FROM THE ZOO TO THE BULLPEN: SAMPLE-EFFICIENT LLM PROFILING VIA MULTI-CHANNEL MODEL EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Choosing among the hundreds of thousands of available large language models (LLMs) requires a thorough understanding of each candidate model's properties. In practice, each property is measured by its own benchmark suite, so profiling a model on many properties means paying for many separate evaluations. To address this, model embeddings compress each model into a fixed-length vector from which many properties can be predicted. Existing methods, such as EmbedLLM, build these vectors from correctness patterns and question features, and often need large probe batteries to do so. However, many behaviors that are important when choosing a model, such as its verbosity, safety and style, show up more in how it answers than in whether it is correct. We introduce BULLPEN (Behavioral Understanding of Large Language models via Profile ENcodings), a framework that extends EmbedLLM in three ways. First, BULLPEN organizes the downstream tasks that prior work uses and adds 20 task benchmarks we built on behavior, style and provenance. Second, it is the first framework to read a model's own answer text jointly with its correctness and the question text. Third, it encodes models from a few hundred questions instead of the full 24,118-question pool. Across 260 open-weight models, our answer-text encoder scores 0.48 ± 0.10 on the new tasks against 0.35 ± 0.10 for the best prior encoder. Our encoders that read answer text keep at least 91% of their full-pool skill from only 400 questions (1.3% of the inference compute of the full question pool), while a correctness-only baseline keeps 83% with the same 400 questions. Because these embeddings need only model outputs, they support behavioral auditing and low-cost profiling, and could extend to API-only models. Code and data: `https://github.com/wcquesera/bullpen`.

1 INTRODUCTION

Hugging Face now hosts over 300,000 text-to-text models (Patel et al., 2026), and more are released every week. Most of these models are derivatives of a few shared base models (via fine-tunes, merges, and quantizations), yet they differ widely in what they do well, how they answer and where they came from. To choose among these models, users can consult published benchmark results, read the metadata attached to each model, or evaluate candidate models themselves (Hendrycks et al., 2021a; Chiang et al., 2024; Liang et al., 2023; Cai et al., 2026). However, published results often do not exist for a given model, metadata is usually limited to a short description and coarse labels such as architecture, size and license, and running one's own benchmarks on many candidate models is costly. Public leaderboards also score only a fraction of models and are rarely rerun as new ones appear (e.g., only 42 of the 260 models in this paper have an LMArena rating). To address this, a profile of every model can be built from its own outputs, relating it to the other models.

Model embeddings are one way to represent a model's properties, with one fixed-length vector per model. Embeddings have the advantage that model profiles can be compared with one another in a shared geometric space. A model *encoder* produces the vector from the model's responses to a set of probe questions. As established by prior work (Zhuang et al., 2025), we call each use of the vector a *downstream task*, a property of the model predicted from its embedding by a small prediction head. Existing encoders build this vector from how each model performs on a shared probe. For example,

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

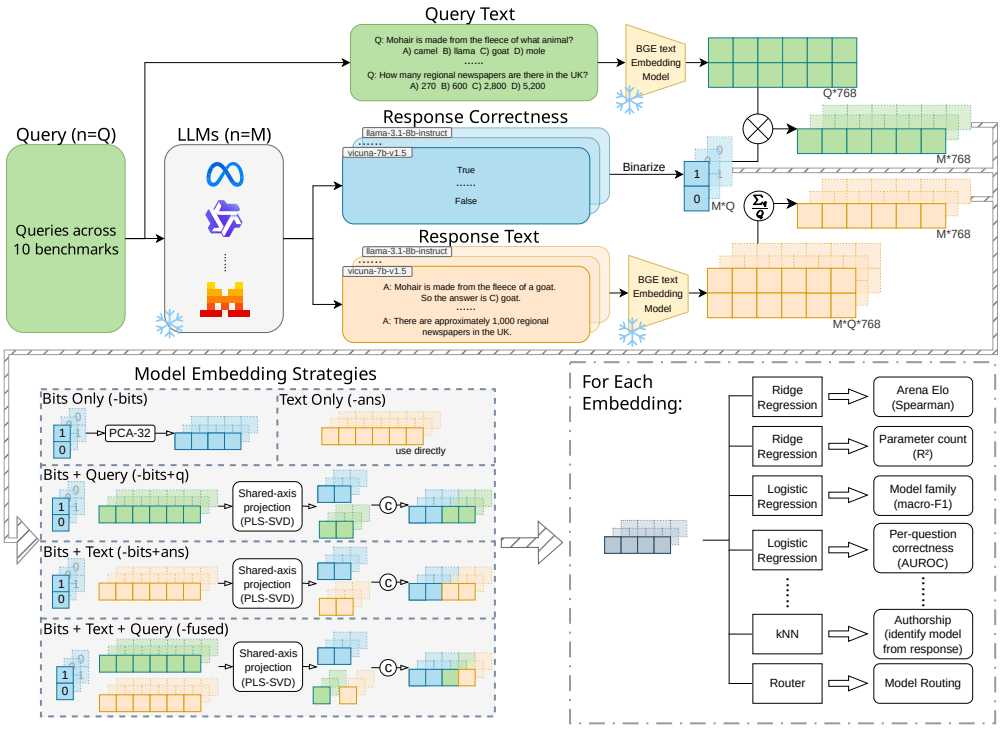


Figure 1: Overview of BULLPEN. Top: each LLM answers the probe questions, giving three channels (question text, correctness bits, answer text). Bottom left: the five embedding strategies. Bottom right: one prediction head per downstream task.

EmbedLLM (Zhuang et al., 2025) learns the vector by reconstructing whether each model answers each question of the probe correctly, with an embedding of the question text as input, and evaluates it on correctness forecasting, routing and benchmark prediction. LOCUS reads the same inputs with an attention encoder (Patel et al., 2026). Embeddings based on item response theory (IRT) do the same with a model's ability and each question's difficulty (Chen et al., 2025; Yao et al., 2026b; Song et al., 2025). These methods therefore describe a model through two *channels*, the *correctness bits* (one record per probe question of whether the model answered it correctly) and the *question text*.

Current model encoders have two limitations. First, most read only the correctness bits and the question text while leaving out what the model writes, which we call the *answer text*. Holistic evaluations argue that models should be compared on behaviors beyond accuracy (Liang et al., 2023), and many of these behaviors are properties of the answers themselves. How long and stylized a model's answers are can sway both human and LLM-judge ratings (Dubois et al., 2024; Li et al., 2024). How often a model refuses safe requests varies widely between models (Röttger et al., 2024). Even the model family can be read from its answers alone (Sun et al., 2025). An embedding built from the answer text therefore has direct access to these properties, whereas one built from correctness sees them only through their effect on accuracy. Fingerprinting methods (Pasquini et al., 2025; Wu et al., 2026) read the answer text but not correctness. No existing encoder combines the answer text with correctness and the question text, although the correctness bits are graded from these same answers, so every correctness-based encoder already pays for the answer text and then discards it. Second, these encoders are evaluated almost entirely on competence and preference, through correctness forecasting, routing, benchmark prediction and human ratings (Table 1). Behavior and style are not among these tasks, and provenance is shown mostly through clusters of similar models or lineage trees (Zhuang et al., 2025; Wu et al., 2026), so no one has measured how much of these properties the channels capture.

Here, we introduce BULLPEN (Behavioral Understanding of Large Language models via Profile ENcodings) to address both limitations. BULLPEN treats one pass over the probe as a source of three channels, namely the correctness bits, the question text and the model's own answer text.

Table 1: Model-embedding papers by input channels and task class.

	behavior, style			*provenance*		
input channels	answer statistics length, hedging, divergence	style effect Arena style deltas, reward-model style	refusal, safety refusal rate, probe refusal (LLMmap)	identity, lineage authorship, family, architecture, base vs chat	checkpoint specs size, license, release date, context	adoption downloads, providers, quantizations
correctness + question text	BULLPEN	BULLPEN	BULLPEN	EmbedLLM, LOCUS, IrtNet, BULLPEN	BULLPEN	BULLPEN
answer text	BULLPEN	BULLPEN	BULLPEN	LLMmap, LLM DNA, BULLPEN	BULLPEN	BULLPEN
correctness + answer text	BULLPEN	BULLPEN	BULLPEN	BULLPEN	BULLPEN	BULLPEN
correctness + question text + answer text	BULLPEN	BULLPEN	BULLPEN	BULLPEN	BULLPEN	BULLPEN

Our encoders read any subset of the three into one model embedding (Figure 1). We also build 20 downstream task benchmarks on behavior, style and provenance from public leaderboards, model metadata and properties of the models' own answers (Table 2).

Our contributions are as follows.

(1) **A three-channel extension of EmbedLLM.** EmbedLLM embeds a model from its correctness bits and the question text. BULLPEN adds the answer text from the same probe pass, and its encoders can read any combination of the three channels.

(2) **Twenty downstream task benchmarks on behavior, style, and provenance.** We build downstream tasks from public leaderboards, model metadata and properties of the models' own answers, such as verbosity, refusal, Arena style ratings and authorship.

(3) **Encoders that read answer text, and what that signal adds.** We represent the answer text jointly with the correctness bits and the question text. Our answer-text encoders outscore every prior encoder on the new tasks, most clearly on verbosity, refusal and authorship.

(4) **Profiling from a few hundred questions.** With 400 questions, our encoders that read answer text keep at least 91% of the skill they reach with the whole 24,118-question pool, which would cost about 75 times as much to collect.

2 Related work

Four lines of work are closest to BULLPEN.

Model embeddings and routing. Sending each input to the right model can cut cost and raise quality. Choosing the model most likely to answer a query well, often under a cost limit (Hu et al., 2024), is a case of algorithm selection (Rice, 1976). Some routers send each query to either a strong or a cheaper model, depending on how hard the query looks (Ding et al., 2024; Ong et al., 2025). Others, such as FrugalGPT (Chen et al., 2024), try cheaper models first and move to a stronger one only when the answer looks unreliable. A router needs a profile of each candidate model, and several build it explicitly from past performance (Wang et al., 2026; Xu et al., 2026; Okamoto et al., 2026; Li et al., 2026). EmbedLLM (Zhuang et al., 2025) turned such profiles into reusable model embeddings, learned from correctness with the question text as side information. LOCUS (Patel et al., 2026) encodes the same inputs with an attention encoder that embeds a new model from a few queries. Embeddings based on item response theory (IRT), such as IrtNet, JE-IRT and IRT-Router, fit each model's ability and each question's difficulty to the correctness bits, and predict a question's difficulty from its text (Birnbaum, 1968; Chen et al., 2025; Yao et al., 2026b; Song et al., 2025). Other work extracts more signal from correctness, such as the wrong answer a model picks (Fei et al., 2026) or how much models' mistakes overlap (Goel et al., 2025). These methods evaluate their embeddings mostly on correctness forecasting, routing and benchmark prediction.

Efficient evaluation. Running every model on every benchmark is costly, so several methods evaluate a model from a small, well-chosen set of questions. Anchor Points and tinyBenchmarks pick representative questions whose answers estimate full benchmark scores (Vivek et al., 2024; Maia Polo et al., 2024), and core-set methods select prompt subsets for the same purpose (Yao et al.,

Table 2: The 20 new tasks, by class.

class	what it measures	tasks
behavior, style (9 tasks)		
answer statistics (3 tasks)	statistics of the model's own answers: length, hedging, divergence from other models	answer length; hedging; answer divergence[‡]
style effect (4 tasks)	how much a rating owes to answer style: LMArena's style-control deltas and the reward-model premium given correctness	style delta (LMArena); style delta, creative (LMArena); style delta, math (LMArena); reward style (Skywork-RM)
refusal, safety (2 tasks)	whether a model refuses (our probe, LLMmap's probes)	refusal rate; probe refusal rate (LLMmap data)
provenance (11 tasks)		
identity, lineage (4 tasks)	which model, family or lineage a checkpoint is; who wrote an answer	authorship[†]; architecture (HF)[†]; model family (HF)[†]; base vs chat (HF)[‡]
checkpoint specs (4 tasks)	facts of the checkpoint on Hugging Face: license, size, context, release date	license (HF); release date (HF); context length (HF); parameter count (HF)[‡]
adoption (3 tasks)	community uptake: downloads, serving providers, Ollama quantizations	downloads (HF); serving providers (HF); Ollama quantizations

[†]: notion from model fingerprinting (Pasquini et al., 2025; Wu et al., 2026; Sun et al., 2025), posed here as prediction from an embedding, with our own labels; [‡]: partial precedent. Established tasks: Table A1.

2026a). Model embeddings face the same cost. LOCUS embeds a new model from a few queries (Patel et al., 2026), and LLMmap identifies a model from a handful of crafted prompts (Pasquini et al., 2025). We also ask each model only a few hundred questions, but use its answers to build the whole embedding, answer text included.

Model fingerprinting. Fingerprinting methods also build a representation of each model from its outputs, and use it to recognize which model produced a piece of text. LLMmap (Pasquini et al., 2025) identifies which model sits behind an interface from its answers to a few crafted queries, and LLM DNA (Wu et al., 2026) derives a representation from a model's outputs to relate models. Behavioral fingerprinting (Pei et al., 2025) describes models through diagnostic prompts judged by another model, and Sun et al. (2025) show that a classifier can tell model families apart from their outputs alone. These methods read only the answer text.

Style and behavior evaluation. Accuracy misses much of what users value, so holistic evaluations compare models on more than accuracy (Liang et al., 2023). Chatbot Arena turns pairwise human votes into ratings (Chiang et al., 2024), AlpacaEval controls its win rates for answer length (Dubois et al., 2024), and Arena's style-controlled ratings remove the effect of length and formatting by regression (Li et al., 2024). Reward models score answers directly (Lambert et al., 2025), and XSTest measures how often models refuse safe requests (Röttger et al., 2024).

3 BULLPEN

3.1 PROBLEM

Our goal is to predict many properties of a language model from one pass over a fixed set of questions. We are given models $\mathcal{M}$ and a *probe* $\mathcal{Q}$, a fixed set of questions that every model answers once. An *encoder* E maps each model m to an embedding $z_m \in \mathbb{R}^d$ using any of three channels, namely which questions the model answers correctly, the questions themselves, and the text it writes. We use *model* for the language models being embedded and *encoder* for the methods that embed them. Each downstream task i is a property $y_i(m)$, predicted from z_m by a small head trained on some models and scored on held-out ones. A good encoder scores well across many such tasks,

Table 3: Mean skill over the width-matched floor on the 20 new tasks, by class.

	behavior, style			*provenance*				
	answer statistics	style effect	refusal, safety	identity, lineage	checkpoint specs	adoption	**all tasks**	M_{eff}-weighted
tasks per column	3	4	2	4	4	3	20	15.2 (eff.)
embedding								
reference: one number per model								
leaderboard scalar (d=1)	0.43	**0.57**	-0.02	0.13	0.24	0.16	$0.27_{\pm 0.12}$	$0.25_{\pm 0.12}$
bits	*0.33*$_{\pm 0.22}$	*0.40*$_{\pm 0.11}$	*0.13*$_{\pm 0.09}$	*0.20*$_{\pm 0.11}$	*0.27*$_{\pm 0.18}$	*0.19*$_{\pm 0.04}$	*0.27*$_{\pm 0.07}$	*0.25*$_{\pm 0.07}$
PCA32-bits-budget	0.28	0.28	0.36	0.22	0.27	0.21	$0.26_{\pm 0.08}$	$0.26_{\pm 0.09}$
PCA32-bits	0.36	0.34	0.02	0.27	**0.31**	0.22	$0.27_{\pm 0.07}$	$0.26_{\pm 0.07}$
IRT-2PL	0.34	**0.60**	0.03	0.12	0.24	0.16	$0.27_{\pm 0.11}$	$0.24_{\pm 0.11}$
bits + question text	*0.38*$_{\pm 0.22}$	*0.45*$_{\pm 0.14}$	*0.17*$_{\pm 0.13}$	*0.22*$_{\pm 0.12}$	*0.32*$_{\pm 0.18}$	*0.22*$_{\pm 0.08}$	*0.31*$_{\pm 0.08}$	*0.29*$_{\pm 0.09}$
EmbedLLM (retrained)	0.40	**0.52**	0.18	0.23	0.33	**0.25**	$0.33_{\pm 0.08}$	$0.32_{\pm 0.09}$
LOCUS (retrained)	0.40	**0.56**	0.01	0.17	0.27	0.20	$0.29_{\pm 0.10}$	$0.27_{\pm 0.10}$
RouteLLM-MF (retrained)	0.41	**0.52**	0.06	0.19	0.28	0.23	$0.30_{\pm 0.09}$	$0.28_{\pm 0.10}$
IRT-Router (retrained)	0.36	0.39	-0.04	0.20	**0.34**	0.20	$0.27_{\pm 0.09}$	$0.26_{\pm 0.09}$
JE-IRT (retrained)	0.41	**0.60**	0.08	0.23	**0.37**	0.24	$0.35_{\pm 0.09}$	$0.32_{\pm 0.09}$
IrtNet (retrained)	0.38	0.41	0.23	0.22	0.33	0.17	$0.30_{\pm 0.08}$	$0.29_{\pm 0.09}$
BULLPEN-bits+q-budget	0.35	0.34	0.41	0.25	**0.33**	0.20	$0.31_{\pm 0.08}$	$0.30_{\pm 0.09}$
BULLPEN-bits+q-full	0.33	0.30	**0.42**	0.28	**0.33**	0.24	$0.31_{\pm 0.08}$	$0.31_{\pm 0.09}$
answer text	*0.68*$_{\pm 0.06}$	*0.49*$_{\pm 0.08}$	*0.29*$_{\pm 0.08}$	*0.28*$_{\pm 0.12}$	*0.31*$_{\pm 0.18}$	*0.29*$_{\pm 0.08}$	*0.39*$_{\pm 0.08}$	*0.38*$_{\pm 0.10}$
LLMmap (retrained)	0.39	0.15	0.15	0.19	0.27	0.24	$0.24_{\pm 0.07}$	$0.25_{\pm 0.07}$
LLM DNA (recomputed)	0.67	**0.55**	0.03	0.22	0.29	0.27	$0.35_{\pm 0.10}$	$0.34_{\pm 0.11}$
BULLPEN-ans-budget	**0.82**	**0.62**	**0.41**	**0.35**	**0.35**	**0.32**	$\mathbf{0.47}_{\pm 0.10}$	$\mathbf{0.46}_{\pm 0.11}$
BULLPEN-ans-full	**0.82**	**0.64**	**0.46**	**0.35**	**0.32**	**0.32**	$\mathbf{0.48}_{\pm 0.10}$	$\mathbf{0.47}_{\pm 0.11}$
bits + answer text	*0.70*$_{\pm 0.02}$	*0.45*$_{\pm 0.14}$	*0.34*$_{\pm 0.14}$	*0.32*$_{\pm 0.11}$	*0.34*$_{\pm 0.20}$	*0.30*$_{\pm 0.09}$	*0.41*$_{\pm 0.09}$	*0.40*$_{\pm 0.09}$
BULLPEN-bits+ans-budget	0.70	0.45	0.40	0.30	**0.33**	**0.28**	$0.40_{\pm 0.09}$	$0.40_{\pm 0.10}$
BULLPEN-bits+ans-full	0.70	0.45	0.29	**0.34**	**0.35**	**0.32**	$0.41_{\pm 0.08}$	$0.41_{\pm 0.09}$
bits + question text + answer text	*0.70*$_{\pm 0.00}$	*0.48*$_{\pm 0.13}$	*0.39*$_{\pm 0.21}$	*0.31*$_{\pm 0.11}$	*0.35*$_{\pm 0.19}$	*0.29*$_{\pm 0.09}$	*0.42*$_{\pm 0.09}$	*0.41*$_{\pm 0.10}$
BULLPEN-fused-budget	0.69	0.48	0.33	0.30	**0.33**	**0.28**	$0.40_{\pm 0.09}$	$0.40_{\pm 0.10}$
BULLPEN-fused-full	0.71	0.49	**0.45**	**0.33**	**0.36**	**0.30**	$\mathbf{0.43}_{\pm 0.09}$	$\mathbf{0.43}_{\pm 0.10}$

Rows grouped by input channels (q: question text, ans: answer text); -budget: the K=400 questions. Bold and green: tied with the column's best (95% interval); underlined: sole best; red: column minimum; gray rows: group means. M_{eff}-weighted: tasks weighted by their share of the effective number of independent tasks (Li & Ji, 2005). LLMmap: 247 of 260 models answered its 8 probes, the rest take the mean signature; its probe-refusal cell is omitted. Scoring rules: Appendix E.

measured as skill over random embeddings of the same width (Eq. 3), while it reads only K of the probe questions per model.

3.2 Data

We use a 31,286-question subset of the EmbedLLM probe (Zhuang et al., 2025), drawn from ten benchmarks: MMLU (Hendrycks et al., 2021a), GSM8K (Cobbe et al., 2021), GPQA (Rein et al., 2024), MathQA (Amini et al., 2019) and others. We collected answers to this probe from 260 open-weight models (Appendix H). Each model answers each question once, with greedy decoding for most models and sampling for reasoning models, and a rule-based grader marks each answer correct or incorrect (Appendix B).

We split the questions once, keeping each benchmark's share the same in every part. The encoders read the *pool* of 24,118 questions, a smaller set is kept for tuning, and a held-out evaluation set supplies the labels that we compute from the models' answers (Appendix B gives the sizes). No encoder reads an evaluation question.

3.3 Channels

Each model has three *channels*, all from one pass over the probe. We write $\mathcal{P}$ for the pool questions that model m answered. Its *correctness bit* b_{mq} is 1 if it answered question q correctly and 0 otherwise. Its *answer text* is the average of its answers after an off-the-shelf text-embedding model, bge-base-en-v1.5, turns each answer into 768 numbers (Xiao et al., 2024). The *question text* is the same for every model, so it can describe a model only through the questions the model gets right. We embed each question q as u_q with the same model and weight it by $b_{mq} - \bar{b}_q$, where $\bar{b}_q$ is the

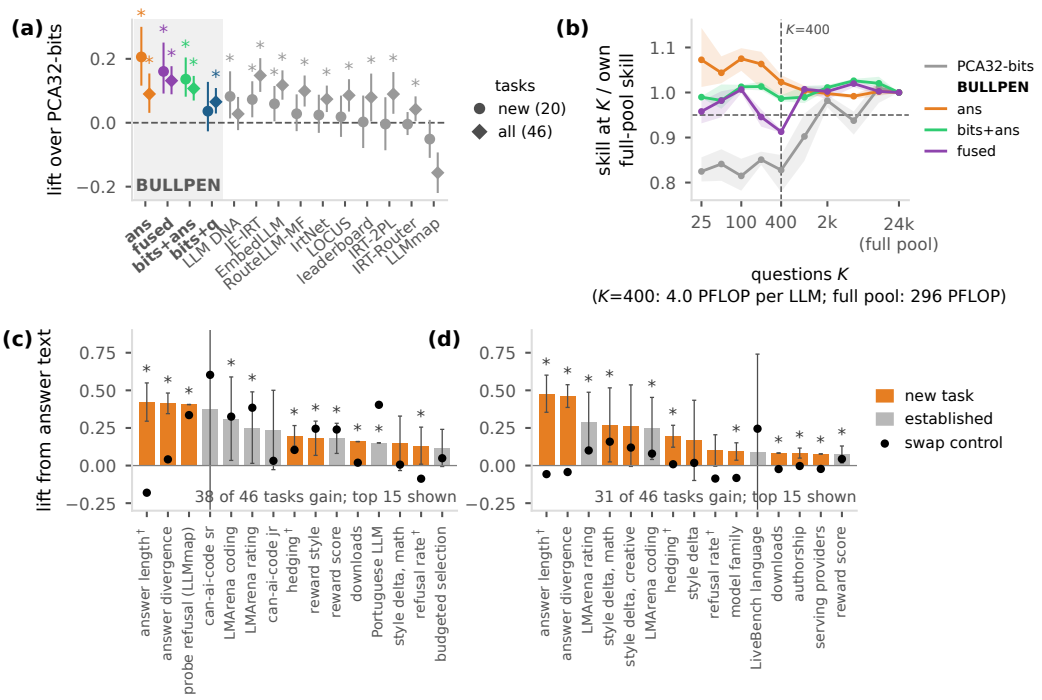


Figure 2: Answer-text lift and question budget. (a) Lift over PCA32-bits on the new and all tasks; $*$: 95% interval above zero; gray: prior encoders. (b) Skill at K questions over each encoder's own full-pool skill. (c, d) Per-task lift from adding answer text to the bits (BULLPEN-bits+ans − PCA32-bits) and to bits+q (BULLPEN-fused − BULLPEN-bits+q); dots: swap control.

share of models that answer q correctly,

$$q_m = \frac{1}{|\mathcal{P}|} \sum_{q \in \mathcal{P}} \left(b_{mq} - \bar{b}_q\right) u_q. \tag{1}$$

The vector q_m therefore points towards the questions on which the model beats the average model.

3.4 Question selection

To keep the cost per model low, BULLPEN embeds each model from K fixed questions. We fit a standard two-parameter IRT model to the training models' correctness bits and keep the K pool questions with the highest Fisher information, which are the questions that best separate strong models from weak ones (Appendix B). An encoder marked *-full* reads all pool questions that every model answered, and one marked *-budget* reads only the K selected ones.

3.5 Encoders

An encoder maps a model's outputs to its model embedding z_m. The main text reports one encoder per set of channels (Appendix B lists the other combinations). Each encoder outputs a 32-dimensional embedding, except BULLPEN-ans, which keeps all 768 dimensions of the sentence encoder. *PCA32-bits*, the correctness-only baseline, keeps 32 principal components of the matrix of correctness bits. *BULLPEN-ans* is the mean of the model's answer embeddings, scaled to length one. As a control, we compare each encoder that reads answer text against its *swap control*, the same encoder given another model's answers instead, which measures how much of its skill comes from the model's own answers.

The joint encoders BULLPEN-bits+q, BULLPEN-bits+ans and BULLPEN-fused compute the covariance between the channels they read, and keep for each channel the directions that vary most

with the others, using multi-block partial least squares SVD (PLS-SVD; cf. Tucker 1958; Westerhuis et al. 1998). With one row per model in a table X_k for each channel k,

$$W_k = \mathrm{svd}_{\mathrm{left}}\big(X_k^\top\,[\,\tilde{X}_j\,]_{j\neq k}\big), \qquad z_m = \big[\,z(x_{m,1}W_1)\;\cdots\;z(x_{m,J}W_J)\,\big]. \tag{2}$$

Here each other channel $\tilde{X}_j$ is scaled to unit Frobenius norm, so the correctness bits do not dominate the signal from the other channels (for the -full encoders the bits have far more columns than the others), and $z(\cdot)$ standardizes each channel's scores.

3.6 Heads and score

For each downstream task (Section 4), a prediction head predicts the model's property from its embedding. Most heads are linear (i.e., a ridge regression for numeric labels and a logistic regression for categorical ones). Each prediction head is trained on a subset of the models and scored on the remaining held-out models. Each task i gets a normalized score s_i, the task's metric (such as a rank correlation) rescaled so that 0 is chance and 1 is a perfect prediction. With many embedding dimensions and few models, a head can also fit patterns by luck, so we subtract the average score of random Gaussian embeddings of the same width, the *floor*. This also makes encoders of different sizes comparable. We report *skill over floor*,

$$\Delta_i(E) = s_i(\text{embeddings of } E) - s_i(\text{random embeddings of the same width}), \tag{3}$$

so 0 means no gain over random numbers. Our summaries average Δ_i over tasks. Because some tasks rank models alike, Table 3 also gives a weighted mean in which each group of such tasks counts as about one task. The weights come from the effective number of independent tasks, which we estimate from the eigenvalues of the task-by-task correlation matrix (Li & Ji, 2005).

4 Downstream tasks

The tasks test both halves of our claim, that answer text adds properties correctness misses and that it keeps what correctness carries. The *new* tasks cover behavior, style and provenance, which practitioners choose models by but prior embeddings were not evaluated on (Table 2). The *established* tasks are of the kinds prior embeddings are evaluated on, and check that the correctness signal survives. No label is computed from anything an encoder reads.

Label sources. Each task's labels describe the models from one source. Leaderboard scores and preference ratings come from public boards, provenance labels come from each model's Hugging Face page and configuration file, and behavior labels are statistics of the models' own answers to the evaluation questions. We match each source to our models by name and drop boards and page fields that are not well represented in our set of models (Table A2 in Appendix A, which also gives each task's source and metric).

Behavior and style. These tasks ask how a model answers. Some labels are measured on the model's own answers to the evaluation questions, such as answer length. Hedging counts uncertainty words such as "I think", and refusal rate counts explicit declines such as "I cannot". Answer divergence measures how far a model's answers are from other models' answers to the same questions. Style effects measure how much of a rating comes from the look of the answers, such as the gap between LMArena's ratings (LMArena, 2025) with and without a control for length and formatting (Li et al., 2024). Another measures how much a reward model prefers a model's answers beyond their correctness.

Provenance. These tasks ask what a checkpoint is and how others use it. Most labels come from each model's Hugging Face page and configuration file, such as its architecture, family, license and parameter count. Others count downloads, serving providers and compressed versions of the model. Following fingerprinting work, the model-family task asks a head to name a held-out model's family (Pasquini et al., 2025; Sun et al., 2025). The authorship task asks whether a model can be recognized from its writing, by checking that the answer statistics its embedding predicts lie closest to its own.

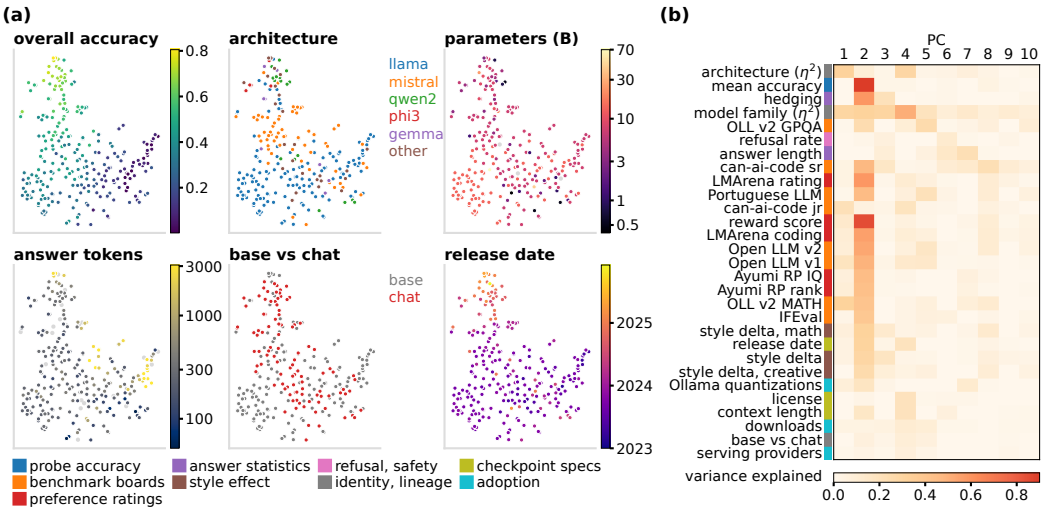


Figure 3: Structure of BULLPEN-fused-full over the 260 models. (a) A t-SNE map colored by six model properties; gray points lack the property. (b) The share of each task's variance each principal component explains (ρ^2, Spearman; η^2 for architecture and model family) over the 29 tasks labeled on at least 30 models, the first ten matched one to one to the components (Hungarian matching), the rest under the component that explains most of their variance (strip: task class); a component's sign is arbitrary.

Competence and preference. The established tasks (Table A1 in Appendix A) predict accuracy on held-out probe questions, published benchmark scores, and human or judge ratings such as LMArena's. They also include model selection, which routes each question to the model predicted to answer it best (Zhuang et al., 2025; Chen et al., 2025; Patel et al., 2026).

5 Results

Protocol. Every number in this section is measured on held-out models. We split the models into five folds, keeping every model released by the same organization (for example, all models from one company's Hugging Face account) in the same fold, and refit the encoder and every prediction head without the predicted fold. We compare against eight prior encoders that cover each set of channels prior work reads. EmbedLLM, LOCUS, RouteLLM-MF and three IRT encoders read correctness and the question text, and LLMmap and LLM DNA read answer text. A single leaderboard score and PCA32-bits show how far a trivial profile gets. We rebuild each on our data (Appendix B).

5.1 Behavior, style and provenance tasks

Answer text carries most of the signal for behavior, style and provenance. Encoders that read it lead on the 20 new tasks (Table 3). BULLPEN-ans-full has a score of 0.48 ± 0.10 while the best prior encoder only achieves 0.35 ± 0.10, and the 6 highest-scoring encoders are all BULLPEN encoders that read answer text. Over the new tasks, BULLPEN-ans-full improves on PCA32-bits by +0.21 ([+0.12, +0.30]), the largest lift of any encoder (Figure 2a). The largest gains are on answer length, answer divergence, refusal rate and authorship (Table A5 in Appendix E).

The swap control shows that these gains come from each model's own answers, and not from having any coherent text embedding as input. When BULLPEN-ans-full is given another model's answers instead of its own, its skill falls by 0.46 ([0.40, 0.52]), from 0.46 to about zero, the level of a random embedding (Table A7 in Appendix F). BULLPEN-fused's performance also falls, but notably much less, by 0.07 ([0.03, 0.11]), due to the signal from its other input channels.

5.2 COMPETENCE AND PREFERENCE TASKS

Reading answer text keeps most of the correctness signal that prior work measures. The 26 established tasks check this (Table A5 in Appendix E). Encoders that read correctness lead here, with JE-IRT highest at 0.65 ± 0.08. BULLPEN-fused-full reaches 0.55 ± 0.07, level with IrtNet (0.55 ± 0.07) and above PCA32-bits (0.44 ± 0.08). On human preference, the leaderboard score alone matches the best encoder (0.71 against 0.71 for LOCUS).

5.3 CHANNEL COMBINATIONS AND EMBEDDING STRUCTURE

Each channel wins a different kind of task (Tables 3 and A5). The correctness bits and the question text carry competence, which the established tasks reward, and the answer text carries behavior, style and provenance, which the new tasks reward. BULLPEN-fused-full, which reads all three channels, keeps both. It is second on the new tasks with 0.43 ± 0.09, behind only the BULLPEN-ans encoders. Over all 46 tasks it ties the best-performing encoder, JE-IRT, with 0.50 against 0.52 (margin +0.02, [−0.04, +0.06], an interval that includes zero).

We examine the structure of the BULLPEN-fused space with t-SNE and principal component analysis (PCA). In the t-SNE projection, which preserves local neighborhood relationships, accuracy follows a continuous gradient along a single direction. Architecture and training stage (base versus chat), by contrast, form distinct regions that run across this gradient (Figure 3a). The PCA panel gives the fraction of variance in each property explained by each of the first ten principal components (PCs) (Figure 3b). Competence-related measures load predominantly on PC2, which explains most of the variance in mean accuracy and reward score, and benchmark and preference measures such as LMArena rating and Open LLM Leaderboard scores follow the same pattern. Lineage variables load instead on PC1 and PC4, which together explain the largest share of the variance in architecture and model family (Appendix D). Competence is thus one axis, and lineage and architecture occupy others.

5.4 SAMPLE EFFICIENCY

At 400 of the 24,118 pool questions, which costs 1.3% of the inference compute of the whole pool, BULLPEN-fused keeps 91% (91–92% across seeds) of the skill it reaches with the whole pool, while the correctness-only PCA32-bits keeps 83% (78–86%) (Figure 2b). Each answer carries more information than one correct-or-incorrect bit, so the encoders that read answer text need fewer questions to profile a model than encoders that use only correctness bits.

6 LIMITATIONS

First, we have a limited number of models, and public leaderboards label only some of them. Small differences, such as BULLPEN-fused-full against JE-IRT over all tasks, therefore cannot yet be separated. More models with full profiles would sharpen these comparisons and let us try more complex encoders and heads. Second, every answer-text result rests on one text-embedding model (BGE) and one set of generation settings, and stronger text encoders may capture more of the answer-text signal. Lastly, every model we embed is open-weight and answers single-turn questions, but this work does not profile closed API models. Because BULLPEN needs only a model's answers, our future work includes adding API models to our model set, and extending the questions to multi-turn and agentic behavior. Appendix B lists further limitations.

7 CONCLUSION

BULLPEN extends model embeddings with each model's own answer text, read alongside its correctness bits and the question text. On new tasks about behavior, style and provenance, embeddings that read answer text predict properties that correctness-based embeddings miss. One embedding that reads all three channels is second only to the answer-text encoders on these tasks and ties the best-performing encoder over all tasks. Because a few hundred questions recover most of this signal, BULLPEN makes it practical to profile a large and growing pool of models by how they answer as well as by what they get right.

AI USE STATEMENT

We used LLM coding agents to assist in writing and testing the code for data collection, processing and all experiments, in giving feedback on the experimental design and checking our interpretation of the results, in finding and checking related literature, and in drafting and editing the manuscript. Some prediction targets are scores assigned by reward models or LLM judges (Appendix A). The authors reviewed all AI-assisted work, verified the code with tests, and checked every claim, number and citation in the text. We take full responsibility for the content of this paper.

ETHICS STATEMENT

The data consist of answers of public open-weight models to public benchmarks, together with public leaderboard records. No human subjects were involved.

REPRODUCIBILITY STATEMENT

Code and configuration are available at `https://github.com/wcquesera/bullpen`, and the repository links to the data. Section 3 and Appendix B define the question split, the encoders and the score. Appendix A gives each task's source, join, coverage and metric, and Appendix B also gives the decoding settings and quality gates for the collected answers and the retraining settings of every competitor.

REFERENCES

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245/.

Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. The moral machine experiment. *Nature*, 563(7729):59–64, 2018. doi: 10.1038/s41586-018-0637-6.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open LLM leaderboard. https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard, 2023.

Allan Birnbaum. Some latent trait models and their use in inferring an examinee's ability. In Frederic M. Lord and Melvin R. Novick (eds.), *Statistical Theories of Mental Test Scores*, pp. 397–479. Addison-Wesley, Reading, MA, 1968.

Rui Cai, Weijie Jacky Mo, Xiaofei Wen, Qiyao Ma, Wenhui Zhu, Xiwen Chen, Muhao Chen, and Zhe Zhao. ModelLens: Finding the best for your task from myriads of models. *arXiv preprint arXiv:2605.07075*, 2026. URL https://arxiv.org/abs/2605.07075.

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. URL https://arxiv.org/abs/2305.05176.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Proceedings*

of the 41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024. URL https://proceedings.mlr.press/v235/chiang24b.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*, 2024.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. Length-controlled AlpacaEval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

Xiao Fei, Yang Zhang, Sarah Almeida Carneiro, and Michalis Vazirgiannis. Every wrong answer counts: Option-level psychometrics for LLM multiple-choice benchmarks. *arXiv preprint arXiv:2608.02966*, 2026.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open LLM leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

Shashwat Goel, Joschka Strüber, Ilze Amanda Auzina, Karuna K Chandra, Ponnurangam Kumaraguru, Douwe Kiela, Ameya Prabhu, Matthias Bethge, and Jonas Geiping. Great models think alike and this undermines AI oversight. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19621–19678. PMLR, 2025. URL https://proceedings.mlr.press/v267/goel25b.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. doi: 10.2307/2333955. URL https://doi.org/10.2307/2333955.

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. In *Agentic Markets Workshop at ICML 2024*, 2024. URL https://openreview.net/forum?id=IVXmV8Uxwh.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1755–1797, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.96. URL https://aclanthology.org/2025.findings-naacl.96/.

J. Li and L. Ji. Adjusting multiple testing in multilocus analyses using the eigenvalues of a correlation matrix. *Heredity*, 95(3):221–227, 2005. doi: 10.1038/sj.hdy.6800717.

Rongjun Li, Ziyu Zhou, and Yihang Wu. FlyRoute: Self-evolving agent profiling via data flywheel for adaptive task routing. *arXiv preprint arXiv:2605.22057*, 2026. URL https://arxiv.org/abs/2605.22057.

Tianle Li, Anastasios Angelopoulos, and Wei-Lin Chiang. Does style matter? disentangling style and substance in Chatbot Arena. LMSYS Org blog, https://lmsys.org/blog/2024-08-28-style-control/, 2024.

Percy Liang, Rishi Bommasani, Tony Lee, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. arXiv:2211.09110.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and Yang Liu. Skywork-Reward-V2: Scaling preference data curation via human-AI synergy. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ofgxkMLqic.

LMArena. LMArena leaderboard. https://huggingface.co/spaces/lmarena-ai/arena-leaderboard, 2025. Hugging Face Space.

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. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Mika Okamoto, Ansel Kaplan Erol, and Glenn Matlin. Trust by design: Skill profiles for transparent, cost-aware LLM routing. *arXiv preprint arXiv:2602.02386*, 2026. URL https://arxiv.org/abs/2602.02386.

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 *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Lorenzo Pacchiardi, Lucy G. Cheke, and José Hernández-Orallo. 100 instances is all you need: Predicting the success of a new LLM on unseen data by testing on a few instances. In *KDD Workshop on Evaluation and Trustworthiness of Generative AI Models*, 2024. URL https://arxiv.org/abs/2409.03563.

Dario Pasquini, Evgenios M. Kornaropoulos, and Giuseppe Ateniese. LLMmap: Fingerprinting for large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 299–318, Seattle, WA, 2025. USENIX Association. URL https://www.usenix.org/conference/usenixsecurity25/presentation/pasquini.

Shivam Patel, William Cocke, and Gauri Joshi. LOCUS: Low-dimensional model embeddings for efficient model exploration, comparison, and selection. *arXiv preprint arXiv:2601.21082*, 2026. URL https://arxiv.org/abs/2601.21082.

Zehua Pei, Hui-Ling Zhen, Ying Zhang, Zhiyuan Yang, Xing Li, Xianzhi Yu, Mingxuan Yuan, and Bei Yu. Behavioral fingerprinting of large language models. In *Open Conference of AI Agents for Science 2025*, 2025. URL https://openreview.net/forum?id=s4gTj3fOIo.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2311.12022.

John R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976. doi: 10.1016/S0065-2458(08)60520-3. URL https://doi.org/10.1016/S0065-2458(08)60520-3.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/. arXiv:2308.01263.

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 (Volume 1: Long Papers)*, pp. 15629–15644, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.761. URL https://aclanthology.org/2025.acl-long.761/.

Mingjie Sun, Yida Yin, Zhiqiu Xu, J. Zico Kolter, and Zhuang Liu. Idiosyncrasies in large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57854–57885. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25z.html.

Ledyard R. Tucker. An inter-battery method of factor analysis. *Psychometrika*, 23(2):111–136, 1958.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, St. Julian's, Malta, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.95. URL https://aclanthology.org/2024.eacl-long.95/.

Chenxu Wang, Hao Li, Yiqun Zhang, Linyao Chen, Jianhao Chen, Ping Jian, Qiaosheng Zhang, and Shuyue Hu. ICL-Router: In-context learned model representations for LLM routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33413–33421, 2026. doi: 10.1609/aaai.v40i39.40628. URL https://ojs.aaai.org/index.php/AAAI/article/view/40628.

Johan A. Westerhuis, Theodora Kourti, and John F. MacGregor. Analysis of multiblock and hierarchical PCA and PLS models. *Journal of Chemometrics*, 12(5):301–321, 1998.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. LiveBench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=sKYHBTAxVa.

Zhaomin Wu, Haodong Zhao, Ziyang Wang, Jizhou Guo, Qian Wang, and Bingsheng He. LLM DNA: Tracing model evolution via functional representations. In *International Conference on Learning Representations (ICLR)*, pp. 7252–7279, 2026. URL https://openreview.net/forum?id=UIxHaAqFqQ.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 641–649. ACM, 2024. doi: 10.1145/3626772.3657878.

Jingjun Xu, Hongji Pu, Tao Feng, Haozhen Zhang, Jiaxuan You, and Ge Liu. RouteProfile: Graph-based profiling for cold-start LLM routing. *arXiv preprint arXiv:2605.00180*, 2026. URL https://arxiv.org/abs/2605.00180.

Jihan Yao, Gantavya Bhatt, Arnav Das, Peter Jin, Ke Bao, Qiaolin Yu, Khushi Bhardwaj, Chang Su, Jialei Wang, Yikai Zhu, Sugam Devare, Damon Mosk-Aoyama, Zhen Dong, Venkat Krishna Srinivasan, Yineng Zhang, Oleksii Kuchaiev, Jiantao Jiao, Banghua Zhu, and Jeff Bilmes. Coresets before score sets: Evaluation-unsupervised prompt subset selection for LLM benchmarks. *arXiv preprint arXiv:2607.09739*, 2026a. URL https://arxiv.org/abs/2607.09739.

Louie Hong Yao, Nicholas Jarvis, Tiffany Zhan, Saptarshi Ghosh, Linfeng Liu, and Tianyu Jiang. A geometric lens on LLM abilities through joint embedding item response theory. *Transactions on Machine Learning Research*, 2026b. ISSN 2835-8856. URL https://openreview.net/forum?id=VQe6p9wn5g.

Yuchen Zeng and Dimitris Papailiopoulos. You don't need to run every eval. *arXiv preprint arXiv:2606.24020*, 2026. URL https://arxiv.org/abs/2606.24020.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *International Conference on Learning Representations (ICLR)*, pp. 76913–76926, 2025. URL https://openreview.net/forum?id=Fs9EabmQrJ. Spotlight.

A TASK CONSTRUCTION

Keeping labels out of the encoder. No task label is ever an encoder input, because an embedding that contained its own label would make the evaluation circular. The encoders read only a model's correctness bits, the question text and the model's own answer text. Labels that we compute from the answers, such as accuracy or answer length, use only the held-out evaluation questions, which no encoder reads. An embedding therefore cannot score well by having seen the questions its labels come from.

Label sources. The *Hugging Face metadata* labels are downloads, license class, architecture, model family, context length, release date and parameter count, read from each model card and configuration file. The *published leaderboards* are the Open LLM Leaderboard v1 and v2 averages, the v2 MATH Lvl 5, GPQA and IFEval columns, LiveBench, can-ai-code (junior and senior pass rates) and a Portuguese community board. The *preference boards* are Chatbot Arena and LMArena ratings, their style-control deltas (the rating minus the style-controlled rating, overall and per category), AlpacaEval 2 length-controlled win rates, and the Ayumi role-play rank and IQ scores. The *safety and values* labels are the UGI willingness score, the refusal rate on the LLMmap probes, and the distance to human preferences in the Moral Machine experiment (Awad et al., 2018). The *serving and packaging* labels count how many quantized versions ollama.com offers for a checkpoint and how many inference providers serve it through the Hugging Face router. The *probe-derived* labels are accuracy on held-out probe questions, answer behavior (length, hedging, refusal and answer divergence), reward-model scores of each (question, answer) pair, authorship, and the routing and selection decisions a deployer would make from the embedding. We score length, hedging and refusal from the answers by pattern matching. Four of these boards label too few of our models to become tasks (Table A2).

Task admission criteria. A board becomes a task only if we can match it to our models by name, it labels enough of them, and we can set its chance level. We match names exactly first, then through an alias map that we checked by hand, and we never use fuzzy matching. Before matching, we ignore case and punctuation and remove the organization prefix, because sources write one checkpoint as `meta-llama/Meta-Llama-3-8B-Instruct`, `Meta-Llama-3-8B-Instruct` or `llama-3-8b-instruct`. A board runs only if it labels at least nine of our models, and it is scored only if every one of the five folds leaves at least eight labeled models to train the head and four to score it. The four boards of Table A2 run but fail the second test. Where a metric has a known chance level, such as one half for a two-class AUROC or 0 for a rank correlation, we use it. Otherwise, we measure the chance level by refitting the same head on embeddings shuffled across models. We do this because, with a few dozen labeled models, a ridge head shrinks its predictions toward the mean of the training folds. That mean is anti-correlated with the held-out labels, so a chance level fixed at 0 would reward every embedding for this shrinkage.

Task types. Each task's type, set by what the head predicts, fixes its metric. *Regression* tasks (34 tasks) predict one number per model. We score most of them by Spearman correlation, because published scores come on different scales, such as an Elo in the thousands and a win rate in percent. The few with a meaningful scale (mean accuracy, parameter count, reasoning tilt and specialization) report R^2. *Classification* tasks (7 tasks) predict a category from the model card or a verdict on each

question. They report macro-F1 for unbalanced columns with many classes and AUROC for two-class and per-question labels. Authorship reports top-1 retrieval accuracy, with chance $1/n$. *Ranking* tasks (2 tasks) order models on a benchmark the head never trained on, scored by nDCG@10, or predict whether an unseen model lands in a board's top K, scored by AUROC. The *pairwise* task predicts which of two models scores higher on each benchmark. The *routing* task sends each question to the model that the embedding predicts is best, and reports the fraction of the gap to an oracle router that it closes. The *selection* task picks a set of models within a 24 GB serving budget, and reports the fraction of the headroom above the best single affordable model that it closes. We score these two decision tasks against a baseline because even an uninformative embedding can route and select. With raw accuracy as the score, always deploying the most accurate model would beat every encoder, and raw retention would put all encoders within a few percent.

Exclusions. Some public signals look like tasks but would not test an embedding fairly, so we leave them out. Targets computed from the pool questions or from the correctness bits only restate what the encoder reads. Boards with too few labeled models in some fold are too small, since a head fitted on so few models reflects its regularization more than the embedding. Router leaderboards rank routing systems, not single models, and model-card fields with too few positive examples cannot form a class. We also keep the older and current versions of a board apart, because they use different prompts.

B Encoders, score and data in detail

Notation. Each model $m \in \mathcal{M}$ answers each question $q \in \mathcal{Q}$ once, with answer a_{mq}, correctness bit $B_{mq} = \mathbf{1}[a_{mq} \text{ correct}]$ and answer embedding $e_{mq} = e(a_{mq})$. Here e is BGE with mean pooling over tokens (768 dimensions), and $u_q = e(q)$ embeds the question. On a subset of questions $S \subseteq \mathcal{Q}$, B_S holds the bit columns of S and T_S holds each model's pooled answer text $t_m(S) = \bar{e}_m/\|\bar{e}_m\|$, with $\bar{e}_m = |S|^{-1}\sum_{q\in S} e_{mq}$. To choose S_K, we fit a two-parameter IRT model $\hat{p}_{mq} = \sigma\big(\hat{\alpha}_q(\hat{\theta}_m - \hat{b}_q)\big)$ on the training models, with model ability $\hat{\theta}_m$, question discrimination $\hat{\alpha}_q$ and question difficulty $\hat{b}_q$. We keep the K pool questions with the largest Fisher information $I_q = \sum_m \hat{\alpha}_q^2 \hat{p}_{mq}(1-\hat{p}_{mq})$. *Full* encoders read $\mathcal{P} = \mathcal{Q}_{\text{pool}}$ (the 900 pool questions every model answered, or all 24,118 for the models that answered every question), and *budget* encoders read S_{400} in the tables and S_K in the K sweep.

Score. With metric μ_i, chance level c_i and best attainable value μ_i^*,

$$s_i(E) = \text{clip}\Big(\frac{\mu_i(\hat{y}_i, y_i) - c_i}{\mu_i^* - c_i}, \pm 1\Big), \qquad \Delta_i(E) = s_i(E) - \frac{1}{R}\sum_{r=1}^{R} s_i\big(E_{\text{null}}^{d,(r)}\big),$$

where each random embedding $E_{\text{null}}^{d,(r)}$ draws an independent $z_m \sim \mathcal{N}(0, I_d)$ for every model. We use $R = 20$ draws under the fast protocol and 10 per fold and seed under the strict one (Protocol details below). A floor can fall below 0, so Δ_i can exceed 1 (Table A8). Over a set of tasks $\mathcal{T}$, the weighted mean is $\bar{\Delta}_{\mathcal{T}} = \sum_i w_i \Delta_i / \sum_i w_i$. Because some tasks rank models alike, such as Arena rating and Elo, w_i is the task's share of the effective number of independent tests (Li & Ji, 2005). With (λ_k, v_k) the eigenpairs of the task-by-task association matrix of the labels, $w_i = \sum_k f(|\lambda_k|)\, v_{ik}^2$ with $f(x) = \mathbf{1}[x \geq 1] + (x - \lfloor x \rfloor)$. The matrix holds the absolute Spearman correlation between numeric labels, the bias-corrected Cramér's V between categorical labels, and 0 for tasks without shared labels. The weights sum to M_{eff} (36.9 over the 46 tasks), an uncorrelated task keeps $w_i = 1$, and k perfectly correlated tasks share one unit.

Joint objectives. BULLPEN-zConcat minimizes the squared reconstruction error over both z-scored channels, so the channel with more columns carries more weight. BULLPEN-zSlice keeps competence once, as c_m. It then adds what the bits hold beyond competence and what the text holds beyond both, each at unit total variance. PLS maximizes $w_b^\top \bar{B}^\top \bar{T} w_t$, and CCA (Hotelling, 1936), with ridge $\lambda = 0.1$, maximizes the correlation between the two channels. The shared part of the CCA embedding, h_m, averages the z-scored canonical variates that exceed the 95th percentile of a 200-permutation null, so BULLPEN-CCA and BULLPEN-CCAfused choose their own width.

Table A1: The 26 established tasks.

class	task	prior source
competence (21 tasks)		
probe accuracy	mean accuracy	benchmark prediction (Zhuang et al., 2025; Chen et al., 2025)
	held-out benchmark	benchmark prediction (Zhuang et al., 2025; Chen et al., 2025)
	benchmark ranking	benchmark prediction (Zhuang et al., 2025; Chen et al., 2025)
	pairwise, per benchmark	benchmark prediction (Zhuang et al., 2025; Chen et al., 2025)
	per-question correctness	correctness forecasting (Zhuang et al., 2025; Chen et al., 2025; Patel et al., 2026; Pacchiardi et al., 2024)
	specialization	specialization clusters (Chen et al., 2025)
	reasoning tilt	new readout of correctness
	error type	new readout of correctness
	item information	IRT item parameters (Chen et al., 2025; Yao et al., 2026b)
benchmark boards	Open LLM v1	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026; Beeching et al., 2023)
	Open LLM v2	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026; Fourrier et al., 2024)
	OLL v2 MATH	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026; Fourrier et al., 2024; Hendrycks et al., 2021b)
	OLL v2 GPQA	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026; Fourrier et al., 2024; Rein et al., 2024)
	IFEval	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026; Fourrier et al., 2024; Zhou et al., 2023)
	Portuguese LLM	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026)
	can-ai-code jr	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026)
	can-ai-code sr	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026)
	LiveBench language	benchmark prediction (Zhuang et al., 2025; Cai et al., 2026; Zeng & Papailiopoulos, 2026; White et al., 2025)
model selection	top-k selection	portfolio selection (Patel et al., 2026)
	query routing	routing (Zhuang et al., 2025; Ong et al., 2025; Patel et al., 2026)
	budgeted selection	portfolio selection (Patel et al., 2026)
preference (5 tasks)		
preference ratings	LMArena rating	Arena preferences (Chiang et al., 2024)
	LMArena coding	Arena preferences (Chiang et al., 2024)
	Ayumi RP rank	new readout of judge rating
	Ayumi RP IQ	new readout of judge rating
	reward score (Skywork-RM)	new readout of reward model (Liu et al., 2026)

Competence (accuracy on held-out probe questions, published benchmark boards, model selection) and preference. Each is a kind of target prior model embeddings are evaluated on; the right column names that kind and where it comes from. Table 2 lists the 20 new tasks.

Models and question split. We collected answers in two stages. The first 104 models answered every probe question. We then added 156 models that public leaderboards list, so that more models carry external labels, and these answered a 1,181-question subset of the probe. We split the probe

Table A2: Tasks the main tables do not score.

task	class	n	reason
AlpacaEval	preference ratings	20	too few labeled models
Arena Elo (LMSYS)	preference ratings	17	too few labeled models
UGI willingness	refusal, safety	15	too few labeled models
Moral Machine (Takemoto)	refusal, safety	23	too few labeled models

Each labels too few of our models for five model-held-out folds. n: models with a label. The fast-protocol budget sweeps (Figures 2b and A2) still score them.

Table A3: Every encoder in the appendix tables. PCA_d: top-d component scores of the column-centered input, $\bar{X}$ that centering, $z(\cdot)$ a column z-score, $\tilde{z}(\cdot)$ a column z-score scaled to unit total block variance, $\mathrm{res}(X|Y)$ the ridge residual of X on $[\mathbf{1}\ Y]$, PCA95 a PCA keeping 95% of the variance, π a derangement of $\mathcal{M}$ (a plain permutation for BULLPEN-AE). A swap control reads what its encoder reads. Main-text encoders in bold.

Name	Reads	d	$E(m)$
Leaderboard scalar	B	1	$c_m = \lvert\mathcal{P}\rvert^{-1} \sum_{q\in\mathcal{P}} B_{mq}$
IRT-2PL	B	1	θ_m of the 2PL, logits $\hat{\alpha}_q(\theta_m - \hat{b}_q)$
PCA32-bits	B	32	$\mathrm{PCA}_{32}(z(B_{\mathcal{P}}))_m$
PCA32-bits (centered) / -budget	B	32	$\mathrm{PCA}_{32}(B_{\mathcal{P}})_m$ ($B_{S_{400}}$)
PCA32-⟨channels⟩ / -budget	B, u, T	32	$\mathrm{PCA}_{32}([\,\tilde{z}(X_1)\ \cdots\ \tilde{z}(X_J)\,])_m$ over the channels in Reads; -budget reads S_{400}
BULLPEN-LogPCA-full	B	32	L2-penalized Bernoulli likelihood, logits $\delta_q + z_m^\top w_q$
BULLPEN-AE-full / -budget	B	32	autoencoder, cross-entropy on $B_{\mathcal{P}}$ ($B_{S_{400}}$)
BULLPEN-QMF-full	B, u	32	EmbedLLM-style factorization, $\sigma(w^\top(p_m \odot Pu_q))$, $z_m = p_m$
BULLPEN-bits+q	B, u	32	$[\,z(\bar{B}_{\mathcal{P}}W_b)\ \ z(Q_{\mathcal{P}}W_q)\,]_m$, PLS-SVD of the bits and the question channel (Eq. 2), 16 + 16
BULLPEN-ans	T	768	$t_m(\mathcal{P})$
BULLPEN-AE-full / -budget	T	32	autoencoder, squared error on $z(T)$
BULLPEN-zConcat-budget	B, T	32	$\mathrm{PCA}_{32}\big([\,z(B_{S_{400}})\ \ z(T_{\mathcal{P}})\,]\big)_m$; the K sweep reads T_{S_K}
BULLPEN-zSlice-budget	B, T	32	$[\,c_m\ \ b_m^\perp\ \ t_m^\perp\,]$; $b^\perp = \mathrm{PCA95}(\mathrm{res}(\mathrm{PCA}_{32}(B_{S_K})\lvert c))$, $t^\perp = \mathrm{PCA95}(\mathrm{res}(T_{S_K}\lvert c, \mathrm{PCA}_{32}(B_{S_K})))$
BULLPEN-bits+ans	B, T	32	$[\,z(\bar{B}_{\mathcal{P}}W_b)\ \ z(\bar{T}_{\mathcal{P}}W_t)\,]_m$; (W_b, W_t) top 16 singular pairs of $\bar{B}_{\mathcal{P}}^\top \bar{T}_{\mathcal{P}}$
BULLPEN-CCA-full	B, T	51–53	$[\,h_m\ \ p_m^b\ \ p_m^t\,]$, CCA of $\mathrm{PCA}_{32}(B_{\mathcal{P}})$ and $\mathrm{PCA}_{32}(T_{\mathcal{P}})$; $p^v = \mathrm{PCA95}(\mathrm{res}(\text{view } v\lvert h))$
BULLPEN-AE-full	B, T	32	autoencoder, cross-entropy on B plus squared error on $z(T)$
BULLPEN-CCAfused-full	B, u, T	28–34	BULLPEN-CCA with bits view $\mathrm{PCA}_{32}(\mathrm{LOCUS}_{128}(B_{\mathcal{P}}, u))$; -budget reads S_{400}
BULLPEN-fused	B, u, T	32	multi-block PLS-SVD of the bits, question and answer channels (Eq. 2), 12 + 10 + 10
Swap control			$E(B, T_{\pi(\cdot)})_m$, $E \in$ {zConcat, PCA32, PLS, Fused, AE}
Null	–	d	$z_m \sim \mathcal{N}(0, I_d)$

once, keeping each benchmark's share the same in every part, into a pool of 24,118 questions, 3,072 tuning questions and 4,096 evaluation questions. The subset splits the same way into 900 pool, 122 tuning and 159 evaluation questions, so all 260 models answered the same 900 pool questions.

Choosing the subset. The subset gives each of the 68 graded parts of the probe a quota proportional to the square root of its size, with at least 12 questions each. Each quota is split evenly over five difficulty levels, cut on the pass rates of the models that answered every question.

Embedding a new model. A new model answers only the K selected questions. We keep the encoder's loadings fixed and place the model's bits by a ridge fit over the questions it answered. We compute and project its question and answer channels in the same way. The held-out models in our experiments are placed by exactly this procedure.

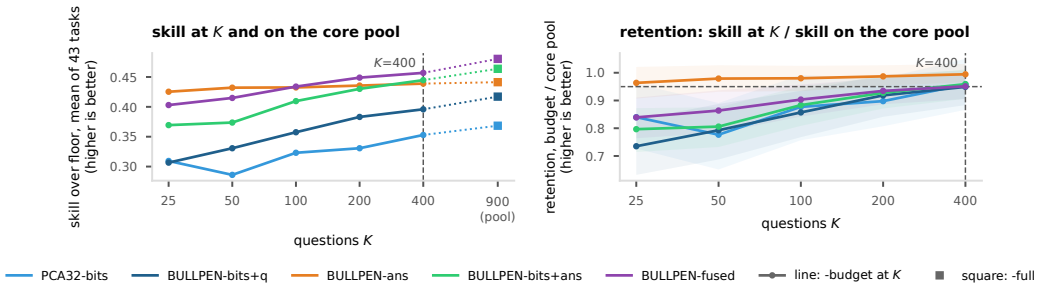


Figure A1: The held-out check of Figure 2b: budget encoders refitted on K questions under the strict protocol (all 260 models, 5 publisher-grouped folds $\times$ 3 seeds), against the 900-question core pool every model answered. Left: skill over the floor; squares: the same encoder on the whole core pool (900). Right: retention, skill at K divided by skill on the core pool (dashed: 95% and K=400, the strict rule's K^*); band: 95% bootstrap over runs and tasks. Higher is better.

Budget against the full pool. This check asks how much skill an encoder keeps when it reads only K questions instead of the whole 24,118-question pool. For each seed, we divide an encoder's skill over floor at K by its skill over floor on the whole pool, and average this ratio over three seeds. We use the 104 models that answered every question, the fast protocol, and the 47 tasks the encoder scores (the main tasks except those that predict held-out benchmarks, plus the four boards of Table A2). Table A4 gives the results. The encoders that read answer text keep nearly all of their skill from about 400 questions, while PCA32-bits needs about 2,000 questions to reach the same level. Values above 100% are within seed noise. We did not run BULLPEN-bits+q on the whole pool.

Table A4: Share of whole-pool skill kept with K questions, in percent (range over seeds at $K = 400$)

Encoder	$K = 400$	$K = 900$	$K = 2{,}000$
BULLPEN-ans	102 (99–104)	100	100
BULLPEN-bits+ans	99 (98–99)	99	101
BULLPEN-fused	91 (91–92)	101	100
PCA32-bits	83 (78–86)	90	98

Competitors. The eight prior encoders are EmbedLLM, LOCUS, RouteLLM-MF, IrtNet, JE-IRT and IRT-Router, which read the correctness bits and the question text (Zhuang et al., 2025; Patel et al., 2026; Ong et al., 2025; Chen et al., 2025; Yao et al., 2026b; Song et al., 2025), LLMmap, which reads the answers to its own eight published probes (Pasquini et al., 2025), and LLM DNA, which needs no training and which we recompute from our answers (Wu et al., 2026). We refit each competitor in every fold on the same training models and the same 900 pool questions as our encoders. Each keeps its published width and follows the released code where there is one. EmbedLLM and IrtNet were designed for the whole EmbedLLM probe, which has far more questions than our pool. We changed the published recipes in five ways. First, we removed inputs that are our prediction targets. IRT-Router's model profile (release date, developer and type) repeats our provenance labels, so our IRT-Router reads only the bits and the question text. RouteLLM-MF keeps its Bradley–Terry loss, but we train it on pairs derived from correctness, in which a correct model beats an incorrect one, following its own golden-label augmentation. We do not use Arena votes, which are our preference targets. The competitor audits below restore both removed inputs. Second, EmbedLLM, RouteLLM-MF, IrtNet, JE-IRT and IRT-Router learn a lookup table with one vector per training model, and only JE-IRT describes how to embed an unseen model. For all five, we place every model, training models included, by fitting its vector to its pool bits with the question parameters frozen, as JE-IRT does. Scoring the trained vectors instead does not help these competitors (competitor audits below). LOCUS, LLMmap and LLM DNA embed a model in one forward pass. Third, IrtNet and JE-IRT initialize the model table with a small variance instead of the published unit variance. With unit variance and 900 questions, the selected checkpoint barely moves from its random start, so these two methods would score like a random embedding.

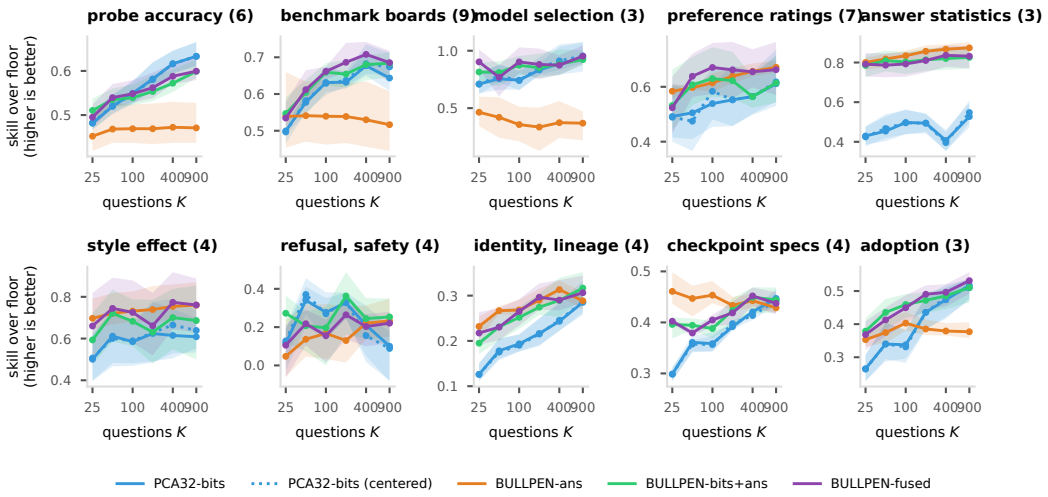


Figure A2: The K sweep per task class (Tables 2 and A1), on the core view: skill over floor of each encoder refitted on K questions (its -budget variant), averaged over the class's tasks with M_{eff} weights (panel title: the class and its number of tasks). Fast protocol (every model embedded, task heads out of fold), so levels differ from the strict Figure A1; at the pool an encoder reads all 900 core questions, and BULLPEN-bits+q is not in this sweep. Lines: mean over 3 seeds; band: lowest to highest seed. Higher is better. The budget of Table 3 is one K (400) for all classes. The sweep scores the tasks of the budget sweep (Appendix B), so class counts differ from Tables 2 and A1.

Fourth, four competitors read a frozen text encoder from our pipeline in place of their own. LLM DNA, LLMmap and RouteLLM-MF use bge-base in place of Qwen3-Embedding-8B, multilingual-e5-large and text-embedding-3-small, in that order. IRT-Router uses all-mpnet-base-v2 in place of bert-base, and EmbedLLM, LOCUS, IrtNet and JE-IRT use all-mpnet-base-v2 as published. Fifth, LOCUS draws fewer anchor questions than published, because our pool is smaller than its anchor set.

Competitor audits. We checked that our changes to the published recipes do not handicap the competitors. Table 3 runs LLMmap as published, on the answers to its own probes, which 247 of the 260 models answered. A version retrained on our pool answers scores a little higher (the own-probe version is lower by -0.10 on the new tasks, $[-0.16, -0.04]$), and both stay below BULLPEN-fused-full on the new tasks. The released LLMmap weights give the same result as a retrained network. Restoring IRT-Router's model profile lowers its score, and training RouteLLM-MF on Chatbot Arena's own votes does not improve it on the 19 of our models that appear in Arena's public battle data (a difference of -0.17, $[-0.31, -0.01]$). Scoring the lookup-table competitors on their own trained vectors, instead of our placement rule, lowers or does not change their scores.

Prediction heads. The prediction heads are small and mostly linear (Section 3.6). A head D_i is a ridge regression for numeric labels or a logistic regression for categorical and per-question labels on z_m. It is fitted out of fold over five model folds, so the prediction $\hat{y}_i(m) = D_i^{(-f(m))}(z_m)$ comes from a head that never saw model m. Two kinds of task add a simple step after the linear head. The routing and selection tasks send each question, or spend a budget, on the models whose predicted correctness is highest. The authorship task predicts each held-out model's profile of answer statistics with a ridge head and checks whether the nearest true profile is the model's own.

Protocol details. We use two protocols. The *strict* protocol is the held-out protocol of Section 5, and Table 3 and the held-out results of Section 5 use it. It averages three seeds, and each seed redraws the folds and the encoder's initialization. Where a task has one label per model (31 of the 46 tasks), we pool the held-out predictions of all five folds and score them once against a floor pooled the same way. Model selection and the per-question tasks cannot be pooled, so for them we average over folds. The tasks that predict held-out benchmarks hold out 13 benchmark groups in turn, with

MMLU split by category. The *fast* protocol fits the encoder on every model and only the heads out of fold. It is cheaper, and we use it for the whole-pool budget sweep (Figure 2b) and for the embedding-structure figure, which shows every model, as the embedding maps of LOCUS do (Patel et al., 2026). All intervals are 95% paired bootstraps. For pooled scores, each draw resamples the models. In Table 3, it resamples folds and seeds in the class columns and tasks in the two right-hand columns.

Further limitations. The folds keep each organization's models together, but they do not keep all models that share a base model together. Model selection and the per-question tasks cannot be pooled, so they have no pooled held-out test. Because name matching ignores the organization, a re-upload can take another model's board row. LLMmap's own probes cover 247 of the 260 models, and Table 3 imputes the rest, so LLMmap's row there is a lower bound.

Collection. Non-reasoning models decode greedily, with up to 3,072 new tokens. Reasoning models loop under greedy decoding, so they sample with temperature 0.6, top-p 0.95 and top-k 20, with up to 12,288 tokens. We drop a model's collection if it covers less than 90% of the subset, if it has no valid output, or if over half its answers on the subset are blank, template tokens or prompt echoes.

C Per-group results for every embedding

Our encoders are named `BULLPEN-<channels>-<full|budget>`, where *full* reads the whole question pool (the 900 pool questions every model answered) and *budget* reads the 400 questions chosen by Fisher information. Table A3 (Appendix B) gives each encoder's width and construction. Figure A3 splits the lift over PCA32-bits of Figure 2 by task class.

D Structure of the BULLPEN-fused embedding

We ran PCA and t-SNE on the BULLPEN-fused-full embedding. Figure 3b relates each of its first ten principal components to every task that has a per-model label. Each of these components holds between 5% and 10% of the variance. PC2 tracks competence, following mean probe accuracy ($\rho = -0.95$), reward-model score (-0.92) and LMArena rating (-0.77). Lineage loads on other components, with architecture on PC1 ($\eta = 0.57$) and model family on PC4 ($\eta = 0.72$). The t-SNE map in Figure 3a shows the same split. Accuracy runs along one direction of the map, while architecture and base versus chat models form regions that cut across it. Newer models are also more accurate ($\rho = 0.56$), so the release-date panel largely repeats the accuracy panel. Both panels use the embedding fitted on all models (seed 0), and the t-SNE runs on its z-scored values with perplexity 30, and the heatmap keeps tasks labeled on at least 30 models.

E Every encoder on every task

How Table 3 is scored and marked. Each cell is the mean skill over the floor of the encoder's width, averaged over the five organization-grouped folds, three seeds and the column's tasks. The leaderboard scalar uses the exact floor for $d = 1$. We fixed the class columns before seeing any score. An encoder is bold when the 95% paired-bootstrap interval of its difference from the column's best includes 0, on the tasks both encoders score. The bootstrap resamples held-out folds and seeds in the class columns and tasks in the two right-hand columns, which also give $\pm$ half the task-bootstrap interval of the mean. An underline marks a sole winner. Green shading marks the bold cells, and red marks the column's lowest embedding other than the leaderboard scalar. Gray italic rows average a channel group. After correction for multiple comparisons, 0 of the 10 class leaders separate from their runner-up. Answer text alone leads every correctness encoder, with or without the question text, on 13 of the 20 new tasks, and on 6 of them the interval excludes zero.

Figure A4 and Table A5 expand Table 3 to one column per task.

Table A5: Skill over the width-matched floor of every encoder on every task.

	benchmark accuracy									public leaderboards								human preference				response style			answer behavior						safety		model metadata										model selection		
	per-question correctness	held-out benchmark	pairwise, per benchmark	benchmark ranking	specialization	error type	item information	reasoning tilt	mean accuracy	IFEval	can-ai-code_jr	can-ai-code_sr	LiveBench language	OLL_v2 GPQA	OLL_v2 MATH	Portuguese LLM	Open LLM v1	Open LLM v2	LMArena rating	LMArena coding	Ayumi RP IQ	Ayumi RP rank	style delta, creative	style delta	style delta, math	reward style	reward score	answer divergence	authorship	hedging	answer length	refusal rate	Moral Machine	probe refusal rate (LLMmap data)	architecture	context length	license	Ollama quantizations	parameter count	base vs chat	model family	downloads	release date	serving providers	top-k selection	budgeted selection	query routing
floor																																														
leaderboard scalar (d=1)	0.01	**0.80**	**0.76**	**0.83**	-0.00	0.56	0.74	0.27	0.97	0.50	-0.06	0.75	0.93	0.43	0.52	0.58	0.75	0.68	0.79	0.72	0.59	0.56	0.45	0.49	0.52	**0.81**	**0.90**	0.56	0.04	0.74	-0.00	0.01	0.06	-0.06	0.06	0.34	0.06	0.19	0.00	0.37	0.05	0.17	0.56	0.13	0.93	0.02	0.53
shuffled text (control)																																														
PCA32-bits+q+ans (shuffled text)	0.45	0.67	0.42	0.77	0.07	0.44	0.65	0.30	0.87	0.09	-0.05	0.42	0.27	0.29	0.41	0.17	0.60	0.29	0.28	0.25	0.34	0.32	0.12	0.10	-0.07	0.50	0.60	0.21	0.01	0.46	0.13	-0.01	0.23	0.32	0.30	0.30	0.22	0.14	0.04	0.35	0.16	0.27	0.52	0.10	0.71	0.75	0.36
PCA32-bits+ans (shuffled text)	0.41	0.68	0.41	0.77	0.07	0.44	0.64	0.26	0.85	0.12	-0.03	0.40	0.15	0.27	0.43	0.13	0.63	0.25	0.26	0.25	0.28	0.32	0.08	0.13	0.12	0.51	0.58	0.16	0.03	0.43	0.13	0.00	0.04	0.59	0.29	0.29	0.23	0.11	0.03	0.32	0.15	0.21	0.52	0.10	0.72	0.80	0.35
PCA32-bits+q+ans (family-shuffled text)	0.45	–	–	–	0.07	0.43	0.59	0.30	0.88	0.21	0.24	0.45	0.15	0.24	0.53	0.19	0.62	0.25	0.31	0.24	0.39	0.42	0.17	0.06	0.09	0.47	0.60	0.14	0.01	0.44	0.21	0.03	0.06	0.60	0.28	0.26	0.16	0.11	0.08	0.34	0.17	0.31	0.50	0.16	0.62	0.59	0.34
PCA32-bits+ans (family-shuffled text)	0.40	–	–	–	0.07	0.43	0.59	0.32	0.85	0.20	0.19	0.38	0.41	0.23	0.42	0.22	0.62	0.25	0.25	0.25	0.06	0.33	0.33	0.08	0.01	0.09	0.51	0.62	0.13	0.01	0.44	0.19	0.04	0.08	0.52	0.28	0.28	0.16	0.10	0.06	0.35	0.18	0.27	0.52	0.18	0.68	0.70	0.34
BULLPEN- AE- full (bits + noise text)	0.30	–	–	–	-0.01	0.46	0.64	0.17	0.73	0.33	-0.16	0.29	0.29	0.25	0.35	0.28	0.65	0.51	0.49	0.38	0.23	0.23	0.12	0.21	0.28	0.66	0.76	0.39	0.03	0.63	0.05	-0.05	-0.08	0.05	0.07	0.28	0.06	0.07	-0.02	0.12	0.05	0.09	0.47	0.06	0.54	0.70	0.41
PCA32-ans (shuffled text)	-0.02	0.03	0.02	0.04	0.00	0.01	0.03	0.05	-0.05	-0.09	-0.12	-0.01	-0.06	-0.08	-0.03	0.17	0.04	-0.08	0.17	0.10	-0.07	-0.06	0.06	0.03	-0.01	0.06	0.03	-0.01	0.01	-0.01	0.00	-0.01	-0.04	-0.20	0.01	-0.01	0.05	0.08	-0.09	0.05	-0.01	0.00	-0.01	-0.02	0.09	0.10	0.02
PCA32- ans (family-shuffled text)	-0.03	–	–	–	0.00	-0.00	-0.11	0.01	0.03	-0.10	-0.02	-0.22	0.14	0.03	-0.01	0.07	-0.05	-0.09	0.07	0.08	0.08	0.10	0.05	0.03	-0.06	0.09	0.06	-0.01	0.00	-0.05	0.05	-0.04	0.13	-0.27	-0.05	-0.03	-0.02	0.01	-0.01	0.09	0.04	0.05	0.01	0.08	-0.13	-0.22	0.02
BULLPEN- bits+ans-full (shuffled text)	0.44	0.74	0.42	0.80	0.06	0.54	0.80	0.39	0.92	0.31	-0.07	0.79	0.81	0.36	0.50	0.52	0.70	0.57	0.65	0.60	0.48	0.47	0.27	0.30	0.40	0.69	0.79	0.35	0.04	0.62	0.08	-0.02	0.01	0.30	0.28	0.45	0.30	0.27	0.11	0.30	0.18	0.31	0.59	0.17	0.80	0.76	0.50
BULLPEN- fused-full (shuffled text)	0.44	0.73	0.45	0.80	0.11	0.50	0.68	0.43	0.92	0.32	-0.02	0.72	0.83	0.39	0.56	0.36	0.64	0.58	0.43	0.45	0.44	0.47	0.21	0.29	0.38	0.59	0.75	0.22	0.03	0.51	0.17	0.01	0.07	0.66	0.31	0.39	0.20	0.28	0.04	0.33	0.21	0.34	0.57	0.12	0.76	0.73	0.48
BULLPEN- bits+ans-full (family-shuffled text)	0.45	–	–	–	0.08	0.54	0.78	0.35	0.91	0.31	0.00	0.48	0.79	0.32	0.54	0.56	0.68	0.49	0.64	0.50	0.48	0.42	0.28	0.29	0.42	0.69	0.80	0.36	0.03	0.62	0.08	-0.02	-0.02	-0.11	0.29	0.43	0.29	0.16	0.12	0.29	0.23	0.32	0.60	0.22	0.72	0.67	0.49
BULLPEN- fused-full (family-shuffled text)	0.45	–	–	–	0.12	0.50	0.68	0.30	0.93	0.27	0.13	0.65	0.60	0.28	0.59	0.41	0.61	0.45	0.39	0.25	0.44	0.47	0.29	0.30	0.37	0.60	0.74	0.28	0.03	0.52	0.17	0.05	-0.02	-0.37	0.31	0.39	0.23	0.21	0.14	0.33	0.23	0.36	0.58	0.18	0.68	0.66	0.47
BULLPEN- zConcat-budget (shuffled text)	0.39	–	–	–	0.04	0.43	0.63	0.23	0.79	0.23	0.05	0.18	0.50	0.24	0.40	0.18	0.51	0.39	0.25	0.29	0.23	0.19	0.07	0.11	-0.01	0.55	0.62	0.11	0.02	0.46	0.07	0.00	-0.03	0.17	0.11	0.21	0.18	0.13	0.04	0.29	0.13	0.20	0.53	0.06	0.62	0.75	0.31
BULLPEN- CCAfused-full (shuffled text)	0.45	–	–	–	0.02	0.40	0.63	0.32	0.85	0.24	-0.04	0.42	0.24	0.15	0.32	0.16	0.48	0.32	0.27	0.33	0.31	0.24	0.10	0.17	0.03	0.47	0.60	0.24	0.01	0.40	0.14	-0.02	-0.07	-0.02	0.03	0.19	0.11	0.09	0.03	0.24	0.15	0.13	0.46	0.07	0.52	0.75	0.31
BULLPEN- ans-full (family-shuffled text)	-0.02	–	–	–	0.01	-0.00	-0.08	0.02	-0.02	-0.00	0.03	-0.38	-0.02	0.07	0.06	-0.02	-0.04	0.03	0.20	0.11	0.12	0.07	0.25	0.18	0.09	0.11	0.11	0.03	0.00	-0.08	-0.00	-0.06	-0.28	-0.15	-0.10	-0.05	0.09	-0.04	0.00	0.09	0.08	0.11	-0.02	0.12	0.07	-0.25	-0.04
BULLPEN-ans- full (shuffled text)	-0.02	0.05	0.05	0.02	-0.00	0.02	0.02	0.04	0.04	-0.11	-0.08	-0.18	-0.28	0.01	0.02	-0.02	0.02	-0.00	-0.12	-0.10	-0.04	-0.08	-0.02	-0.05	-0.06	0.05	0.07	-0.00	0.00	-0.00	-0.05	-0.03	-0.22	0.12	0.01	-0.02	0.07	0.04	0.05	0.09	0.01	0.03	0.00	0.05	0.29	0.06	0.01
bits																																														
IRT-2PL	0.35	**0.80**	**0.76**	**0.83**	-0.00	0.54	0.69	0.23	0.77	**0.82**	-0.04	0.77	0.95	0.45	0.49	0.64	0.75	0.70	0.78	0.71	0.55	0.54	**0.51**	0.50	0.57	**0.81**	**0.90**	0.33	0.03	0.71	-0.03	0.02	0.02	0.04	0.06	0.37	0.03	0.21	0.01	0.35	0.05	0.16	0.57	0.10	0.95	0.49	0.56
PCA32-bits (centered)	**0.51**	0.67	0.49	0.77	**0.13**	0.46	0.68	0.30	0.88	0.25	0.03	0.66	0.66	0.44	0.49	0.03	0.63	0.53	0.19	0.28	0.40	0.42	0.13	0.17	0.31	0.49	0.61	0.32	0.02	0.51	0.30	0.02	0.25	0.30	0.36	0.38	0.14	0.18	0.13	0.42	0.30	0.35	0.54	0.20	0.69	0.67	0.42
PCA32-bits-budget	0.50	0.67	0.47	0.78	0.11	0.46	0.73	0.27	0.87	0.22	0.13	0.76	0.54	0.25	0.46	0.33	0.59	0.35	0.22	0.24	0.45	0.46	0.15	0.21	0.21	0.56	0.62	0.18	0.01	0.50	0.16	0.05	0.09	0.66	0.34	0.37	0.14	0.18	0.05	0.32	0.23	0.27	0.53	0.17	0.65	0.60	0.39
PCA32-bits	**0.51**	0.67	0.48	0.76	0.12	0.45	0.70	0.30	0.88	0.21	-0.11	0.19	0.58	0.37	0.48	0.11	0.58	0.43	0.27	0.27	0.40	0.44	0.23	0.28	0.39	0.45	0.55	0.31	0.02	0.51	0.26	0.07	0.09	-0.03	0.38	0.39	0.18	0.17	0.14	0.36	0.31	0.29	0.53	0.20	0.70	0.71	0.43
PCA32-bits-budget (centered)	0.50	–	–	–	0.11	0.46	0.72	0.32	0.87	0.19	0.14	0.71	0.43	0.22	0.45	0.32	0.59	0.39	0.28	0.32	0.46	0.45	0.13	0.27	0.26	0.56	0.64	0.19	0.01	0.51	0.17	0.08	-0.05	0.69	0.37	0.38	0.16	0.15	0.06	0.32	0.25	0.26	0.55	0.17	0.65	0.72	0.38
BULLPEN- AE- full	0.44	–	–	–	0.04	0.53	0.59	0.34	**0.96**	0.36	-0.01	0.77	0.82	0.37	0.47	0.50	0.72	0.61	0.78	0.69	0.58	0.56	0.39	0.51	0.44	0.76	0.84	0.54	0.03	0.67	0.13	-0.04	0.06	0.10	0.12	0.38	0.12	0.20	0.02	0.26	0.11	0.25	0.57	0.11	0.75	0.76	0.54
BULLPEN- AE-budget	0.43	–	–	–	0.01	0.53	0.54	0.33	0.95	0.41	-0.02	0.70	0.78	0.33	0.56	0.58	0.72	0.62	0.73	0.58	0.56	0.54	0.30	0.48	0.35	0.73	0.81	0.55	0.03	0.67	0.15	-0.05	0.11	0.11	0.13	0.35	0.15	0.19	0.03	0.23	0.08	0.24	0.55	0.11	0.77	0.73	0.49
BULLPEN- LogPCA-full	0.47	–	–	–	0.12	0.47	0.66	0.32	0.76	0.30	0.09	0.32	0.65	0.41	0.48	0.15	0.70	0.48	0.33	0.30	0.53	0.50	0.27	0.27	0.35	0.56	0.65	0.38	0.02	0.49	0.18	0.04	-0.13	0.42	**0.48**	0.42	0.22	0.19	0.17	0.42	0.30	0.38	0.54	0.15	0.80	0.78	0.50
bits + question text																																														
JE-IRT (retrained)	0.48	0.78	0.63	0.82	0.11	0.52	0.66	0.51	0.91	0.48	0.26	0.86	**0.97**	**0.49**	0.58	0.56	**0.78**	**0.71**	0.73	**0.75**	0.56	0.56	0.49	0.59	0.56	0.76	0.87	0.36	0.05	0.67	0.20	0.05	-0.04	0.11	0.20	0.42	0.27	0.19	0.16	0.37	0.28	0.33	0.62	0.21	**1.00**	0.73	0.50
EmbedLLM (retrained)	0.47	0.78	0.58	0.81	0.11	0.51	0.55	**0.54**	0.93	0.45	0.12	0.67	0.73	0.46	**0.69**	0.55	0.71	0.66	0.68	**0.75**	0.48	0.49	0.46	0.43	0.44	0.73	0.84	0.43	0.04	0.64	0.15	0.07	0.32	0.29	0.22	0.40	0.22	0.27	0.10	0.39	0.26	0.32	0.61	0.15	0.83	0.79	0.49
RouteLLM-MF (retrained)	0.43	0.75	0.51	0.80	0.07	0.48	0.64	0.45	0.90	0.42	**0.28**	0.64	0.77	0.43	0.64	0.59	0.69	0.64	0.70	0.67	0.52	0.50	0.39	0.45	0.46	0.78	0.88	0.48	0.02	0.62	0.13	0.04	-0.09	0.08	0.17	0.36	0.12	0.26	0.08	0.34	0.24	0.28	0.56	0.15	0.84	0.78	0.47
LOCUS (retrained)	0.44	0.74	0.39	0.81	0.06	0.49	0.61	0.41	**0.96**	0.42	0.09	0.42	0.81	0.35	0.53	0.47	0.66	0.65	**0.84**	**0.75**	0.55	0.53	0.39	0.55	0.53	0.77	**0.90**	0.36	0.04	0.65	0.20	-0.06	-0.11	0.09	0.14	0.32	0.13	0.18	0.04	0.29	0.21	0.27	0.59	0.15	0.77	**0.90**	0.49
IrtNet (retrained)	0.42	0.76	0.58	0.79	0.08	0.47	0.61	0.48	0.84	0.41	0.12	0.36	0.67	0.43	0.65	0.46	0.74	0.66	0.60	0.62	0.42	0.44	0.22	0.37	0.40	0.67	0.80	0.40	0.04	0.63	0.10	0.04	0.19	0.41	0.21	0.38	0.20	0.10	0.12	0.39	0.24	0.28	0.62	0.12	0.87	0.67	0.43
RouteLLM-MF-32 (retrained, d=32)	0.43	–	–	–	0.04	0.53	0.66	0.40	0.88	0.40	0.24	0.59	0.66	0.40	0.57	0.59	0.69	0.58	0.68	0.65	0.50	0.52	0.22	0.41	0.46	0.73	0.81	0.43	0.03	0.65	0.07	0.01	-0.11	-0.14	0.14	0.38	0.14	0.21	0.10	0.31	0.18	0.27	0.56	0.14	0.72	0.79	0.50
LOCUS-32 (retrained, d=32)	0.43	–	–	–	0.02	0.53	0.61	0.25	0.95	0.37	-0.03	0.54	0.82	0.31	0.52	0.61	0.74	0.60	0.74	0.63	0.50	0.48	0.27	0.35	0.45	0.73	0.82	0.47	0.03	0.67	0.18	-0.07	0.04	-0.14	0.13	0.37	0.08	0.21	0.03	0.25	0.15	0.25	0.57	0.13	0.73	0.83	0.53
IRT-Router (retrained)	0.41	0.69	0.56	0.77	0.06	**0.49**	**0.82**	0.30	0.81	0.29	0.06	0.72	0.48	0.40	0.61	0.42	0.71	0.63	0.45	0.42	0.30	0.30	0.17	0.32	0.42	0.67	0.80	0.34	0.02	0.58	0.17	0.05	-0.33	-0.12	0.26	0.43	0.24	0.16	0.09	0.28	0.25	0.28	0.60	0.15	0.79	0.71	0.47
PCA32-bits+q	0.50	0.67	0.49	0.78	0.11	0.45	0.72	0.33	0.88	0.29	0.12	0.06	0.59	0.40	0.51	0.14	0.58	0.49	0.29	0.36	0.35	0.40	0.14	0.29	0.39	0.55	0.63	0.23	0.02	0.49	0.29	0.07	0.02	-0.10	0.35	0.34	0.17	0.20	0.11	0.36	0.25	0.31	0.52	0.13	0.67	0.58	0.43
PCA32-bits+q-budget	0.49	–	–	–	0.10	0.45	0.70	0.21	0.86	0.35	0.22	0.82	0.46	0.22	0.47	0.28	0.62	0.42	0.22	0.24	0.44	0.41	0.20	0.26	0.27	0.49	0.58	0.19	0.03	0.51	0.20	0.05	0.03	0.71	0.28	0.37	0.14	0.13	0.15	0.26	0.23	0.28	0.56	0.14	0.65	0.69	0.37
BULLPEN- QMF- budget	0.48	–	–	–	0.07	**0.55**	0.61	0.40	0.89	0.47	0.09	0.80	0.80	0.41	0.57	**0.71**	0.77	0.66	0.65	0.67	**0.61**	**0.58**	0.37	0.40	0.46	0.77	0.84	0.36	0.03	0.70	0.05	-0.01	-0.03	0.16	0.11	**0.48**	0.15	0.20	-0.03	0.34	0.16	0.29	0.62	0.13	0.78	0.55	**0.58**
BULLPEN- bits+q-full	0.47	0.73	0.50	0.81	0.12	0.51	0.76	0.38	0.93	0.30	0.04	0.80	0.59	0.40	0.57	0.36	0.71	0.60	0.33	0.37	0.48	0.52	0.09	0.27	0.23	0.60	0.70	0.26	0.03	0.50	0.23	0.09	0.00	0.75	0.41	0.39	0.20	0.20	0.11	0.40	0.29	0.36	0.61	0.15	0.79	0.78	0.48
BULLPEN- bits+q-budget	0.47	0.74	0.49	0.80	0.11	0.51	0.77	0.41	0.92	0.26	0.08	**0.88**	0.59	0.19	0.46	0.43	0.70	0.45	0.37	0.33	0.51	0.46	0.19	0.23	0.36	0.56	0.70	0.26	0.03	0.57	0.21	0.06	0.05	0.76	0.34	0.37	0.20	0.15	0.14	0.38	0.25	0.32	0.61	0.14	0.74	0.66	0.43
BULLPEN-QMF- full	0.46	–	–	–	0.02	0.51	0.54	0.42	0.79	0.40	-0.09	0.83	0.73	0.39	0.48	0.53	0.74	0.58	0.61	0.55	0.59	0.55	0.35	0.38	0.34	0.71	0.79	0.53	0.03	0.62	0.07	0.02	0.13	0.27	0.19	0.27	0.09	0.22	0.03	0.25	0.12	0.19	0.52	0.09	0.72	0.83	0.42
answer text																																														
LLM DNA (recomputed)	0.32	–	–	–	0.06	0.40	0.61	0.20	0.50	0.31	0.03	0.37	0.55	0.29	0.48	0.22	0.42	0.38	0.53	0.47	0.31	0.30	0.45	0.53	0.53	0.69	0.72	0.87	0.05	0.66	0.48	0.07	0.29	-0.01	0.17	0.38	0.18	0.29	0.08	0.38	0.28	0.34	0.52	0.18	0.53	0.75	0.34
PCA32-ans	0.37	0.58	0.21	0.56	0.01	0.38	0.48	0.08	0.49	0.08	0.03	0.23	0.68	0.12	0.24	0.23	0.39	0.15	0.51	0.42	0.33	0.36	0.21	0.34	0.43	0.42	0.41	0.77	0.16	0.67	0.75	0.16	0.17	0.75	0.22	0.37	0.19	0.24	-0.05	0.44	0.44	0.42	0.58	0.29	0.43	0.62	0.15
PCA32-ans-budget	0.37	–	–	–	0.02	0.38	0.51	0.08	0.50	0.05	0.10	0.21	0.50	0.16	0.19	0.14	0.34	0.12	0.41	0.42	0.34	0.38	0.18	0.30	0.35	0.40	0.41	0.76	0.17	0.70	0.73	0.16	-0.05	0.72	0.21	0.34	0.23	0.18	-0.05	0.44	0.38	0.41	0.57	0.30	0.41	0.53	0.15
LLMmap (retrained)	0.17	–	–	–	0.04	0.22	0.34	0.13	0.27	0.36	0.18	-0.18	-0.21	0.09	0.12	0.04	0.24	0.23	0.07	0.18	0.07	0.05	0.06	0.06	0.06	0.42	0.43	0.34	0.03	0.40	0.44	0.15	0.15	–	0.12	0.30	0.14	0.05	0.16	0.39	0.23	0.37	0.49	0.30	0.54	0.38	0.20
BULLPEN- ans- full	0.36	0.65	0.50	0.64	0.09	0.39	0.52	0.29	0.69	0.35	0.12	0.33	0.49	0.21	0.45	0.15	0.54	0.30	0.59	0.60	0.41	0.37	0.50	**0.64**	**0.74**	0.68	0.71	**0.90**	**0.22**	0.77	0.80	0.31	0.15	0.60	0.25	0.38	0.25	0.28	0.01	0.52	0.41	0.41	0.64	0.27	0.70	0.77	0.25
BULLPEN- ans-budget	0.36	0.64	0.50	0.62	0.09	0.39	0.50	0.32	0.67	0.36	0.15	0.30	0.55	0.26	0.45	0.19	0.53	0.33	0.58	0.54	0.37	0.35	0.50	0.61	0.70	0.68	0.69	0.89	0.21	0.78	0.78	0.30	0.03	0.51	0.22	0.40	**0.31**	0.29	0.06	0.53	0.42	0.42	0.62	0.24	0.72	0.66	0.25
BULLPEN- AE- full	0.30	–	–	–	0.01	0.41	0.47	0.16	0.58	0.24	-0.11	0.53	0.69	0.18	0.34	0.14	0.49	0.21	0.54	0.55	0.32	0.32	0.41	0.53	0.59	0.66	0.67	0.72	0.06	0.86	0.31	0.05	**0.33**	0.15	0.11	0.34	0.08	0.36	-0.04	0.27	0.18	0.29	0.49	0.22	0.51	0.60	0.30
BULLPEN- AE-budget	0.30	–	–	–	0.01	0.41	0.46	0.17	0.58	0.31	-0.07	0.49	0.65	0.14	0.34	0.15	0.51	0.20	0.52	0.49	0.23	0.22	0.49	0.60	0.55	0.59	0.63	0.70	0.05	0.85	0.25	-0.02	0.18	0.33	0.14	0.34	0.13	**0.37**	0.02	0.29	0.16	0.27	0.50	0.20	0.55	0.53	0.29
answer statistics																																														
answer statistics (d=6)	0.31	–	–	–	-0.03	0.36	0.14	0.13	0.30	0.29	-0.13	0.28	0.48	0.09	0.35	0.11	0.29	0.24	0.25	0.34	0.43	0.36	0.26	0.32	0.35	0.64	0.59	0.52	0.14	**0.89**	**0.91**	**0.47**	-0.08	0.12	0.11	0.28	0.06	0.27	**0.32**	0.34	0.13	0.36	0.55	0.17	0.32	0.64	0.24
bits + answer text																																														
PCA32-bits+ans	0.47	0.68	0.39	0.76	0.10	0.45	0.64	0.30	0.75	0.20	0.09	0.64	0.70	0.42	0.48	0.37	0.56	0.42	0.53	0.41	0.42	0.40	0.22	0.39	0.49	0.55	0.63	0.67	0.13	0.62	0.69	0.11	0.17	0.62	0.34	0.37	0.27	0.16	0.10	0.49	0.44	**0.45**	0.57	0.25	0.69	0.75	0.37
PCA32-bits+ans-budget	0.47	–	–	–	0.09	0.44	0.64	0.31	0.82	0.17	0.06	0.34	0.68	0.27	0.40	0.33	0.55	0.38	0.60	0.46	0.40	0.41	0.11	0.33	0.32	0.53	0.57	0.64	0.15	0.63	0.67	0.18	0.16	0.44	0.27	0.35	0.23	0.25	0.08	0.47	0.39	0.36	0.61	0.27	0.58	0.69	0.34
BULLPEN- bits+ans-full	0.46	0.71	0.44	0.78	0.12	0.50	0.68	0.42	0.86	0.24	0.13	0.56	0.66	0.35	0.57	0.26	0.66	0.45	0.52	0.59	0.49	0.48	0.26	0.37	0.54	0.63	0.73	0.72	0.14	0.71	0.68	0.20	0.11	0.37	0.33	0.47	0.17	0.24	0.11	0.47	0.41	**0.45**	0.63	0.27	0.72	0.83	0.44
BULLPEN- bits+ans-budget	0.46	0.70	0.44	0.78	0.11	0.49	0.71	0.39	0.85	0.28	0.17	0.52	0.74	0.36	0.52	0.25	0.55	0.47	0.58	0.62	0.41	0.38	0.22	0.41	0.53	0.64	0.72	0.73	0.14	0.71	0.67	0.21	0.22	0.59	0.28	0.44	0.20	0.22	0.05	0.42	0.35	0.35	0.63	0.26	0.69	0.74	0.39
BULLPEN- CCA- full	0.47	–	–	–	0.11	0.45	0.60	0.32	0.73	0.19	0.13	0.54	0.86	0.40	0.49	0.26

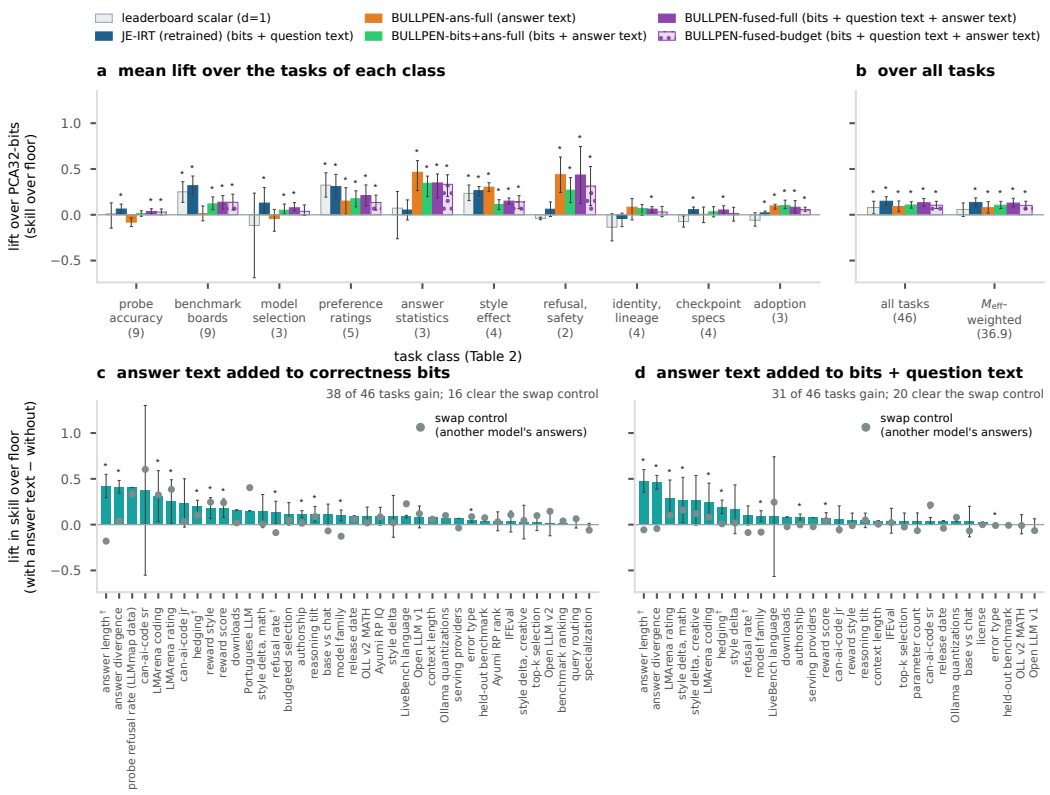


Figure A3: What each embedding adds over the correctness baseline, in the task classes of Table 2 (strict protocol; Table 3 prints the raw skill). (a) Mean lift in skill over floor over PCA32-bits (z-scored), per class; (b) the same over all tasks, unweighted and M_{eff}-weighted (Li & Ji, 2005). Error bars: 95% paired task bootstrap; $\star$: the interval is above zero. JE-IRT is the strongest retrained competitor; the retrained competitors have a fixed input, so only our encoders are tested with and without answer text. BULLPEN-fused-budget reads K^* questions. (c,d) Tasks where adding answer text helps, each a pair of embeddings that differ only in that channel: (c) BULLPEN-bits+ans − PCA32-bits, (d) BULLPEN-fused − BULLPEN-bits+q. Error bars: 95% interval over the 15 fold × seed runs; $\star$: that interval excludes zero. Gray dots: the same difference with each model given another model's answers. †: pattern-matched behavior traits.

F Swap controls

The swap control of Section 3.5 gives each encoder another model's answers, so any skill it keeps cannot come from the model's own writing. The tables call these controls shuffled-text twins, and a *family-shuffled* twin takes answers only from models of another organization. Table A7 compares each swap control with the floor, with the bits-only embedding and with its own encoder (Section 5.1). Two comparisons with the bits also differ slightly in budget or width, since the BULLPEN-zConcat control reads 400 bit questions and the BULLPEN-CCAfused control keeps 24–31 dimensions. The BULLPEN-fused swap control still scores above PCA32-bits by +0.06 ([+0.02, +0.11]), which is the part of BULLPEN-fused's lift that does not depend on each model's own answers. Adding answer text to the bits helps on 38 of 46 tasks, and adding it to the bits and question text helps on 31 (Figure 2c,d).

G Tasks that need both channels

Table A8 tests, for each task, whether the joint embedding BULLPEN-bits+ans-full beats both the better of the bits and answer-text embeddings and its own swap control, on the held-out scores

Table A6: Table 3's embeddings on all 46 tasks, by label source.

	probe correctness	published benchmark score	human judgment	model judgment	answer-text statistic	registry record	**all tasks**	M_{eff}-weighted
tasks per column	11	10	5	4	5	11	46	36.9 (eff.)
embedding								
reference: one number per model								
leaderboard scalar (d=1)	0.50	0.60	**0.60**	**0.71**	0.25	0.18	$0.45_{\pm 0.09}$	$0.43_{\pm 0.10}$
bits								
PCA32-bits-budget	0.53	0.43	0.20	0.52	0.31	0.24	$0.38_{\pm 0.06}$	$0.39_{\pm 0.07}$
PCA32-bits	0.55	0.35	0.29	0.46	0.22	0.27	$0.37_{\pm 0.06}$	$0.37_{\pm 0.07}$
IRT-2PL	0.55	0.62	**0.61**	**0.70**	0.21	0.17	$0.46_{\pm 0.08}$	$0.44_{\pm 0.10}$
bits + question text								
EmbedLLM (retrained)	**0.60**	0.59	**0.55**	0.64	0.31	0.27	$0.48_{\pm 0.07}$	$0.47_{\pm 0.07}$
LOCUS (retrained)	0.57	0.52	**0.61**	**0.69**	0.25	0.21	$0.45_{\pm 0.08}$	$0.44_{\pm 0.08}$
RouteLLM-MF (retrained)	0.57	0.59	**0.54**	**0.67**	0.27	0.23	$0.47_{\pm 0.07}$	$0.46_{\pm 0.08}$
IRT-Router (retrained)	0.55	0.51	0.35	0.52	0.20	0.25	$0.41_{\pm 0.07}$	$0.41_{\pm 0.08}$
JE-IRT (retrained)	**0.60**	**0.67**	**0.62**	**0.69**	0.28	0.28	$\mathbf{0.52}_{\pm 0.07}$	$\mathbf{0.51}_{\pm 0.08}$
IrtNet (retrained)	0.56	0.54	0.44	0.58	0.32	0.25	$0.44_{\pm 0.07}$	$0.43_{\pm 0.07}$
BULLPEN-bits+q-budget	0.57	0.48	0.30	0.56	0.37	0.27	$0.43_{\pm 0.07}$	$0.44_{\pm 0.07}$
BULLPEN-bits+q-full	**0.59**	0.49	0.26	0.58	0.37	0.29	$0.43_{\pm 0.07}$	$0.44_{\pm 0.07}$
answer text								
LLMmap (retrained)	0.22	0.14	0.09	0.24	0.33	0.23	$0.20_{\pm 0.05}$	$0.20_{\pm 0.06}$
LLM DNA (recomputed)	0.40	0.36	**0.50**	0.51	0.41	0.26	$0.38_{\pm 0.06}$	$0.37_{\pm 0.07}$
BULLPEN-ans-budget	0.46	0.38	**0.59**	0.52	0.65	**0.34**	$\mathbf{0.45}_{\pm 0.06}$	$\mathbf{0.45}_{\pm 0.06}$
BULLPEN-ans-full	0.47	0.36	**0.61**	0.54	**0.68**	**0.33**	$\mathbf{0.46}_{\pm 0.06}$	$\mathbf{0.45}_{\pm 0.07}$
bits + answer text								
BULLPEN-bits+ans-budget	0.55	0.46	0.47	0.54	0.58	0.30	$\mathbf{0.47}_{\pm 0.06}$	$\mathbf{0.47}_{\pm 0.06}$
BULLPEN-bits+ans-full	0.57	0.46	0.45	0.58	0.54	**0.34**	$\mathbf{0.47}_{\pm 0.06}$	$\mathbf{0.48}_{\pm 0.06}$
bits + question text + answer text								
BULLPEN-fused-budget	0.57	0.49	0.44	0.58	0.55	**0.30**	$\mathbf{0.47}_{\pm 0.06}$	$\mathbf{0.48}_{\pm 0.06}$
BULLPEN-fused-full	**0.58**	0.49	0.51	0.60	0.61	**0.33**	$\mathbf{0.50}_{\pm 0.06}$	$\mathbf{0.50}_{\pm 0.06}$

New and established tasks, grouped by who produced the label: our probe's correctness, a published benchmark score, human judgment, a model judge (LLM or reward model), a statistic of the answer text, or a registry record. LLMmap and LLM DNA skip the per-benchmark refits, so their rows leave out the three benchmark-transfer tasks (43 tasks). Strict protocol: models held out in 5 publisher-grouped folds × 3 seeds; a cell is the mean skill over the encoder's width-matched Gaussian floor (the leaderboard scalar reads the exact d=1 floor), averaged over folds, seeds and the column's tasks. -full: the 900-question pool; -budget: the K=400 questions chosen by Fisher information, the smallest K at which BULLPEN-fused keeps 95% of its full-pool mean over the 43 tasks other than benchmark transfer under this protocol (95.1% at K=400, 93.5% at 200). Bold: the 95% paired-bootstrap CI of the difference from the column's best covers 0, on the tasks both encoders score; the bootstrap resamples held-out folds and seeds in the class columns (their tasks are fixed) and tasks in the two right-hand columns. Underlined: sole winner. The two right-hand columns carry ± half the task-bootstrap interval of the mean. M_{eff}-weighted: each task weighted by its share of the effective number of independent tasks (Li & Ji, 2005), so redundant tasks split one unit. Green shading marks the bold cells; red marks the column's lowest embedding (leaderboard scalar excluded). Gray italic rows: the mean over the embeddings in that channel group. LLMmap: 247 of 260 models answered its 8 probes, the rest take the mean signature; its probe-refusal cell is omitted (the label comes from those probes).

of Table 3. Both tests pool the held-out predictions where a task has one label per model. After correction, 0 of the 31 tasks pass both tests.



Figure A4: Every encoder (rows, grouped by input channel) on every task (columns, grouped by task group). Cell: skill over the encoder's width-matched floor as a share of the column's best encoder (star); values at or below the floor are 0. White: not scored (the three benchmark-transfer tasks are scored only for Table 3's encoders, and LLMmap and LLM DNA skip the per-benchmark refits). Strict protocol: 5 publisher-grouped folds $\times$ 3 seeds.

Table A7: Paired tests of each encoder against its shuffled-text twins.

A	B	Δ	95% CI	wins	q
BULLPEN-fused-full	BULLPEN-fused-full, shuffled	+0.068	[+0.03, +0.11]	37/46	<0.001
BULLPEN-fused-full	BULLPEN-fused-full, family-shuffled	+0.108	[+0.06, +0.17]	35/43	<0.001
BULLPEN-fused-full, shuffled	floor	+0.431	[+0.36, +0.50]	45/46	<0.001
BULLPEN-fused-full, family-shuffled	floor	+0.379	[+0.31, +0.45]	42/43	<0.001
BULLPEN-fused-full, shuffled	PCA32-bits	+0.063	[+0.02, +0.11]	31/46	0.005
BULLPEN-fused-full, family-shuffled	PCA32-bits	+0.030	[−0.00, +0.07]	26/43	0.101
BULLPEN-fused-full, shuffled	PCA32-bits (centered)	+0.043	[+0.01, +0.07]	30/46	0.017
BULLPEN-fused-full, family-shuffled	PCA32-bits (centered)	+0.008	[−0.04, +0.05]	24/43	0.243
BULLPEN-ans-full	BULLPEN-ans-full, shuffled	+0.458	[+0.40, +0.52]	45/46	<0.001
BULLPEN-ans-full	BULLPEN-ans-full, family-shuffled	+0.432	[+0.36, +0.50]	43/43	<0.001
BULLPEN-ans-full, shuffled	floor	−0.001	[−0.03, +0.02]	26/46	0.648
BULLPEN-ans-full, family-shuffled	floor	+0.016	[−0.02, +0.05]	24/43	0.164
BULLPEN-ans-full, shuffled	PCA32-bits	−0.380	[−0.44, −0.32]	1/46	<0.001
BULLPEN-ans-full, family-shuffled	PCA32-bits	−0.345	[−0.42, −0.27]	1/43	<0.001
BULLPEN-ans-full, shuffled	PCA32-bits (centered)	−0.401	[−0.46, −0.34]	1/46	<0.001
BULLPEN-ans-full, family-shuffled	PCA32-bits (centered)	−0.366	[−0.45, −0.29]	3/43	<0.001
BULLPEN-bits+ans-full	BULLPEN-bits+ans-full, shuffled	+0.027	[−0.01, +0.07]	25/46	0.472
BULLPEN-bits+ans-full	BULLPEN-bits+ans-full, family-shuffled	+0.053	[+0.01, +0.10]	28/43	0.033
BULLPEN-bits+ans-full, shuffled	floor	+0.448	[+0.38, +0.52]	44/46	<0.001
BULLPEN-bits+ans-full, family-shuffled	floor	+0.409	[+0.34, +0.48]	41/43	<0.001
BULLPEN-bits+ans-full, shuffled	PCA32-bits	+0.080	[+0.04, +0.13]	35/46	<0.001
BULLPEN-bits+ans-full, family-shuffled	PCA32-bits	+0.061	[+0.02, +0.10]	30/43	0.005
BULLPEN-bits+ans-full, shuffled	PCA32-bits (centered)	+0.060	[+0.02, +0.10]	33/46	0.004
BULLPEN-bits+ans-full, family-shuffled	PCA32-bits (centered)	+0.039	[−0.01, +0.09]	27/43	0.058
PCA32-bits+q+ans	PCA32-bits+q+ans, shuffled	+0.125	[+0.09, +0.17]	39/46	<0.001
PCA32-bits+q+ans	PCA32-bits+q+ans, family-shuffled	+0.120	[+0.08, +0.17]	36/43	<0.001
PCA32-bits+q+ans, shuffled	floor	+0.325	[+0.26, +0.39]	43/46	<0.001
PCA32-bits+q+ans, family-shuffled	floor	+0.319	[+0.26, +0.38]	43/43	<0.001
PCA32-bits+q+ans, shuffled	PCA32-bits	−0.043	[−0.08, −0.01]	13/46	0.001
PCA32-bits+q+ans, family-shuffled	PCA32-bits	−0.030	[−0.07, +0.02]	11/43	0.006
PCA32-bits+q+ans, shuffled	PCA32-bits (centered)	−0.064	[−0.09, −0.04]	8/46	<0.001
PCA32-bits+q+ans, family-shuffled	PCA32-bits (centered)	−0.052	[−0.09, −0.01]	9/43	<0.001
PCA32-ans	PCA32-ans, shuffled	+0.351	[+0.29, +0.41]	46/46	<0.001
PCA32-ans	PCA32-ans, family-shuffled	+0.357	[+0.29, +0.43]	42/43	<0.001
PCA32-ans, shuffled	floor	+0.005	[−0.01, +0.02]	25/46	0.614
PCA32-ans, family-shuffled	floor	−0.008	[−0.03, +0.02]	23/43	0.910
PCA32-ans, shuffled	PCA32-bits	−0.363	[−0.43, −0.30]	1/46	<0.001
PCA32-ans, family-shuffled	PCA32-bits	−0.357	[−0.42, −0.29]	1/43	<0.001
PCA32-ans, shuffled	PCA32-bits (centered)	−0.384	[−0.45, −0.32]	1/46	<0.001
PCA32-ans, family-shuffled	PCA32-bits (centered)	−0.379	[−0.45, −0.31]	1/43	<0.001
PCA32-bits+ans	PCA32-bits+ans, shuffled	+0.124	[+0.08, +0.17]	38/46	<0.001
PCA32-bits+ans	PCA32-bits+ans, family-shuffled	+0.126	[+0.08, +0.17]	38/43	<0.001
PCA32-bits+ans, shuffled	floor	+0.323	[+0.26, +0.39]	45/46	<0.001
PCA32-bits+ans, family-shuffled	floor	+0.310	[+0.25, +0.37]	43/43	<0.001
PCA32-bits+ans, shuffled	PCA32-bits	−0.044	[−0.08, −0.00]	13/46	<0.001
PCA32-bits+ans, family-shuffled	PCA32-bits	−0.039	[−0.08, +0.01]	8/43	0.002
PCA32-bits+ans, shuffled	PCA32-bits (centered)	−0.065	[−0.10, −0.03]	10/46	<0.001
PCA32-bits+ans, family-shuffled	PCA32-bits (centered)	−0.061	[−0.09, −0.03]	10/43	<0.001
BULLPEN-zConcat-budget	BULLPEN-zConcat-budget, shuffled	+0.142	[+0.10, +0.19]	37/43	<0.001
BULLPEN-zConcat-budget, shuffled	floor	+0.276	[+0.22, +0.34]	42/43	<0.001
BULLPEN-zConcat-budget, shuffled	PCA32-bits	−0.073	[−0.11, −0.04]	9/43	<0.001
BULLPEN-zConcat-budget, shuffled	PCA32-bits (centered)	−0.094	[−0.13, −0.06]	9/43	<0.001
BULLPEN-CCAfused-full	BULLPEN-CCAfused-full, shuffled	+0.150	[+0.10, +0.21]	37/43	<0.001
BULLPEN-CCAfused-full, shuffled	floor	+0.263	[+0.20, +0.33]	40/43	<0.001
BULLPEN-CCAfused-full, shuffled	PCA32-bits	−0.091	[−0.13, −0.06]	9/43	<0.001
BULLPEN-CCAfused-full, shuffled	PCA32-bits (centered)	−0.113	[−0.15, −0.08]	5/43	<0.001
BULLPEN-AE-full (bits+text)	BULLPEN-AE-full (bits + noise text)	+0.094	[+0.05, +0.13]	34/43	<0.001
BULLPEN-AE-full (text)	BULLPEN-AE-full (bits + noise text)	+0.062	[+0.02, +0.11]	27/43	0.032
BULLPEN-AE-full (bits + noise text)	floor	+0.289	[+0.22, +0.37]	39/43	<0.001
BULLPEN-AE-full (bits + noise text)	PCA32-bits	−0.059	[−0.10, −0.02]	14/43	0.013
BULLPEN-AE-full (bits + noise text)	PCA32-bits (centered)	−0.081	[−0.13, −0.03]	12/43	0.002
PCA32-bits	leaderboard scalar	−0.062	[−0.13, +0.02]	16/46	0.053
PCA32-bits (centered)	leaderboard scalar	−0.041	[−0.11, +0.03]	19/46	0.159

Per task under the strict protocol (every strict run: the 46 tasks of Table A6 where both encoders have a run that scores the three benchmark-transfer tasks, else the other 43; the wins column gives n. Moral Machine left out, UGI scored only when pooled). Each twin is the same encoder given another model's answers: shuffled, a derangement over all models; family-shuffled, only answers from another publisher family (scored on the tasks its run covers, see wins). Each block tests one encoder against its twins, then each twin against the floor and both PCA32-bits baselines. Δ: mean paired difference in skill (s; skill over floor against the floor row), with a 95% task bootstrap interval; wins: tasks where A scores higher; q: Benjamini–Hochberg within each bits reference's tests. Tasks are correlated ($M_{\text{eff}} \approx 37$ over the 46, ≈ 34 over the 43), so p and q are optimistic.

Table A8: Per-task skill of the bits, answer-text and joint embeddings.

task	group	bits	text	joint	shuffled	Δ joint − best single (strict)	both
bits-led							
per-question correctness	benchmark accuracy	0.51	0.36	0.46	0.44	–	
held-out benchmark	benchmark accuracy	0.67	0.65	0.71	0.74	–	
error type	benchmark accuracy	0.46	0.39	0.50	0.54	–	
item information	benchmark accuracy	0.68	0.52	0.68	0.80	–	
benchmark ranking	benchmark accuracy	0.77	0.64	0.78	0.80	–	
mean accuracy	benchmark accuracy	0.88	0.69	0.86	0.92	-0.00 [-0.00, -0.00]	
specialization	benchmark accuracy	0.13	0.09	0.12	0.06	–	
can-ai-code sr	public leaderboards	0.66	0.33	0.56	0.79	-0.16 [-0.32, -0.02]	
LiveBench language	public leaderboards	0.66	0.49	0.66	0.81	-0.26 [-0.51, -0.06]	
OLL v2 GPQA	public leaderboards	0.44	0.21	0.35	0.36	-0.14 [-0.29, +0.01]	
OLL v2 MATH	public leaderboards	0.49	0.45	0.57	0.50	-0.02 [-0.07, +0.03]	
Open LLM v1	public leaderboards	0.63	0.54	0.66	0.70	-0.09 [-0.15, -0.04]	
Open LLM v2	public leaderboards	0.53	0.30	0.45	0.57	-0.02 [-0.09, +0.06]	
Ayumi RP rank	human preference	0.42	0.37	0.48	0.47	-0.18 [-0.29, -0.08]	
architecture	model metadata	0.36	0.25	0.33	0.28	-0.08 [-0.17, -0.01]	
parameter count	model metadata	0.13	0.01	0.11	0.11	–	
query routing	model selection	0.42	0.25	0.44	0.50	–	
shared							
pairwise, per benchmark	benchmark accuracy	0.49	0.50	0.44	0.42	–	
reasoning tilt	benchmark accuracy	0.30	0.29	0.42	0.39	–	
Ayumi RP IQ	human preference	0.40	0.41	0.49	0.48	-0.20 [-0.32, -0.08]	
context length	model metadata	0.38	0.38	0.47	0.45	-0.05 [-0.10, -0.00]	
top-k selection	model selection	0.69	0.70	0.72	0.80	–	
text-led							
can-ai-code jr	public leaderboards	0.03	0.12	0.13	-0.07	+0.04 [-0.16, +0.20]	
IFEval	public leaderboards	0.25	0.35	0.24	0.31	+0.03 [-0.10, +0.09]	
Portuguese LLM	public leaderboards	0.03	0.15	0.26	0.52	-0.01 [-0.16, +0.15]	
LMArena rating	human preference	0.19	0.59	0.52	0.65	-0.11 [-0.29, +0.01]	
LMArena coding	human preference	0.28	0.60	0.59	0.60	-0.06 [-0.29, +0.05]	
style delta	response style	0.17	0.64	0.37	0.30	-0.17 [-0.34, -0.03]	
style delta, creative	response style	0.13	0.50	0.26	0.27	-0.11 [-0.35, +0.03]	
style delta, math	response style	0.31	0.74	0.54	0.40	-0.15 [-0.30, -0.02]	
answer divergence	answer behavior	0.32	0.90	0.72	0.35	–	
hedging	answer behavior	0.51	0.77	0.71	0.62	+0.09 [+0.01, +0.14]	
answer length	answer behavior	0.30	0.80	0.68	0.08	-0.02 [-0.09, +0.06]	
authorship	answer behavior	0.02	0.22	0.14	0.04	–	
refusal rate	answer behavior	0.02	0.31	0.20	-0.02	-0.18 [-0.29, -0.06]	
reward score	answer behavior	0.61	0.71	0.73	0.79	+0.01 [-0.03, +0.03]	
reward style	answer behavior	0.49	0.68	0.63	0.69	–	
probe refusal rate (LLMmap data)	safety	0.30	0.60	0.37	0.30	-0.01 [-0.28, +0.26]	
base vs chat	model metadata	0.42	0.52	0.47	0.30	-0.03 [-0.08, +0.02]	
model family	model metadata	0.30	0.41	0.41	0.18	+0.00 [+0.00, +0.00]	
downloads	model metadata	0.35	0.41	0.45	0.31	-0.07 [-0.12, -0.02]	
license	model metadata	0.14	0.25	0.17	0.30	-0.06 [-0.12, -0.02]	
release date	model metadata	0.54	0.64	0.63	0.59	-0.00 [-0.05, +0.05]	
Ollama quantizations	model metadata	0.18	0.28	0.24	0.27	+0.02 [-0.07, +0.10]	
serving providers	model metadata	0.20	0.27	0.27	0.17	-0.02 [-0.08, +0.04]	
budgeted selection	model selection	0.67	0.77	0.83	0.76	–	
bits-led (17 tasks), retention: joint / bits		0.99					
text-led (24 tasks), recovery: joint / text		0.86					
tasks that need both (strict)		0 of 31 tested					

Does a task need both channels? Per task, skill over floor (strict protocol, 5 publisher-grouped folds × 3 seeds, the 46 tasks of Table A6) of the bits (PCA32-bits (centered)), answer-text (BULLPEN-ans-full), joint (BULLPEN-bits+ans-full) and shuffled-joint (BULLPEN-bits+ans-full (shuffled text)) embeddings, grouped by which single channel leads by more than 0.02. Δ: joint minus the better single channel under model holdout, with its 95% model-bootstrap interval; – where no per-model label can be pooled. both: $\Delta > 0$ and joint > shuffled (Benjamini–Hochberg $q < 0.05$). Summary rows: the joint's share of bits on bits-led tasks (retention) and of text on text-led tasks (recovery).

H MODELS

Table A9: Models in the embedded pool.

All 260 models, by Hugging Face id. †: one of the 104 fullset models, the pool of the full-view K sweep (Figure 2b). Family: the publisher or model family that keeps a model's relatives in one fold. Params: billions of parameters. Date: month the repository was created. Fold: its fold of five under the strict protocol's first seed. –: not recorded.

model	family	params	date	fold	model	family	params	date	fold
01-ai/Yi-1.5-9B-Chat†	yi	9.0	2024-05	1	deepseek-ai/deepseek-math-7b-instruct†	deepseek-ai	7.0	2024-02	2
01-ai/Yi-6B-Chat	01-ai	6.1	2023-11	2	deepseek-ai/DeepSeek-R1-Distill-Llama-8B†	deepseek-r1	8.0	2025-01	5
Aashraf995/Creative-7B-nerd†	creative	7.6	2024-12	5	deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B†	qwen	0.8	2025-01	2
AdaptLLM/medicine-chat†	adaptllm	7.0	2023-12	2	dphn/dolphin-2.1-mistral-7b	dphn	7.2	2023-10	5
AdaptLLM/medicine-LLM†	adaptllm	7.0	2023-09	2	dphn/dolphin-2.2.1-mistral-7b	dphn	7.2	2023-10	5
AdaptLLM/medicine-LLM-13B†	adaptllm	13.0	2023-12	2	DreadPoor/Aspire-8B-model_stock†	DreadPoor	8.0	2024-09	4
ajibawa-2023/Uncensored-Frank-13B	ajibawa-2023	13.0	2023-09	3	EleutherAI/llemma_7b†	eleutherai	6.7	2023-09	3
ajibawa-2023/Uncensored-Frank-7B	ajibawa-2023	6.7	2023-09	3	EleutherAI/pythia-12b†	eleutherai	12.0	2023-02	3
ajibawa-2023/Uncensored-Jordan-13B	ajibawa-2023	13.0	2023-10	3	eren23/ogno-monarch-jaskier-merge-7b-OH-PREF-DPO†	eren23	7.0	2024-02	4
ajibawa-2023/Uncensored-Jordan-7B	ajibawa-2023	6.7	2023-10	3	EssentialAI/rnj-1-instruct	EssentialAI	8.3	2025-12	2
akjindal53244/Arithmo-Mistral-7B	akjindal53244	7.2	2023-10	3	FelixChao/llama2-13b-math1.2†	felixchao	13.0	2023-08	2
Alchan/mpt-7b-chat	Alchan	6.6	2024-03	3	FelixChao/Scorpio-7B†	felixchao	7.0	2024-02	2
allenai/OLMo-2-1124-7B-Instruct†	olmo	6.4	2024-12	3	FelixChao/vicuna-7B-chemical†	felixchao	7.0	2023-08	2
allenai/tulu-2-dpo-7b	allenai	6.7	2023-11	4	FelixChao/vicuna-7B-physics†	felixchao	7.0	2023-08	2
aloobun/TinyAiroboros-2.2.1	aloobun	1.1	2023-10	1	FPHam/Free_Sydney_V2_13b_HF	FPHam	13.0	2023-10	5
amazon/MistralLite	amazon	7.2	2023-10	5	Fredithefish/Guanaco-7B-Uncensored	Fredithefish	6.7	2023-09	1
ATH-MaaS/Marco-o1	marco	7.6	2024-11	3	golaxy/gowizardlm†	golaxy	7.0	2023-07	4
Austism/chronos-hermes-13b-v2	Austism	13.0	2023-08	3	google/gemma-1.1-2b-it	google	2.5	2024-03	5
baichuan-inc/Baichuan2-7B-Chat	baichuan-inc	7.5	2023-08	3	google/gemma-2-9b	google	9.2	2024-06	5
bardsai/jaskier-7b-dpo-v5.6†	bardsai	7.0	2024-02	1	google/gemma-2-9b-it†	gemma	9.2	2024-06	4
berkeley-nest/Starling-LM-7B-alpha†	berkeley-nest	7.0	2023-11	1	google/gemma-2b-it†	google	2.0	2024-02	5
Biomimicry-AI/ANIMA-Nectar-v2†	biomimicry-ai	7.0	2023-12	5	google/gemma-3-12b-it†	gemma	6.4	2025-03	4
BioMistral/BioMistral-7B†	biomistral	7.2	2024-02	1	google/gemma-3-1b-it†	gemma	0.4	2025-03	4
BioMistral/BioMistral-7B-DARE†	biomistral	7.0	2024-02	1	google/gemma-3-4b-it†	gemma	6.4	2025-02	4
BlueNipples/Trion-M-7b	BlueNipples	7.2	2023-11	2	google/gemma-7b-it†	gemma	8.5	2024-02	4
bofenghuang/vigostral-7b-chat	bofenghuang	7.2	2023-09	1	Gryphe/MythoLogic-13b	Gryphe	13.0	2023-07	1
Brouz/Slerpeno	Brouz	13.0	2023-09	1	Gryphe/MythoMax-L2-13b	Gryphe	13.0	2023-08	1
cais/zephyr_7b_r2d2	cais	7.2	2024-02	3	haoranxu/ALMA-7B-Pretrain	haoranxu	6.7	2023-09	4
CalderaAI/Naberius-7B	CalderaAI	7.2	2023-11	5	Harshvir/Llama-2-7B-physics†	harshvir	2.0	2023-08	2
cgato/Thespis-13b-v0.3	cgato	13.0	2023-10	4	Heralax/Augmental-13b	Heralax	13.0	2023-10	5
cgato/Thespis-13b-v0.4	cgato	13.0	2023-10	4	Heralax/Augmental-13b-v1.50_A	Heralax	13.0	2023-10	5
cgato/Thespis-13b-v0.5	cgato	13.0	2023-10	4	Heralax/Augmental-13b-v1.50_B	Heralax	13.0	2023-10	5
cgato/Thespis-13b-v0.6	cgato	13.0	2023-11	4	Heralax/MistralMakise-Merged-13b	Heralax	13.0	2023-10	5
cloudyu/Mixtral_11Bx2_MoE_19B†	cloudyu	19.0	2023-12	2	Heralax/MythoMakiseMerged-13b	Heralax	13.0	2023-09	5
codefuse-ai/CodeFuse-DeepSeek-33B†	codefuse-ai	33.0	2024-01	2	hfl/chinese-alpaca-2-7b	hfl	6.9	2023-07	1
codellama/CodeLlama-13b-Instruct-hf†	codellama	13.0	2023-08	3	HuggingFaceH4/zephyr-7b-alpha	HuggingFaceH4	7.2	2023-10	2
codellama/CodeLlama-7b-hf†	codellama	6.7	2023-08	3	HuggingFaceH4/zephyr-7b-beta†	huggingfaceh4	7.2	2023-10	5
codellama/CodeLlama-7b-Instruct-hf	codellama	6.7	2023-08	3	HuggingFaceTB/SmolLM2-1.7B-Instruct†	smollm2	1.2	2024-10	1
CorticalStack/pastiche-crown-clown-7b-dare-dpo†	corticalstack	7.0	2024-03	5	huggyllama/llama-13b	huggyllama	13.0	2023-04	4
cris177/llama-2-7b-Arguments	cris177	6.7	2023-10	4	ibm-granite/granite-3.0-2b-instruct	ibm-granite	2.6	2024-10	4
CultriX/NeuralTrix-bf16†	CultriX	7.2	2024-02	2	ibm-granite/granite-3.1-2b-instruct	ibm-granite	2.5	2024-12	4
Danielbrdz/Barcenas-Mistral-7b	Danielbrdz	7.2	2023-10	5	ibm-granite/granite-3.1-8b-instruct†	granite	8.1	2024-12	5
Dans-Archive/Dans-TotSirocco-7b	Dans-Archive	7.2	2023-10	3	IkariDev/Athena-v2	IkariDev	13.0	2023-09	3
deepseek-ai/deepseek-coder-6.7b-instruct†	deepseek-ai	6.7	2023-10	2	IkariDev/Athena-v3	IkariDev	13.0	2023-09	3

Table A9 (continued)

model	family	params	date	fold	model	family	params	date	fold
IkariDev/Athena-v4	IkariDev	13.0	2023-10	3	microsoft/Orca-2-13b†	microsoft	13.0	2023-11	4
IkariDev/Athnete-13B	IkariDev	13.0	2023-10	3	microsoft/Orca-2-7b	microsoft	6.7	2023-11	4
Intel/neural-chat-7b-v3-3†	intel	7.0	2023-12	3	microsoft/phi-1_5†	microsoft	1.0	2023-09	4
internlm/internlm-chat-7b	–	–	2023-07	2	microsoft/phi-2†	phi	2.8	2023-12	5
internlm/internlm3-8b-instruct†	internlm	9.7	2025-01	3	microsoft/Phi-3-mini-4k-instruct	phi	3.6	2024-04	5
itspat/RemmMistral13b	itspat	13.0	2023-11	2	microsoft/Phi-3-small-8k-instruct	phi	6.4	2024-05	5
jaspionjader/fr-18-8b†	fr	6.4	2025-01	3	microsoft/Phi-4-mini-instruct†	phi	3.6	2025-02	5
jebcarter/Psyfighter-13B	jebcarter	13.0	2023-11	4	microsoft/Phi-4-mini-reasoning†	phi	3.8	2025-04	5
jondurbin/airoboros-l2-13b-2.2.1	jondurbin	13.0	2023-09	3	migtissera/SynthIA-7B-v1.3	migtissera	7.2	2023-09	4
jondurbin/airoboros-l2-13b-3.0	jondurbin	13.0	2023-10	3	mistralai/Ministral-8B-Instruct-2410	mistralai	8.0	2024-10	3
jondurbin/airoboros-l2-13b-3.1.1	jondurbin	13.0	2023-10	3	mistralai/Mistral-7B-Instruct-v0.1†	mistral	7.2	2023-09	2
jondurbin/airoboros-l2-13b-gpt4-m2.0	jondurbin	13.0	2023-07	3	mistralai/Mistral-7B-Instruct-v0.2	mistralai	7.2	2023-12	3
jondurbin/airoboros-l2-7b-2.2.1	jondurbin	6.7	2023-09	3	mistralai/Mistral-7B-Instruct-v0.3†	mistral	6.4	2024-05	2
jondurbin/airoboros-l2-7b-3.0	jondurbin	6.7	2023-10	3	mistralai/Mistral-Small-24B-Instruct-2501†	mistral	23.6	2025-01	2
jondurbin/airoboros-m-7b-3.0	jondurbin	7.2	2023-10	3	mlabonne/AlphaMonarch-7B†	mlabonne	7.0	2024-02	3
jondurbin/airoboros-m-7b-3.1.2	jondurbin	7.2	2023-10	3	Neko-Institute-of-Science/metharme-7b†	neko-institute-of-science	7.0	2023-04	2
jondurbin/spicyboros-13b-2.2	jondurbin	13.0	2023-09	3	Neko-Institute-of-Science/pygmalion-7b†	neko-institute-of-science	7.0	2023-04	2
jpacifico/Chocolatine-3B-Instruct-DPO-Revised†	chocolatine	3.8	2024-07	4	NeverSleep/Echidna-13b-v0.1	NeverSleep	13.0	2023-10	4
jphme/em_german_13b_v01	jphme	13.0	2023-09	1	NeverSleep/Echidna-13b-v0.2	NeverSleep	13.0	2023-10	4
kevin009/llamaRAGdrama†	kevin009	7.0	2024-02	4	NeverSleep/Echidna-13b-v0.3	NeverSleep	13.0	2023-10	4
kfkas/Llama-2-ko-7b-Chat	kfkas	6.9	2023-07	5	NeverSleep/HornyEchidna-13b-v0.1	NeverSleep	13.0	2023-10	4
kingbri/airolima-chronos-grad-l2-13B	kingbri	13.0	2023-08	5	NeverSleep/Mistral-11B-SynthIAirOmniMix	NeverSleep	10.7	2023-10	4
kingbri/chronolima-airo-grad-l2-13B	kingbri	13.0	2023-08	5	NeverSleep/Nethena-13B	NeverSleep	13.0	2023-10	4
kyujinpy/Sakura-SOLRCA-Math-Instruct-DPO-v1†	kyujinpy	7.0	2023-12	1	Nexusflow/Starling-LM-7B-beta†	nexusflow	7.0	2024-03	4
LeoLM/leo-hessianai-7b-chat	LeoLM	6.7	2023-09	4	nomic-ai/gpt4all-13b-snoozy†	nomic-ai	13.0	2023-04	5
LeoLM/leo-mistral-hessianai-7b-chat	LeoLM	7.2	2023-10	4	Norquinal/Mistral-7B-claude-chat	Norquinal	7.2	2023-09	4
lesubra/ECE-PRYMMAL-3B-SLERP-V1†	ece	3.8	2024-10	3	Norquinal/PetrolLM	Norquinal	7.2	2023-09	4
LGAI-EXAONE/EXAONE-3.5-7.8B-Instruct†	exaone	6.4	2024-12	3	NousResearch/Nous-Capybara-7B-V1.9	NousResearch	7.2	2023-10	5
Lil-R/2_PRYMMAL-ECE-7B-SLERP†	2	7.6	2024-11	2	NousResearch/Nous-Hermes-13b†	nousresearch	13.0	2023-06	5
LilRg/10PRYMMAL-3B-slerp†	10prymmal	3.8	2024-09	3	NousResearch/Nous-Hermes-Llama2-13b	NousResearch	13.0	2023-07	5
lmsys/vicuna-13b-v1.3	lmsys	13.0	2023-06	4	NousResearch/Yarn-Mistral-7b-64k	NousResearch	7.2	2023-10	5
lmsys/vicuna-7b-v1.3	lmsys	6.7	2023-06	4	oh-yeontaek/llama-2-7B-LoRA-assemble	oh-yeontaek	6.7	2023-09	1
lmsys/vicuna-7b-v1.5†	lmsys	6.7	2023-07	4	Open-Orca/Mistral-7B-OpenOrca	Open-Orca	7.2	2023-09	5
lmsys/vicuna-7b-v1.5-16k†	lmsys	7.0	2023-07	4	openbmb/UltraLM-13b-v2.0	openbmb	13.0	2023-09	2
ludis/tsukasa-limarp-7b	ludis	6.7	2023-09	3	OpenBuddy/openbuddy-codellama2-34b-v11.1-bf16†	OpenBuddy	33.8	2023-09	3
maywell/Synatra-7B-v0.3-RP	maywell	7.2	2023-10	1	OpenBuddy/openbuddy-mistral-7b-v13	OpenBuddy	7.3	2023-10	3
meta-llama/Llama-2-13b-chat-hf†	llama2	13.0	2023-07	1	OpenBuddy/openbuddy-zephyr-7b-v14.1	OpenBuddy	7.3	2023-11	3
meta-llama/Llama-2-7b-chat-hf†	llama2	6.7	2023-07	1	openchat/openchat-3.5-0106†	openchat	3.5	2024-01	2
meta-llama/Llama-3.1-8B-Instruct†	llama3	8.0	2024-07	3	openchat/openchat-3.5-1210	openchat	7.2	2023-12	2
meta-llama/Llama-3.2-1B-Instruct†	llama3	1.2	2024-09	3	openchat/openchat_3.5†	openchat	3.5	2023-10	2
meta-llama/Llama-3.2-3B-Instruct†	llama3	3.2	2024-09	3	PharMolix/BioMedGPT-LM-7B†	pharmolix	7.0	2023-08	5
meta-llama/Meta-Llama-3-8B-Instruct†	meta-llama	3.0	2024-04	2	pipizhao/Pandalyst-7B-V1.1	pipizhao	6.7	2023-09	2
meta-math/MetaMath-Llemma-7B†	meta-math	7.0	2023-11	3	pipizhao/Pandalyst-7B-V1.2	pipizhao	6.7	2023-10	2
meta-math/MetaMath-Mistral-7B†	meta-math	7.0	2023-10	3	posicube/Llama2-chat-AYB-13B	posicube	13.0	2023-10	4

Table A9 (continued)

model	family	params	date	fold	model	family	params	date	fold
posicube/Llama2-chat-AYT-13B	posicube	13.0	2023-09	4	TIGER-Lab/MAmmoTH-7B	TIGER-Lab	6.7	2023-09	4
princeton-nlp/gemma-2-9b-it-SimPO	princeton-nlp	9.2	2024-07	1	tiiuae/falcon-40b-instruct†	tiiuae	40.0	2023-05	3
project-baize/baize-v2-13b†	project-baize	13.0	2023-05	4	tiiuae/Falcon3-10B-Instruct†	falcon	4.5	2024-12	3
PygmalionAI/mythalion-13b	PygmalionAI	13.0	2023-09	5	tiiuae/Falcon3-7B-Instruct†	falcon	3.2	2024-11	3
PygmalionAI/pygmalion-2-13b	PygmalionAI	13.0	2023-09	5	TinyLlama/TinyLlama-1.1B-Chat-v1.0†	tinyllama	1.1	2023-12	1
Q-bert/Optimus-7B†	q-bert	7.0	2023-12	1	Tostino/Inkbot-13b-4k	Tostino	13.0	2023-09	5
QuixiAI/samantha-mistral-instruct-7b	QuixiAI	7.2	2023-09	5	Undi95/Amethyst-13B	Undi95	13.0	2023-09	1
QuixiAI/Wizard-Vicuna-13B-Uncensored	QuixiAI	13.0	2023-05	5	Undi95/Emerhyst-13B	Undi95	13.0	2023-09	1
QuixiAI/Wizard-Vicuna-7B-Uncensored	QuixiAI	6.7	2023-05	5	Undi95/Mistral-11B-Airoboros-RP-v1	Undi95	10.7	2023-10	1
QuixiAI/WizardLM-7B-Uncensored	QuixiAI	6.7	2023-05	5	Undi95/Mistral-ClaudeLimaRP-v3-7B	Undi95	7.2	2023-09	1
Qwen/Qwen-7B-Chat	Qwen	7.7	2023-08	5	Undi95/Mistral-PetroLimaRP-v3-12B	Undi95	11.2	2023-09	1
Qwen/Qwen1.5-0.5B-Chat†	qwen	0.5	2024-01	2	Undi95/MistRP-Airoboros-7B	Undi95	7.2	2023-10	1
Qwen/Qwen1.5-4B-Chat†	qwen	4.0	2024-01	2	Undi95/MLewdBoros-LRPSGPT-2Char-13B	Undi95	13.0	2023-09	1
Qwen/Qwen1.5-7B-Chat†	qwen	7.0	2024-01	2	Undi95/Nete-13B	Undi95	13.0	2023-10	1
Qwen/Qwen2-0.5B-Instruct	Qwen	0.5	2024-06	5	Undi95/OpenRP-13B	Undi95	13.0	2023-09	1
Qwen/Qwen2-1.5B-Instruct	Qwen	1.5	2024-06	5	Undi95/OpenRP-13B-SuperCOT	Undi95	13.0	2023-09	1
Qwen/Qwen2-7B-Instruct	Qwen	7.6	2024-06	5	Undi95/PsyMedRP-v1-13B	Undi95	13.0	2023-10	1
Qwen/Qwen2.5-0.5B-Instruct†	qwen2.5	0.5	2024-09	2	Undi95/ReMM-Lion-13B	Undi95	13.0	2023-09	1
Qwen/Qwen2.5-1.5B-Instruct†	qwen	0.8	2024-09	2	Undi95/ReMM-S-Kimiko-v2-13B	Undi95	13.0	2023-09	1
Qwen/Qwen2.5-3B-Instruct†	qwen	1.8	2024-09	2	Undi95/ReMM-v1-LRPSGPT-2Char-13B	Undi95	13.0	2023-09	1
Qwen/Qwen2.5-7B-Instruct	qwen	4.3	2024-09	2	Undi95/ReMM-v2-Kimiko-v2-13B	Undi95	13.0	2023-09	1
Qwen/Qwen2.5-Coder-7B-Instruct†	qwen	4.3	2024-09	2	Undi95/UndiMix-v1-13b	Undi95	13.0	2023-08	1
Qwen/Qwen2.5-Math-7B-Instruct†	qwen	4.3	2024-09	2	Undi95/UndiMix-v2-13b	Undi95	13.0	2023-09	1
Qwen/Qwen3-4B†	qwen	2.8	2025-04	2	Undi95/UndiMix-v3-13B	Undi95	13.0	2023-09	1
Riiid/sheep-duck-llama-2-13b	Riiid	13.0	2023-10	5	Undi95/UndiMix-v4-13B	Undi95	13.0	2023-09	1
rishiraj/CatPPT-base†	rishiraj	7.0	2023-12	3	Undi95/Unholy-v1-10L-13B	Undi95	13.0	2023-09	1
royallab/Pygmalion-2-13b-SuperCOT2	royallab	13.0	2023-09	4	Undi95/Unholy-v1-12L-13B	Undi95	13.0	2023-09	1
sail/Sailor2-8B-Chat†	sailor	4.9	2024-12	4	Undi95/Unholy-v1.1-13B	Undi95	13.0	2023-09	1
SciPhi/SciPhi-Mistral-7B-32k†	sciphi	7.0	2023-10	1	Undi95/Utopia-13B	Undi95	13.0	2023-11	1
SciPhi/SciPhi-Self-RAG-Mistral-7B-32k†	sciphi	7.0	2023-10	1	Undi95/Xwin-MLewd-13B-V0.2	Undi95	13.0	2023-10	1
shleeeee/mistral-ko-tech-science-v1†	shleeeee	7.0	2023-12	1	Undi95/ZettaPi-13B	Undi95	13.0	2023-09	1
SkunkworksAI/Mistralic-7B-1	–	–	2023-09	2	upstage/SOLAR-10.7B-Instruct-v1.0†	upstage	10.7	2023-12	1
stabilityai/StableBeluga-13B	stabilityai	13.0	2023-07	4	vanillaOVO/WizardCoder-Python-7B-V1.0	vanillaOVO	6.7	2024-06	4
stabilityai/StableBeluga-7B	stabilityai	6.7	2023-07	4	Weyaxi/SlimOpenOrca-Mistral-7B	Weyaxi	7.2	2023-10	5
stabilityai/stablelm-tuned-alpha-7b†	stabilityai	7.0	2023-04	4	WizardLM/WizardLM-13B-V1.2	WizardLM	13.0	2023-07	2
teknium/CollectiveCognition-v1.1-Mistral-7B	teknium	7.2	2023-10	2	XiaomiMiMo/MiMo-7B-RL†	mimo	7.6	2025-04	4
teknium/Mistral-Trismegistus-7B	teknium	7.2	2023-10	2	Xwin-LM/Xwin-LM-13B-V0.2	Xwin-LM	13.0	2023-10	4
teknium/OpenHermes-2-Mistral-7B	teknium	7.2	2023-10	2	yam-peleg/Experiment26-7B†	yam-peleg	7.0	2024-02	2
teknium/OpenHermes-2.5-Mistral-7B	teknium	7.2	2023-10	2	zai-org/agentlm-13b	zai-org	13.0	2023-10	5
TheBloke/guanaco-7B-HF	TheBloke	6.7	2023-05	2	zai-org/agentlm-7b	zai-org	6.7	2023-10	5
TheBloke/Llama-2-70B-Chat-AWQ†	thebloke	70.0	2023-09	5	zai-org/chatglm3-6b	zai-org	6.2	2023-10	5
THUDM/chatglm2-6b	–	–	–	2	zai-org/glm-4-9b-chat†	glm	8.1	2024-06	1