.1$ and 10) and all five additional checkpoints.

The result supports added predictive information in the learned demands after accounting for domain effects. Cross-validation evaluates profile fitting in a frozen basis, not generalization of the entire basis-learning procedure.

I Ablations and selection diagnostics

Table 12 removes one architectural component at a time from the canonical EmbedLLM model, with accuracies averaged over three seeds. Removing the partition leaves accuracy unchanged and spreads the prediction across 130 active directions out of 4096. Removing decoder coupling leaves accuracy and task-side label identification unchanged and weakens agent-side recovery on the controlled pool, from 0.878 ± 0.059 to 0.820 ± 0.024. Freezing the dictionary after reconstruction warmup costs about five accuracy points and lowers mean label identification from 0.87 to 0.70, with two frozen features at or below chance and one catch-all feature firing on half the tasks.

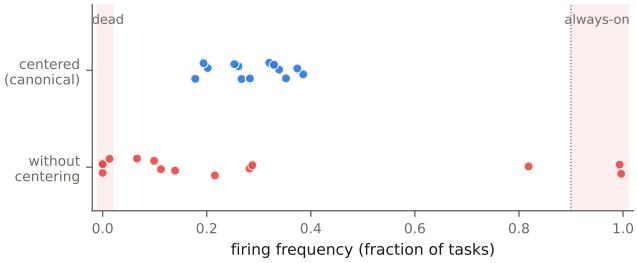


Figure 8: **Centering keeps skills from going dead or firing everywhere.** With centering all 14 skills of the controlled model fire on 18 to 39 percent of tasks. Without it, three become dead and two fire on nearly every task, at the same accuracy and within 0.003 validation cross-entropy.

Skill count. Figure 10 varies C with the remaining settings fixed. Accuracy stays inside the canonical seed-variability band from $C = 16$ through 64 and drops about one point at $C = 128$, where a third of the directions are dead. The dead fraction is not monotonic, at 19% for $C = 32$ and 11% for $C = 64$, so the runs do not identify a unique optimum and instead support a conservative choice of a small partition.

Shift invariance of the implemented checks. The three selection checks are $\mathrm{corr}(\beta_i, \mathrm{acc}_i) > 0.95$, maximum firing frequency below 90%, and a mean raw profile norm above 0.5, with thresholds fixed before the recovery metrics were computed. The first two are invariant to the joint shift of profiles and thresholds and the third is not, so we also audit the better-posed replacement $\mathbb{E}_i \|\mathbf{c}_i -$

Table 11: **Decomposition ablations on the controlled MMLU-Pro pool.** Each ablation is evaluated over three training seeds, and the complete model over six. Predictive performance is nearly unchanged, while selectivity AUC and diagnostic outcomes deteriorate. "Runs passing" counts checkpoints satisfying all three diagnostics. Bold marks an explicitly failed diagnostic.

	Test acc	Val BCE	Select. AUC	$\mathrm{corr}(\beta, \mathrm{acc})$	Runs passing
Complete model (canonical)	0.685 ± 0.002	0.565	0.878 ± 0.059	0.995	4 of 6
No data-based β initialization	0.684 ± 0.002	0.568	0.821 ± 0.063	$\mathbf{0.721}$	0 of 3
No intercept ($\beta = 0$)	0.686 ± 0.001	0.567	0.824 ± 0.023	n/a	0 of 3
No centering	0.685 ± 0.001	0.567	0.833 ± 0.042	0.995	0 of 3

The correlation is undefined when β is fixed to zero.

Table 12: **Architecture ablations on EmbedLLM (ModernBERT-Large).** Each row retrains the canonical configuration with one component removed. Test accuracy is the mean over three seeds. Active skills / task is the number of active, nameable skills per task. Mean intrusion is reported where the skill partition remains annotatable. Validation BCE is from the reported seed.

	Test acc	Val BCE	Active skills / task	Mean intrusion
SkillBasis (canonical)	0.746 ± 0.001	0.525	2 of 16	0.87
No partition (all features predict)	0.744 ± 0.002	0.539	130 of 4096	–
No decoder coupling (free profiles)	0.744 ± 0.004	0.531	2 of 16	0.85
Unsupervised dictionary (frozen SAE)	0.693 ± 0.012	0.610	2 of 16	0.70

$\boldsymbol{\tau}\|_2$. The two forms agree on all 15 checkpoints of Table 11. They disagree on one canonical checkpoint outside that set, where the EmbedLLM ModernBERT-Large seed-1 run has raw norm 0.456 and gap norm 1.094, and the run stays in the three-seed average of Table 1 because prediction uncertainty is reported over all retrainings rather than over the subset a post-hoc check selects.

Retraining audit. Table 13 reports six replicates at fixed architecture and swept hyperparameters, where the four passing runs are the four highest-recovery runs. Across the wider set of 18 canonical and ablated checkpoints, passing runs average 0.913 ± 0.032 and failing runs 0.822 ± 0.041, although the comparison mixes deliberately altered configurations. We therefore present the audit as checkpoint-level evidence and not as validation of universal thresholds.

J HYPERPARAMETERS

Table 14 lists the search ranges and the fixed values, and Table 15 gives the selected configuration for each reported model. Centering and data-based intercept initialization are held fixed in every sweep. The controlled-pool grid deliberately includes $C = 14$, the number of MMLU-Pro domains, so the search had the chance to settle on one skill per domain. The selected model does not do so, since its 14 skills have only eight distinct dominant domains and six domains are dominant for none.

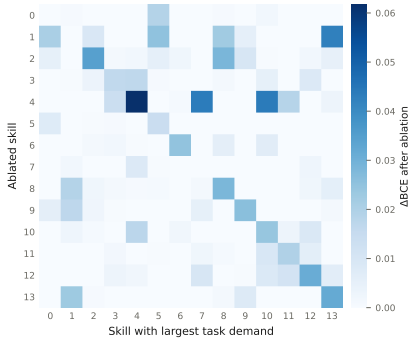


Figure 9: **Each skill matters most where it is demanded.** Zeroing a skill's demand raises test cross-entropy wherever the skill fires, and the mean effect on tasks where it is the dominant demand is 9.6 times the mean effect elsewhere.

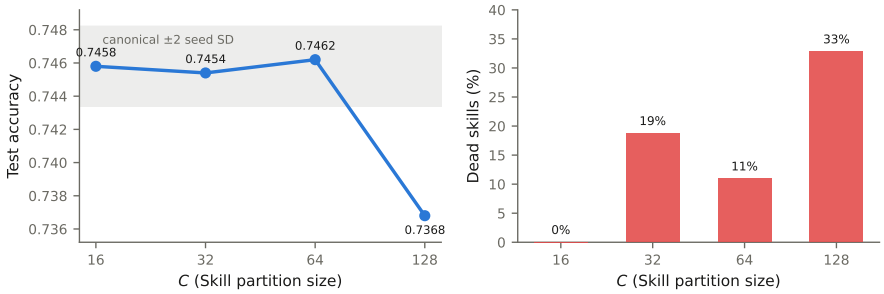


Figure 10: **Accuracy is flat across a wide range of skill counts and drops where dead directions become common.** Each setting is one run on EmbedLLM with ModernBERT-Large, and the gray band is the canonical $C = 16$ accuracy within two standard deviations across its training seeds.

Table 13: **The four runs that pass the diagnostics are the four runs with the highest selectivity AUC.** Six training replicates are evaluated, holding the architecture and swept hyperparameters fixed. Selectivity AUC measures how well a run's learned skills recover the planted specialization structure. The ability-sink and activation-frequency diagnostics pass in every row. "Fail: norm" means that the implemented raw-profile-norm diagnostic falls below its 0.5 threshold. This small audit establishes an association rather than validating a general decision rule.

Run	Raw profile norm	Selectivity AUC	Checks
1	0.703	0.943	Pass
2	0.801	0.934	Pass
3	0.603	0.912	Pass
4	0.684	0.861	Pass
5	0.497	0.844	Fail: norm
6	0.417	0.774	Fail: norm

Table 14: **Hyperparameter search spaces.** Every dimension is discrete. Each dataset-encoder pair is swept with roughly 40 Bayesian-search runs and Hyperband early termination on validation BCE, and the selected run is the lowest-loss one that passes the checks of Section 3. The EmbedLLM grid was refined once after the initial sweep concentrated at its lower learning-rate edge, adding 10^{-5}.

Parameter	EmbedLLM	RouterBench	MMLU-Pro experts
dictionary multiple	$\{1, 2, 4\}$	$\{1, 2\}$	$\{1, 2\}$
C skill partition	$\{16, 32, 64\}$	$\{8, 16, 32\}$	$\{14, 16, 28\}$
active skills per task	$\{2, 4, 8, 16\}$	$\{1, 2, 4\}$	$\{1, 2, 4\}$
active reconstruction features	$\{32, 64, 128\}$	$\{32, 64, 128\}$	$\{32, 64\}$
reconstruction warmup epochs	$\{0, 5, 10\}$	$\{0, 5, 10\}$	$\{5, 10\}$
λ_{rec}	$\{0.5, 2, 8, 30\}$	$\{0.5, 2, 8, 30\}$	$\{0.05, 0.2, 1, 3\}$
λ_{aux}		$\{0.1, 0.3, 1\}$	
λ_{orth}		$\{10^{-5}, 10^{-3}, 10^{-1}\}$	
λ_e (agent L_2)		$\{10^{-6}, 10^{-4}, 10^{-2}\}$	
learning rate ($\times 10^{-5}$)	$\{1, 3, 10, 30\}$	$\{3, 10, 30\}$	$\{5, 10, 20, 40\}$
batch size		$\{512, 1024\}$	
Fixed (decomposition design and schedule)			
input centering		on	
β init from data		on	
λ_β (ability L_2)		10^{-4}	
dead-feature threshold (steps)		200	
max epochs / patience		300 / 20	

Table 15: **Selected hyperparameters of the six canonical models.** Each is the lowest-validation-loss sweep configuration that passes the three selection checks of Section 3. Appendix I reports a retraining audit and the one disagreement between the implemented raw-norm check and its shift-invariant restatement.

	EmbedLLM		RouterBench		MMLU-Pro experts	
Parameter	mpnet	M-BERT	mpnet	M-BERT	mpnet	M-BERT
dictionary multiple	2	4	2	1	2	2
C	32	16	8	16	16	14
active skills	8	2	4	4	2	4
active recon. features	32	128	128	128	32	32
recon. warmup	5	10	10	0	5	5
λ_{rec}	30	8	0.5	2	1	0.05
λ_{aux}	0.3	1	0.1	0.1	0.1	0.3
λ_{orth}	10^{-3}	10^{-3}	0.1	0.1	10^{-3}	10^{-5}
λ_e	10^{-4}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-4}
learning rate ($\times 10^{-5}$)	3	3	3	3	20	10
batch size	1024	512	1024	512	1024	1024