JUDGE2VEC: SELECTING COMPACT LLM JURIES FROM UNLABELED JUDGMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLMs are increasingly used as judges because human evaluation does not scale, yet relying on a single judge can be unreliable and querying many judges is costly. We introduce Judge2Vec, an unsupervised framework that selects a compact jury from a large, heterogeneous pool using only the judges' multi-criterion ratings on unlabeled pilot examples. Rather than ranking judges by a single reliability score, Judge2Vec represents each judge with a criterion-specific competence embedding estimated through leave-one-family-out agreement with a family-balanced cross-family consensus, then composes a jury whose strengths cover different criteria while avoiding redundant judges. Remarkably, selecting fewer judges can improve both quality and efficiency: across eight text benchmarks spanning qualitative and quantitative evaluation, Judge2Vec achieves the highest performance among all compared methods, while four selected judges outperform full-pool aggregation and reduce average inference cost by 83.9%. Our approach also extends to multimodal and agentic evaluation. Beyond a fixed jury, the learned competence representations support budget-adaptive deployment and transfer across tasks, enabling jury construction for unseen tasks without any target-task examples or judge evaluations. These results show that LLM evaluation need not choose between a single unreliable judge and an expensive large jury: a small jury, intelligently composed from unlabeled judgments, can be both more reliable and more efficient.

1 INTRODUCTION

As large language models (LLMs) become more capable and widely deployed, evaluating their outputs reliably has become increasingly important (Chen et al., 2026). Human evaluation provides a trusted reference, but it is costly, slow, and difficult to scale to the volume of outputs produced by modern LLM systems. This has driven the growing use of LLMs themselves as evaluators: LLM judges assess generated outputs in tasks such as summarization and dialogue, and provide preference signals for reward modeling and reinforcement learning from AI feedback (Bai et al., 2022).

Relying on a single LLM judge can be unreliable: judge quality varies substantially across tasks and evaluation criteria (Verga et al., 2024; Qian et al., 2026) (Appendix A). Using multiple judges can improve reliability, but querying many LLMs for every case is costly. More importantly, supervision is scarce at both levels: human or ground-truth labels for the cases to be judged are unavailable at scale, which is the very reason for using LLM judges, and labels for assessing which LLMs are reliable judges are even less available. This raises our central question: *Given a sizable, heterogeneous pool of LLM judges, can we select a small jury using only their judgments on unlabeled pilot examples, while preserving the evaluation quality of a much larger pool?*

Importantly, using more judges is not necessarily better. Judges differ in reliability, and models from the same family may exhibit correlated behavior. Aggregating the entire pool can therefore dilute strong judgments with weaker ones and overrepresent redundant judgments from similar models. A carefully selected jury can potentially be not only cheaper, but also more reliable than the full pool.

We introduce **Judge2Vec**, an unsupervised framework for this problem. The key idea is to use a small set of unlabeled pilot examples to reveal how candidate judges differ before selecting a jury. Rather than reducing each judge's assessment of a case to a single overall score, we ask every judge to score the same cases along multiple evaluation criteria. These criterion-level judgments expose different strengths and weaknesses that an overall score can hide: a judge may be strong on correctness but

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.

weak on coherence, while another shows the opposite pattern. Judge2Vec summarizes these behaviors as criterion-aware competence profiles and selects a small set of judges whose strengths complement one another. Only the selected jury is then queried at deployment.

The main technical challenge is that the pilot examples have no human or ground-truth labels: we must estimate which judges are reliable using only the judgments themselves. Judge2Vec addresses this challenge by representing each judge with a *criterion-aware competence embedding*. For each evaluation criterion, we measure how well a judge's rankings of the pilot examples agree with a family-balanced consensus of other model families. We exclude the judge's entire model family from its reference, yielding *leave-one-family-out (LOFO) agreement*. This prevents closely related models from directly reinforcing one another, while family balancing prevents model families with more judges from dominating the consensus. The resulting embedding captures where a judge appears strong or weak across evaluation criteria, rather than reducing judge reliability to a single score.

Judge2Vec then treats jury construction as a problem of *composition rather than ranking*. Selecting the individually strongest judges can produce a redundant jury: several judges may be strong on the same criteria while sharing the same weaknesses. Instead, our selection algorithm favors judges that strengthen the jury's weakest criterion while avoiding redundant competence profiles. Thus, a judge can be valuable not because it is the best judge overall, but because it complements the judges already selected. Selection requires only the unlabeled pilot judgments; no ground-truth labels are used. At deployment, only the selected judges evaluate new items, with ratings aggregated by criterion.

This formulation enables two extensions. First, because LLMs differ in inference cost, we introduce a cost-aware replacement procedure that substitutes expensive jury members with cheaper alternatives while minimizing quality loss. This produces configurations for different deployment budgets. Second, we investigate whether judge competence learned on previous tasks can be reused for a new task. By aligning criterion-aware representations across tasks, Judge2Vec can construct a jury for an unseen task without using any target-task examples, even unlabeled ones.

Our experiments show that a small jury selected from unlabeled pilot judgments can outperform substantially costly alternatives. Across eight datasets spanning qualitative and quantitative evaluation, Judge2Vec achieves the highest performance among all compared methods. It uses only four of 19 candidate judges while outperforming methods that query the full pool. The resulting jury costs \$4.67 per 1,000 evaluated items on average, compared with \$28.97 for the full 19-judge pool, an 83.9% reduction in inference cost. Judge2Vec also lies on the observed quality–cost Pareto frontier in multimodal evaluation. Finally, zero-shot cross-task transfer retains evaluation quality close to Judge2Vec constructed from an in-domain target pilot, despite using no target-task examples.

Contributions. We introduce a new paradigm for scalable LLM evaluation: rather than choosing a single judge or querying a large jury, we show that a *small, carefully composed jury can be both cheaper and more reliable*, even without human or ground-truth labels for jury selection. Judge2Vec exploits *multi-criterion judgments on unlabeled pilot examples* to learn criterion-aware competence profiles from leave-one-family-out agreement with a family-balanced consensus, exposing complementary strengths while mitigating correlated behavior within model families. This turns jury construction from *ranking individual judges* into *composing complementary judges*: our selector seeks broad competence coverage while avoiding weak or redundant judges that can degrade full-pool aggregation. The resulting compact juries achieve the highest performance among the baselines across all eight datasets while substantially reducing inference cost, showing that intelligent selection can improve both quality and efficiency. Beyond a fixed jury, the same competence representations support budget-adaptive deployment and transfer to unseen tasks without target-task examples, enabling scalable evaluation across qualitative, quantitative, multimodal, and agentic settings.

2 Related Work

LLM judges and juries. LLMs are increasingly used as automated evaluators to score or compare generated responses (Liu et al., 2023; Zheng et al., 2023), but individual judges exhibit biases and task- and criterion-specific weaknesses (Verga et al., 2024; Qian et al., 2026), motivating multi-judge approaches. PoLL (Verga et al., 2024) averages judgments from smaller models across families to reduce cost and model-specific bias; RoPoLL (Acharya et al., 2026) uses geometric-median aggregation to limit the influence of faulty or biased judges; CARE (Zhao et al., 2026) separates

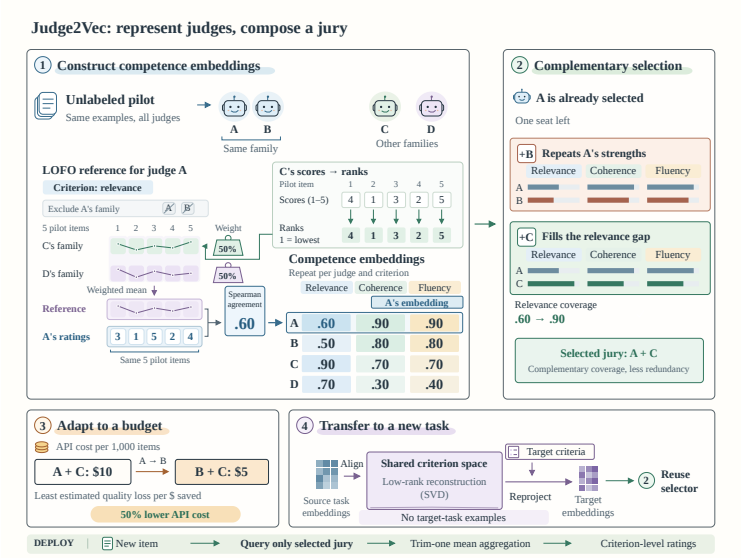


Figure 1: **Overview of Judge2Vec.** Candidate judges score a shared unlabeled pilot set on multiple criteria. Judge2Vec uses these judgments to construct criterion-aware competence embeddings and select a compact jury with complementary strengths. Only the selected jury is queried at deployment. The same representations also support budget adaptation and cross-task transfer.

shared judge biases from differences in answer quality; and BT-σ (Qian et al., 2026) extends the Bradley-Terry model (Hunter, 2004) to jointly infer response rankings and judge reliability from pairwise comparisons. These methods primarily improve how a given set of judges is aggregated. Judge2Vec instead asks which judges should constitute the jury, using unsupervised, criterion-aware competence representations to select a compact set of judges with complementary strengths.

Crowdsourcing and annotator reliability. Crowdsourcing methods address noisy or inconsistent human annotations by estimating annotator reliability and aggregating judgments to infer latent labels. Dawid–Skene (Dawid & Skene, 1979), MACE (Hovy et al., 2013), and GLAD (Whitehill et al., 2009) are representative approaches. Applied to LLM evaluation, these methods can treat candidate judges as annotators, but require querying the entire candidate pool on each new item before aggregation. Judge2Vec instead uses a small set of unlabeled pilot items to select a compact jury with complementary criterion-specific strengths, so only the selected judges are queried at deployment.

Expert team formation and complementary selection. Expert team formation selects candidates with multidimensional skill profiles whose complementary strengths collectively cover task requirements, often under communication, workload, or cost constraints (Lappas et al., 2009; Anagnostopoulos et al., 2012). Judge2Vec shares this principle of complementary selection but the judges' competence profiles are unknown. It thus first infers criterion-specific competence from unlabeled judgments and then uses these learned profiles to compose a complementary jury.

A more detailed discussion of related work is provided in Appendix B.

3 Judge2Vec: Unsupervised LLM Jury Construction

The goal of Judge2Vec is to replace a large, heterogeneous pool of LLM judges with a small jury that is both effective and inexpensive to deploy, without using human or ground-truth labels for jury construction. The key resource is a small *unlabeled pilot set*: before deployment, all candidate judges evaluate the same pilot examples on multiple criteria. These multi-criterion judgments reveal criterion-specific strengths and weaknesses that would be hidden by a single overall score.

Judge2Vec constructs a jury from unlabeled pilot items in two stages. It first estimates criterion-specific competence through family-balanced cross-family agreement (Section 3.2), then selects a jury that balances criterion coverage and redundancy (Section 3.3). Figure 1 summarizes the framework.

3.1 Problem Setup

Let $\mathcal{J} = \{1, \ldots, J\}$ denote a pool of J candidate LLM judges, belonging to G disjoint model families $\{\mathcal{F}_1, \ldots, \mathcal{F}_G\}$; $f(j) = g$ indicates that judge j belongs to family $\mathcal{F}_g$. Models within a family may share training, architecture, or lineage and thus exhibit correlated evaluation behavior.

Before jury selection, every candidate judge evaluates the same M pilot examples on D criteria. Importantly, the pilot is *unlabeled*: neither human judgments nor task-specific ground truth is available to the selection procedure. Let $\mathcal{O} \subseteq \mathcal{J} \times \{1, \ldots, M\} \times \{1, \ldots, D\}$ contain the judge–item–criterion triples for which a numeric rating is available, and let $T_{j,i,d} \in \mathbb{R}$ denote judge j's rating of pilot item i on criterion d. Missing or unparseable ratings are excluded from $\mathcal{O}$.

Given the pilot ratings, $\mathcal{O}$, the family assignments f, and a desired jury size $1 \leq K < J$, our goal is to select a jury $\mathcal{S} \subset \mathcal{J}$, $|\mathcal{S}| = K$, whose members collectively provide strong coverage across the D criteria without redundant competence profiles. Section 3.3 operationalizes this goal through a greedy selection utility that balances weakest-criterion coverage and redundancy. After selection, only judges in $\mathcal{S}$ are queried on new test items. Thus, the full pool is used only to characterize candidate judges on the small pilot set, while the recurring deployment cost is incurred only by the selected jury.

3.2 Criterion-Aware Competence Representation

A central challenge is to estimate judge competence without knowing the correct evaluation of any pilot example. Judge2Vec uses *cross-family agreement* as an unsupervised signal: a judge is more credible on a criterion when its ordering of pilot examples agrees with the consensus of independently represented model families. We estimate this agreement separately for every criterion, rather than collapsing a judge's behavior into one overall reliability score.

Rank-normalized judgments. Because different LLMs may use rating scales differently, we compare their rankings of pilot examples rather than raw scores. For judge j and criterion d, let $n_{j,d} = |\{i : (j, i, d) \in \mathcal{O}\}|$ be the number of available ratings, and let $r_{j,i,d}$ be the rank of $T_{j,i,d}$ among them, with higher ratings receiving higher ranks and ties assigned average ranks. We normalize the ranks as $\widetilde{T}_{j,i,d} = \frac{r_{j,i,d}}{n_{j,d}+1}$, placing judgments from different judges on a common scale.

Family-balanced leave-one-family-out consensus. Agreement among closely related models is not independent evidence of competence: models from the same family may share systematic biases. Therefore, when estimating judge j's competence, we exclude its entire family from the reference. For item i and criterion d, define $\mathcal{R}_{j,i,d} = \{q \in \mathcal{J} : f(q) \neq f(j),\ (q, i, d) \in \mathcal{O}\}$, and let $\mathcal{G}_{j,i,d} = \{f(q) : q \in \mathcal{R}_{j,i,d}\}$ be the represented reference families.

Simply averaging all judges in $\mathcal{R}_{j,i,d}$ would give larger families greater influence. Instead, we first average normalized ranks within each family and then weight the families equally, giving the cross-family consensus $\bar{s}_{-f(j),i,d} = \frac{1}{|\mathcal{G}_{j,i,d}|} \sum_{g \in \mathcal{G}_{j,i,d}} \left(\frac{1}{|\mathcal{R}_{j,i,d} \cap \mathcal{F}_g|} \sum_{q \in \mathcal{R}_{j,i,d} \cap \mathcal{F}_g} \widetilde{T}_{q,i,d} \right)$. This *leave-one-family-out (LOFO)* construction prevents a judge from reinforcing itself through related models and prevents larger families from receiving disproportionate weight.

Criterion-specific competence. For judge j and criterion d, let $\mathcal{I}_{j,d} = \{i : (j, i, d) \in \mathcal{O},\ |\mathcal{R}_{j,i,d}| > 0\}$ contain pilot items with both the judge rating and an out-of-family reference. We define

$$\rho_{j,d} = \rho_{\mathrm{S}}\big((T_{j,i,d})_{i \in \mathcal{I}_{j,d}}, (\bar{s}_{-f(j),i,d})_{i \in \mathcal{I}_{j,d}}\big), \qquad C_{j,d} = \max\{0, \rho_{j,d}\}, \tag{1}$$

where ρ_{S} is Spearman rank correlation. Thus, $C_{j,d}$ measures whether judge j and the cross-family consensus order the pilot items similarly on criterion d, independent of the exact rating scale.

Stacking the criterion-specific coordinates gives the Judge2Vec embedding

$$\mathbf{c}_j = (C_{j,1}, \ldots, C_{j,D}) \in [0, 1]^D, \qquad \mathbf{C} = [\mathbf{c}_1^\top; \ldots; \mathbf{c}_J^\top] \in [0, 1]^{J \times D}. \tag{2}$$

Each coordinate has a direct interpretation: $C_{j,d}$ estimates judge j's competence on criterion d from its agreement with the cross-family consensus. The resulting embedding therefore captures a judge's strengths and weaknesses across evaluation criteria rather than collapsing its competence into a single score. This criterion-aware representation is crucial for jury construction: two judges with similar

Algorithm 1 Complementary jury selection

Require: Competence matrix $\mathbf{C}$; jury size K
Ensure: Ordered jury $\mathcal{S}$ of K judges
$\mathcal{S} \leftarrow ()$
$j^\star \leftarrow \arg\max_{j \in \mathcal{J}} \min_{1 \le d \le D} C_{j,d}$
Append $j^\star$ to $\mathcal{S}$
while $|\mathcal{S}| < K$ **do**
  for each $j \in \mathcal{J} \setminus \mathcal{S}$ **do**
    $q_j \leftarrow g(j \mid \mathcal{S})$
  end for
  $j^\star \leftarrow \arg\max_{j \in \mathcal{J} \setminus \mathcal{S}} q_j$
  Append $j^\star$ to $\mathcal{S}$
end while
return $\mathcal{S}$

Algorithm 2 Cost-aware jury replacement

Require: Initial jury $\mathcal{S}_0$; competence matrix $\mathbf{C}$; judge costs $c_1, \ldots, c_J$
Ensure: Juries with decreasing deployment cost
$t \leftarrow 0$
while a cheaper replacement exists **do**
  $(a^\star, b^\star) \leftarrow \arg\min_{a \in \mathcal{S}_t,\, b \in \mathcal{J} \setminus \mathcal{S}_t,\, c_b < c_a} r(a \to b \mid \mathcal{S}_t)$
  $\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t$
  Replace $a^\star$ by $b^\star$ in $\mathcal{S}_{t+1}$
  $t \leftarrow t + 1$
end while
return $\mathcal{S}_0, \ldots, \mathcal{S}_t$

overall reliability may have different strengths across criteria, while judges from different model families may nevertheless have redundant competence profiles.

3.3 Complementary Jury Selection

The competence embeddings turn jury construction from a problem of *ranking judges individually* into *composing judges collectively*. Selecting the top-K judges by an overall score can be redundant: several strong judges may cover the same criteria while sharing the same weaknesses. Conversely, a judge that is not the strongest individually may be valuable if it fills a weakness of the current jury.

For a nonempty jury $\mathcal{S}$, let $\mathrm{cov}_d(\mathcal{S}) = \max_{p \in \mathcal{S}} C_{p,d}$ denote its coverage of criterion d. For an unselected judge j, we measure both the jury's weakest-criterion coverage after adding j and the redundancy of j with the existing jury:

$$\mathrm{cover}(j \mid \mathcal{S}) = \min_{1 \le d \le D} \max\{C_{j,d}, \mathrm{cov}_d(\mathcal{S})\}, \qquad \mathrm{sim}(j \mid \mathcal{S}) = \max\left\{0, 1 - \frac{\min_{p \in \mathcal{S}} \|\mathbf{c}_j - \mathbf{c}_p\|_2}{\max_{a,b \in \mathcal{J}} \|\mathbf{c}_a - \mathbf{c}_b\|_2}\right\}.$$

The coverage term follows a max–min principle, rewarding candidates that strengthen the jury's weakest criterion; the similarity term is larger when j has a profile close to one already in the jury.

We combine the two through the selection utility

$$g(j \mid \mathcal{S}) = \frac{\mathrm{cover}(j \mid \mathcal{S})}{1 + \mathrm{sim}(j \mid \mathcal{S})}. \tag{3}$$

A candidate therefore receives high utility when it improves the jury's weakest competence while contributing a nonredundant profile.

Algorithm 1 implements this criterion through greedy jury construction. It initializes the jury with the judge having the strongest worst-criterion competence, $s_1 = \arg\max_{j \in \mathcal{J}} \min_d C_{j,d}$, and subsequently adds the unselected judge with the highest $g(j \mid \mathcal{S})$ until K judges have been selected.

This selection mechanism explains why a compact jury can, in principle, outperform the full candidate pool rather than merely approximate it at lower cost. Aggregating every available judge gives weak or redundant judges a voice simply because they are present. Judge2Vec instead uses the pilot set to filter the pool and compose a jury whose members provide complementary criterion-level competence. Selection can therefore improve both *evaluation quality* and *efficiency*.

3.4 Jury Aggregation

Once the jury is selected, only its members evaluate deployment items. For item i and target criterion d, let $T^{\mathrm{dep}}_{p,i,d}$ denote judge p's rating and let $\mathcal{S}_{i,d} \subseteq \mathcal{S}$ contain the judges with an available rating.

For ordinal and continuous criteria, let $v_{(1)} \le \cdots \le v_{(n_{i,d})}$ denote the sorted ratings, where $n_{i,d} = |\mathcal{S}_{i,d}|$. To reduce the influence of isolated extreme judgments, we use the *trim-one mean*

$\hat{y}_{i,d} = \sum_{k=2}^{n_{i,d}-1} v_{(k)}/(n_{i,d}-2)$ for $n_{i,d} > 2$, removing one rating from each extreme. With one or two available ratings, we use their arithmetic mean. Aggregation is performed independently for each target criterion, preserving the criterion-aware structure throughout the evaluation pipeline.

4 Extending Judge2Vec Across Budgets and Tasks

A useful judge representation should support more than a single fixed jury. In this section, we show that the same competence embeddings can be reused in two ways: *across budgets*, by replacing expensive jury members with cheaper alternatives while preserving quality, and *across tasks*, by transferring competence from source tasks to new tasks without target-task examples.

4.1 Cost-Aware Jury Adaptation

Selecting a compact jury already avoids the recurring cost of querying the full candidate pool, but deployment budgets may vary. To accommodate different budget constraints, we progressively replace expensive members with cheaper judges that preserve the jury's competence and complementarity.

Let $\mathcal{S}_0 = \mathcal{S}$ be the jury returned by Algorithm 1, and let $\mathcal{S}_t$ denote the jury after t replacements. For judge j, let $c_j > 0$ be its inference cost, and $\kappa(\mathcal{S}_t) = \sum_{j \in \mathcal{S}_t} c_j$. For a selected judge $a \in \mathcal{S}_t$ and a cheaper candidate $b \in \mathcal{J} \setminus \mathcal{S}_t$ with $c_b < c_a$, the estimated utility sacrifice per unit of cost saved is

$$r(a \to b \mid \mathcal{S}_t) = \frac{g(a \mid \mathcal{S}_t \setminus \{a\}) - g(b \mid \mathcal{S}_t \setminus \{a\})}{c_a - c_b}, \tag{4}$$

where g is defined in Equation 3. A smaller ratio indicates a better quality–cost tradeoff.

Algorithm 2 repeatedly chooses the feasible replacement with the smallest ratio in Equation 4, producing fixed-size juries $\mathcal{S}_0, \mathcal{S}_1, \ldots$ with strictly decreasing cost. Given a budget B, a practitioner can choose a configuration satisfying $\kappa(\mathcal{S}_t) \le B$. Adaptation uses only the competence embeddings and judge costs, requiring no human labels, task-reference labels, or additional judge evaluations.

4.2 Zero-Shot Cross-Task Jury Transfer

Although Section 3 requires no labeled pilot data, constructing a jury for a new task still requires querying the full candidate pool on an unlabeled pilot set. We ask whether even this step can be avoided by transferring judge competence constructed from previous tasks. The challenge is that different tasks may use different evaluation criteria. We address this through three steps: align source-task competence embeddings in a shared criterion space, extract their common low-rank structure, and reproject the resulting representation onto the target task's criteria. This yields a target-task competence matrix without using any target-task examples or judge evaluations.

Source-criterion alignment. Suppose we have L source tasks. For task ℓ with D_ℓ criteria, let $\mathbf{C}_\ell \in \mathbb{R}^{J \times D_\ell}$ contain the competence embeddings of the same J judges. Inspired by the task-agnostic evaluation dimensions in FLASK (Ye et al., 2024), we define a shared criterion space with $D_U = 10$ dimensions (Appendix C). We prompt Claude Fable 5 to assign nonnegative semantic-alignment weights $\mathbf{W}_\ell \in \mathbb{R}^{D_\ell \times D_U}$ between each source criterion and the shared criteria. After row normalization, $\mathbf{A}_\ell = \mathrm{RowNorm}(\mathbf{W}_\ell)$, $\mathbf{P}_\ell = \mathbf{C}_\ell \mathbf{A}_\ell \in \mathbb{R}^{J \times D_U}$ express judge competence from different tasks aligned into common criterion coordinates.

Shared low-rank judge representation. We concatenate the aligned matrices as $\mathbf{P} = [\mathbf{P}_1 \mid \cdots \mid \mathbf{P}_L] \in \mathbb{R}^{J \times LD_U}$ and apply rank-r_{svd} truncated SVD, yielding $\widetilde{\mathbf{P}} = \mathbf{U}_{r_{\mathrm{svd}}} \mathbf{\Sigma}_{r_{\mathrm{svd}}} \mathbf{V}_{r_{\mathrm{svd}}}^\top$. This focuses the representation on dominant variation across the source tasks. We partition $\widetilde{\mathbf{P}}$ into blocks $\widetilde{\mathbf{P}}_\ell \in \mathbb{R}^{J \times D_U}$ and average them to obtain the shared judge representation $\bar{\mathbf{P}} = \frac{1}{L} \sum_{\ell=1}^{L} \widetilde{\mathbf{P}}_\ell \in \mathbb{R}^{J \times D_U}$.

Target-criterion reprojection. For an unseen target task with D_{tgt} criteria, we construct semantic-alignment weights $\mathbf{W}_{\mathrm{tgt}} \in \mathbb{R}^{D_{\mathrm{tgt}} \times D_U}$ using the same procedure and define $\mathbf{B}_{\mathrm{tgt}} = \mathrm{RowNorm}(\mathbf{W}_{\mathrm{tgt}})^\top$. We then reproject the shared representation into the target criterion space as $\widehat{\mathbf{C}}_{\mathrm{tgt}} = \bar{\mathbf{P}} \mathbf{B}_{\mathrm{tgt}} \in \mathbb{R}^{J \times D_{\mathrm{tgt}}}$. Each row of $\widehat{\mathbf{C}}_{\mathrm{tgt}}$ is a predicted criterion-aware competence profile for a

Table 1: **Evaluation datasets.**

Qualitative datasets	Domain	D
HANNA (Chhun et al., 2024)	Story generation	6
SummEval (Fabbri et al., 2021)	Summarization	4
HelpSteer2 (Wang et al., 2024c)	Instruction following	5
FED (Mehri & Eskenazi, 2020)	Dialogue	8
Target: human/expert rating.		

Quantitative datasets	Domain	Target	D
MATH (Hendrycks et al., 2021)	Mathematical reasoning	C	4
ExpertQA (Malaviya et al., 2024)	Long-form QA	A	11
HAGRID (Kamalloo et al., 2023)	Long-form QA	A	4
TheAgentCompany (Xu et al., 2026a)	Agentic task completion	P	4
MathVista (Lu et al., 2024)	Visual mathematics	C	4

Target denotes the held-out ground-truth reference used to evaluate the selected jury. Quantitative targets: C = answer correctness; A = human/expert attribution; P = checkpoint completion. D is the number of criteria used to construct the Judge2Vec embedding.

candidate judge on the unseen task. Algorithm 1 then selects a complementary jury directly from these profiles. Thus, competence learned from previous tasks becomes reusable: jury construction for a new task requires neither target-task examples nor target-task judge evaluations.

5 Experiments

We evaluate Judge2Vec along three dimensions: **RQ1** evaluation quality and generality across tasks and modalities; **RQ2** deployment efficiency and quality–cost tradeoffs; and **RQ3** zero-shot transfer to unseen tasks without target-task examples.

5.1 Experimental Setup

Datasets. We evaluate on four *qualitative* datasets, where open-ended outputs are evaluated against human ratings, and five *quantitative* datasets with task-correctness or task-completion labels (Table 1).

For every task, all candidate judges score the same unlabeled pilot examples. Reference labels are used only for held-out evaluation and never for jury construction. We repeat jury selection over five data splits. For qualitative datasets, we use the dataset's built-in evaluation criteria for both competence estimation and held-out evaluation. Quantitative datasets only has a single target criterion, so we supplement it with task-relevant auxiliary criteria for competence estimation and jury selection; evaluation remains against the dataset's original target. Criterion definitions are provided in Appendix D. Held-out test sets contain 175–1,200 items; dataset sizes are reported in Appendix E.

Judge pools and scoring. The text pool contains 19 judges from six model families: Claude (Opus 4.7, Sonnet 4.6, Haiku 4.5), Llama (4 Maverick, 3.3 70B, 3.1 8B), Mistral (Mixtral 8x7B, Magistral Small 2509, Ministral 14B/8B, Mistral 7B), DeepSeek (V3.2, V3.1), Qwen (Qwen3 235B/80B/32B), and Gemma (Gemma 3 27B/12B/4B). The multimodal study uses a separate seven-judge pool supporting both image and text inputs. All judges receive the same scoring prompt and evaluation criteria and are queried at temperature zero. Further details are in Appendix F.

Metrics. For qualitative tasks, we report Kendall τ between predictions and human ratings. For quantitative tasks, we report accuracy on MATH and MathVista, Kendall τ against human attribution judgments on ExpertQA and HAGRID, and against checkpoint completion on TheAgentCompany.

Baselines. We compare with LLM-jury, panel-selection, and annotator-aggregation baselines. LLM-jury methods include PoLL (Verga et al., 2024), RoPoLL (Acharya et al., 2026), CARE (Zhao et al., 2026), and BT-σ (Qian et al., 2026). Panel-selection baselines include the largest representative from each model family, the largest 3 or 5 judges, and random panels of size 3 or 5. Annotator-aggregation baselines include Dawid–Skene (Dawid & Skene, 1979), MACE (Hovy et al., 2013), and GLAD (Whitehill et al., 2009). Full-pool aggregation methods use all 19 text judges.

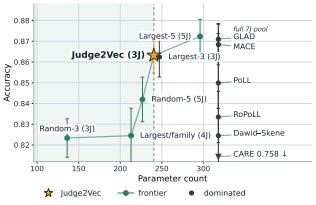
Implementation details. The jury size is $K = 4$. Pilot sizes are 200 for FED, 100 for MATH, 150 for MathVista, and 400 for other datasets. Both jury size and pilot size are selected using only unlabeled pilot judgments; the selection procedures and implementation details are given in Appendix F.

5.2 Evaluation Quality and Generality

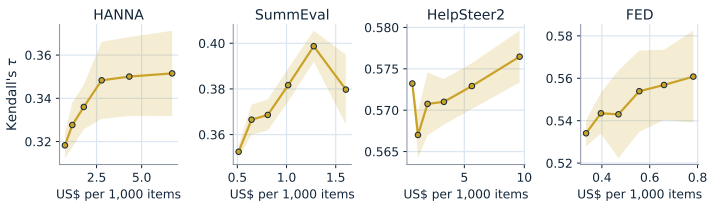
Qualitative evaluation. Table 2 shows that Judge2Vec ranks first on all qualitative datasets. Relative to the strongest baseline on each dataset, it improves Kendall τ by 0.042, 0.100, 0.008, and 0.021 on HANNA, SummEval, HelpSteer2, and FED, respectively, for an average gain of 0.043. Its mean

Table 2: **Qualitative and quantitative evaluation.** Qualitative tasks report Kendall τ against held-out human labels; quantitative tasks use the metric in each column. Mean±population standard deviation across 5 splits. Bold and underlined denote the top two results per column. † denotes a significant improvement over the strongest baseline under paired bootstrap testing (10,000 resamples, $p < 0.05$). BT-σ requires paired responses to the same input, and thus "–" denotes unavailability.

	Qualitative				Quantitative			
Method	HANNA τ	SummEval τ	HelpSteer2 τ	FED τ	MATH Acc.	ExpertQA τ	HAGRID τ	AgentCompany τ
Judge2Vec	**.352±.020**†	**.380±.015**†	**.576±.003**	**.561±.022**†	**.860±.019**†	**.201±.027**†	**.226±.027**	**.336±.020**
Aggregation methods (full 19-judge pool)								
PoLL	.310±.009	.278±.003	.568±.000	.540±.009	.804±.008	.171±.005	.213±.026	.330±.014
RoPoLL	.307±.008	.272±.003	.553±.000	.529±.008	.783±.009	.183±.010	.191±.027	.319±.015
CARE	.266±.081	.258±.094	.557±.000	.503±.048	.805±.017	.105±.013	.083±.067	.310±.030
BT-σ	.144±.015	.242±.003	–	–	–	–	–	–
Dawid–Skene	.303±.010	.272±.005	.564±.002	.534±.011	.814±.011	.171±.006	.212±.026	.327±.011
MACE	.286±.007	.280±.008	.564±.000	.533±.009	.837±.017	.166±.006	.212±.025	.331±.014
GLAD	.259±.007	.256±.009	.559±.000	.531±.009	.839±.010	.166±.008	.210±.025	.323±.014
Panel-selection methods								
Largest-per-family	.286±.006	.270±.003	.561±.000	.540±.010	.836±.012	.163±.006	.199±.024	.327±.015
Largest-3	.251±.007	.223±.005	.552±.000	.511±.011	.491±.022	.129±.013	.199±.032	.312±.015
Largest-5	.260±.008	.238±.004	.561±.000	.515±.011	.710±.009	.149±.010	.209±.025	.309±.013
Random-3	.297±.008	.270±.003	.553±.001	.536±.007	.775±.005	.150±.005	.189±.023	.326±.014
Random-5	.303±.008	.274±.003	.560±.001	.539±.008	.793±.008	.156±.005	.199±.024	.328±.014



(a) MathVista Pareto frontier.



(b) Budget-adaptive jury replacement.

Figure 2: **(a) Accuracy–cost tradeoff on MathVista.** Judge2Vec lies on the observed Pareto frontier and achieves the highest accuracy among methods with an equal or smaller parameter count. **(b) Budget-adaptive jury replacement.** Starting from the initial Judge2Vec jury, our algorithm progressively substitutes cheaper judges, producing jury configurations at different deployment costs. Across tasks, evaluation quality typically degrades gradually as cost decreases.

Kendall τ of 0.467 exceeds the mean of the baselines (0.405) by 15.3%, and it improves over the strongest full-pool aggregation baseline by 10.1%. These results support our central premise: adding more judges does not necessarily improve quality when the additional judges are weak or redundant.

Quantitative evaluation. Judge2Vec also generalizes to quantitative evaluation. As shown in Table 2, the selected jury achieves the highest result on all four datasets. Relative to the second-best method, Judge2Vec improves MATH accuracy by 0.021 and Kendall τ by 0.018, 0.013, and 0.005 on ExpertQA, HAGRID, and TheAgentCompany, respectively, corresponding to relative gains of 2.5%, 9.8%, 6.1%, and 1.5%. Averaged across the four datasets, Judge2Vec improves over the available baselines by 15.7%. The consistent gains across correctness, attribution, and task-completion targets suggest that the competence representation is useful beyond subjective human-rated evaluation.

Generalization across tasks and judges. The results also show that judge quality is highly task-dependent: methods that perform well on one benchmark can be substantially weaker on another. For example, PoLL is competitive on HANNA, HelpSteer2, FED, and HAGRID but performs substantially worse on MATH. In contrast, Judge2Vec remains the strongest method across all datasets.

Multimodal evaluation. We further test whether the selection principle extends beyond text-only judging using MathVista with a separate seven-judge multimodal pool. Figure 2a shows that Judge2Vec lies on the observed accuracy–cost Pareto frontier and achieves the highest accuracy among configurations with an equal or smaller total parameter count. This result indicates that complementary jury construction remains useful when judges must jointly process images and text.

Table 3: **Deployment cost.**
US$ per 1,000 judged items

Method	K	HANNA	SummEval	HelpSteer2	FED	Mean
Judge2Vec	4	6.69 ±5.20	**1.60** ±**.23**	9.61 ±.14	**.78** ±**.21**	4.67
19-judge	19	36.68	27.70	36.67	14.82	28.97
Largest-3	3	17.46	13.82	19.00	7.46	14.44
Largest-5	5	27.04	20.80	27.80	11.01	21.66
Random-3	3	**5.71**±**5.97**	4.31±4.67	**5.70**±**6.40**	2.30±2.46	**4.50**
Random-5	5	9.47±7.21	7.15±5.64	9.46±7.72	3.83±2.98	7.48

Table 4: **Zero-shot cross-task transfer.**
Kendall τ

Method	HANNA	SummEval	HelpSteer2	FED
PoLL (all 19)	.310±.009	.278±.003	.568±.000	.540±.009
Zero-shot transfer (no target examples, even unlabeled ones)				
Judge2Vec	**.358**±**.015**	.356±.014	.571±.006	.556±.017
In-domain reference: uses target pilot examples				
Target pilot	.352±.020	**.380**±**.015**	**.576**±**.003**	**.561**±**.022**

Cost: Measured token usage and Amazon Bedrock on-demand rates. Full 19-judge panel includes PoLL, RoPoLL, CARE, BT-σ, Dawid–Skene, MACE, and GLAD. *Mean* denotes the average across the four tasks. Bold and underlined values denote the lowest and second-lowest cost in each column, respectively. *Transfer:* The in-domain reference denotes Judge2Vec constructed from the target task's own pilot.

5.3 Cost-Aware Deployment

Deployment cost. Judge2Vec reduces deployment cost substantially by querying only the selected jury. Table 3 reports the API cost of judging 1,000 items, computed from measured token usage and Amazon Bedrock on-demand rates. The selected four-judge jury costs $4.67 on average across the four qualitative datasets, compared with $28.97 for the full 19-judge pool, corresponding to an 83.9% reduction. Judge2Vec is only slightly more expensive than Random-3 ($4.50), but achieves higher evaluation quality on every dataset, while costing 3.1–$4.6\times$ less than Largest-3 and Largest-5. Thus, the savings do not come simply from using fewer judges: selecting judges by complementary competence yields a stronger quality–cost tradeoff than full-pool aggregation or other methods.

Budget-adaptive replacement. Starting from the initial jury, the procedure in Section 4.1 progressively replaces expensive judges with cheaper alternatives according to the estimated quality sacrifice per dollar saved, producing a sequence of fixed-size juries with decreasing cost. As shown in Figure 2b, evaluation quality typically degrades gradually as cost decreases rather than collapsing abruptly. For example, on HelpSteer2, deployment cost can be reduced by roughly $15\times$, from 9.61 to 0.66 per 1,000 judged items, while Kendall τ decreases by only 0.003. This provides a practical way to adapt a selected jury to different deployment budgets without additional judge evaluations.

5.4 Zero-Shot Cross-Task Transfer

We test whether judge competence can be reused for an unseen task without any target-task examples. We hold out each task in turn and transfer judge representations from the remaining source tasks. As shown in Table 4, zero-shot Judge2Vec outperforms PoLL on all target tasks without using any target-task examples, even unlabeled ones. Compared with the in-domain reference, the differences in Kendall τ are $+0.006$, -0.024, -0.005, and -0.005 on HANNA, SummEval, HelpSteer2, and FED, respectively. Averaged across the tasks, zero-shot transfer achieves Kendall $\tau = 0.460$, compared with 0.467 for the in-domain reference, retaining 98.5% of its average performance. The small gap suggests that criterion-aware representations learned from prior tasks capture reusable structure in judge competence, enabling effective jury construction even when no target-task pilot is available.

6 Conclusion

In this paper, we introduced Judge2Vec to rethink LLM jury construction as a problem of *composition rather than scale*: evaluation quality depends not on how many judges are queried, but on whether their strengths are complementary. By exploiting multi-criterion judgments on unlabeled pilot examples, Judge2Vec identifies such complementary competence without requiring the human supervision that automated evaluation is intended to replace. Our results challenge the intuitive assumption that more judges yield better evaluation: carefully composed small juries can outperform much larger pools while substantially reducing inference cost. More broadly, criterion-aware competence representations make judge expertise a reusable resource that can be adapted across deployment budgets and transferred to new tasks. As LLMs increasingly provide feedback for model training, data curation, benchmarking, and agent evaluation, the ability to characterize and compose machine evaluators without human labels offers a promising foundation for scalable and reliable AI evaluation.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing for clarity and readability, coding, literature review, and the refinement of LaTeX figures and tables. We also used an LLM to construct the semantic criterion-alignment weights used in the cross-task transfer experiment, as described in Section 4.2. All AI-assisted outputs, including code, literature claims, and manuscript text, were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work follows the ICLR Code of Ethics. It uses existing public benchmark datasets and model-generated evaluations; no new human participants were recruited and no new human-subject study was conducted. LLM judges may inherit social, cultural, or linguistic biases from their training data, and multiple judges may exhibit correlated errors. Judge2Vec is designed to reduce reliance on any single evaluator by estimating criterion-specific competence, excluding a judge's own model family from its LOFO reference, and selecting complementary juries. These mechanisms do not guarantee correctness, fairness, or freedom from shared failure modes. Accordingly, automated juries should not replace appropriate human oversight in high-stakes or consequential decisions. When used for applications such as reward modeling, data filtering, model selection, benchmarking, or agent evaluation, practitioners should audit task-specific and subgroup behavior, retain mechanisms for human review and correction, and comply with applicable dataset licenses, model terms of use, and legal requirements.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the experimental protocol throughout the paper and appendices. Sections 3–4 specify the Judge2Vec competence construction, jury-selection, aggregation, cost-adaptation, and cross-task-transfer procedures, including the corresponding algorithms and equations. Section 5.1 documents the datasets, judge pools and scoring protocol, evaluation metrics, baselines, jury size, and pilot sizes used in the experiments. Appendix C defines the shared criterion space used for transfer; Appendix D gives the complete evaluation and auxiliary criteria; Appendix E reports the held-out test-set sizes; and Appendix F provides additional implementation details, including the procedures for selecting jury and pilot sizes. Together, these sections specify the experimental setup and methodological choices underlying the reported results.

REFERENCES

Anish Acharya, Kris W Pan, and Brian Verkhovsky. Ropoll: Robust panel of llm judges. *arXiv preprint arXiv:2606.30931*, 2026.

Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web*, pp. 839–848, 2012.

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

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longwriter: Unleashing 10,000+ word generation from long context llms. In *International Conference on Learning Representations*, volume 2025, pp. 36528–36546, 2025.

Tianlong Chen, Jian Pei, Minxing Zhang, and Liang Zhao. Trustworthy llm-based agents for data-centric ai. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 13299–13303, 2026.

Cyril Chhun, Fabian M. Suchanek, and Chloé Clavel. Do language models enjoy their own stories? Prompting large language models for automatic story evaluation. *Transactions of the Association*

for Computational Linguistics, 12:1122–1142, 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00689. URL https://doi.org/10.1162/tacl_a_00689.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28 (1):20–28, 1979.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*, 2025.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th conference of the european chapter of the association for computational linguistics: system demonstrations*, pp. 150–158, 2024.

Alexander R. Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics (TACL)*, 9:391–409, 2021.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 7(6), 2026.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. Learning whom to trust with mace. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 1120–1130, 2013.

David R Hunter. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1): 384–406, 2004.

Michinori Jinji, Kyohei Atarashi, Koh Takeuchi, and Hisashi Kashima. MAPLE: Multi-aspect panels of LLM evaluators for open-ended questions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 27071–27088, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1351. URL https://aclanthology.org/2026.findings-acl.1351/.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv preprint arXiv:2307.16883*, 2023.

Mehdi Kargar, Morteza Zihayat, and Aijun An. Finding affordable and collaborative teams from a network of experts. In *Proceedings of the 2013 SIAM international conference on data mining*, pp. 587–595. SIAM, 2013.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353, 2024.

Theodoros Lappas, Kun Liu, and Evimaria Terzi. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 467–476, 2009.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, et al. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2757–2791, 2025a.

Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. Calibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16537–16552, 2025b.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 2511–2522, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. Expertqa: Expert-curated questions and attributed answers. 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. 3025–3045, 2024.

Shikib Mehri and Maxine Eskenazi. Unsupervised evaluation of interactive dialog with DialoGPT. In Olivier Pietquin, Smaranda Muresan, Vivian Chen, Casey Kennington, David Vandyke, Nina Dethlefs, Koji Inoue, Erik Ekstedt, and Stefan Ultes (eds.), *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 225–235, 1st virtual meeting, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.sigdial-1.28. URL https://aclanthology.org/2020.sigdial-1.28/.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 12076–12100, 2023.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802, 2024.

Mengjie Qian, Guangzhi Sun, Mark JF Gales, and Kate M Knill. Who can we trust? llm-as-a-jury for comparative assessment. *arXiv preprint arXiv:2602.16610*, 2026.

Mahdis Saeedi, Hawre Hosseini, Christine Wong, and Hossein Fani. A survey of subgraph optimization for expert team formation. *ACM Computing Surveys*, 57(12):1–40, 2025.

Guangzhi Sun, Anmol Kagrecha, Potsawee Manakul, Phil Woodland, and Mark Gales. SkillAggregation: Reference-free LLM-dependent aggregation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15532–15548, 2025. doi: 10.18653/v1/2025.acl-long.756. URL https://aclanthology.org/2025.acl-long.756/.

Haochen Tan, Zhijiang Guo, Zhan Shi, Lu Xu, Zhili Liu, Yunlong Feng, Xiaoguang Li, Yasheng Wang, Lifeng Shang, Qun Liu, et al. Proxyqa: An alternative framework for evaluating long-form text generation with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6806–6827, 2024.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. In *International Conference on Learning Representations (ICLR)*, 2025.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9440–9450, 2024a.

Xinyu Wang, Zhou Zhao, and Wilfred Ng. Ustf: A unified system of team formation. *IEEE Transactions on Big Data*, 2(1):70–84, 2016.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, et al. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. In *International Conference on Learning Representations*, volume 2024, pp. 43573–43593, 2024b.

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Parag Shah, Yujia Bao, et al. Mcp-bench: Benchmarking tool-using llm agents with complex real-world tasks via mcp servers. In *International Conference on Learning Representations*, volume 2026, pp. 97377–97425, 2026.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh N Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, 37:1474–1501, 2024c.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems*, 22, 2009.

Frank Fangzheng Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *Advances in Neural Information Processing Systems*, 38, 2026a.

Mingyuan Xu, Xinzi Tan, Jiawei Wu, and Doudou Zhou. A judge-aware ranking framework for evaluating large language models without ground truth. *arXiv preprint arXiv:2601.21817*, 2026b.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. Flask: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations*, volume 2024, pp. 55361–55414, 2024.

Minxing Zhang, Yi Yang, Zhuofan Jia, Xuan Yang, Jian Pei, Yuchen Zang, Xingwang Deng, and Xianglong Chen. Mpceval: A benchmark for multi-party conversation generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 10258–10267, 2026a.

Minxing Zhang, Yi Yang, Roy Xie, Bhuwan Dhingra, Shuyan Zhou, and Jian Pei. Generalizability of large language model-based agents: A comprehensive survey. *ACM Computing Surveys*, 58(10): 1–44, 2026b.

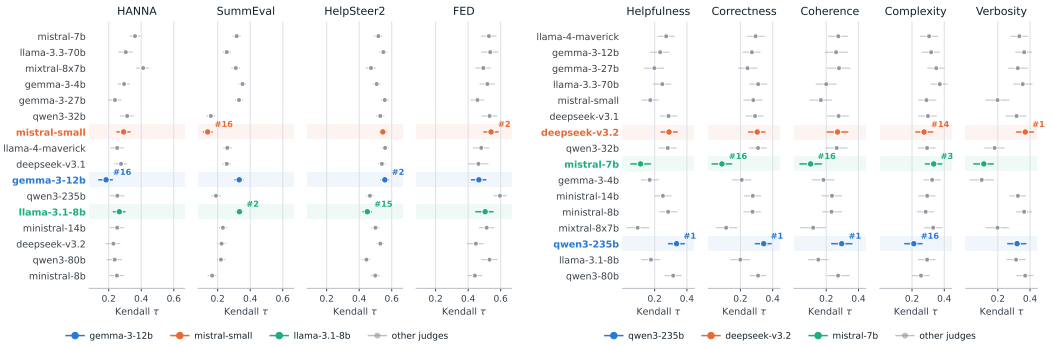
Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(102): 1–44, 2016.

Jitian Zhao, Changho Shin, Tzu-Heng Huang, Satya Sai Srinath Namburi GNVV, and Frederic Sala. Care: Confounder-aware aggregation for reliable llm evaluation. *arXiv preprint arXiv:2603.00039*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. In *International Conference on Learning Representations (ICLR)*, 2025.



(a) Variation across tasks. **(b) Variation across evaluation dimensions.**

Figure 3: **Judge quality varies across tasks and evaluation dimensions.** **(a)** Agreement with human annotations for the same 16 judges across HANNA, SummEval, HelpSteer2, and FED. **(b)** Agreement with human annotations for the same judges across the five HelpSteer2 evaluation dimensions. Within each subfigure, rows use a fixed judge order, horizontal bars denote 95% bootstrap confidence intervals over test items, and labels such as #2 indicate that the judge ranks second among the 16 judges in that setting. Highlighted judges illustrate some of the largest observed rank changes.

A Judge Quality Varies Across Tasks and Evaluation Dimensions

Setup. We measure how well each individual judge agrees with human annotations using the same held-out test split and Kendall τ metric as in the main experiments. The main experiments use the full 19-judge candidate pool. For this per-model diagnostic, we report 16 judges spanning five model families: Llama (3), Mistral (5), Qwen (3), Gemma (3), and DeepSeek (2). The remaining three judges are omitted only from this diagnostic because of external model-comparison disclosure constraints; this omission is unrelated to their performance. For *task-level* agreement, we flatten all item–dimension pairs within a task and compute Kendall τ between the judge and human reference scores, yielding one value per judge per task. For *dimension-level* agreement, we compute Kendall τ over the test items using only the scores for that dimension, yielding one value per judge per dimension. Error bars denote 95% bootstrap confidence intervals from 1,000 resamples of the test items.

Findings. Judge quality is not stable across tasks or evaluation dimensions. Across tasks (Figure 3(a)), the same judge can move by as many as 14 rank positions out of 16. Gemma-3-12B ranks second on HelpSteer2 but last on HANNA, while Mistral Small ranks second on FED but last on SummEval. The variation is also substantially larger than the bootstrap uncertainty: on HANNA, SummEval, and HelpSteer2, the between-judge range in Kendall τ is 2.5–4.4$\times$ the mean confidence-interval width.

The same heterogeneity appears across evaluation dimensions within a single task (Figure 3(b)). On HelpSteer2, Qwen3-235B ranks first among all 16 judges on helpfulness, correctness, and coherence, but last on complexity. Mistral-7B ranks last on correctness and coherence but third on complexity.

These results motivate our criterion-aware competence embeddings: a single scalar notion of "judge quality" cannot represent a judge whose relative strength changes substantially across tasks or from one evaluation criterion to another.

B Extended Related Work

As large language models (LLMs) become more capable and widely deployed (Zhang et al., 2026b), they are increasingly used not only to generate outputs but also to evaluate them, motivating a rapidly growing literature on LLM-based evaluation. Recent surveys provide broad overviews of evaluation protocols, judge reliability, bias, and applications (Li et al., 2025a; Gu et al., 2026). We focus on the lines of work most relevant to Judge2Vec: individual LLM evaluators and meta-evaluation, judge bias and calibration, multi-judge aggregation and LLM juries, rating-based and preference-based judging,

crowdsourcing and annotator reliability, multi-criterion evaluation, and expert team formation and complementary selection.

LLM evaluators and meta-evaluation. Early LLM-as-a-judge work established that strong language models can provide useful reference-free evaluations of open-ended generation. G-Eval uses criterion-specific prompting and chain-of-thought reasoning to improve alignment with human evaluation (Liu et al., 2023), while MT-Bench and Chatbot Arena systematically study strong LLMs as pairwise judges and characterize their agreement with human preferences (Zheng et al., 2023). Other work develops models specifically for evaluation. PandaLM trains a dedicated judge for comparing instruction-tuned models (Wang et al., 2024b); JudgeLM fine-tunes scalable judge models and studies position, knowledge, and format biases (Zhu et al., 2025); and Prometheus 2 supports both direct assessment and pairwise ranking under user-specified evaluation criteria (Kim et al., 2024). JudgeBench instead focuses on meta-evaluation, constructing challenging response pairs with objective preference labels to measure the capabilities of LLM judges themselves (Tan et al., 2025).

These works primarily improve or assess an individual evaluator. Judge2Vec addresses a different problem: given a heterogeneous pool of available judges, it characterizes their criterion-specific capabilities and selects a compact subset whose strengths complement one another. This distinction is particularly important because individual judges can exhibit systematic biases as well as task- and criterion-specific weaknesses, motivating selection across heterogeneous evaluators rather than reliance on a single judge.

Judge bias and calibration. This concern is reinforced by work showing that LLM judgments can depend on factors unrelated to response quality. MT-Bench identifies position, verbosity, and self-enhancement biases (Zheng et al., 2023). Wang et al. study positional bias and proposes calibration through response-order swapping (Wang et al., 2024a). Panickssery et al. show that evaluators can recognize and favor their own generations (Panickssery et al., 2024), while CalibraEval introduces an unsupervised inference-time calibration method for reducing position- and token-related selection bias in pairwise judging (Li et al., 2025b).

These studies expose important reliability limitations of individual LLM judges, motivating the use of multiple judges, or juries, as discussed below. Judge2Vec follows this direction but focuses on how such a jury should be constructed: it represents each candidate judge with a criterion-specific competence embedding and then selects a compact, complementary jury in an unsupervised manner. The goal is to improve evaluation reliability by producing aggregate judgments that better align with human annotations or task-specific reference targets across both qualitative and quantitative evaluation.

Multi-judge aggregation and LLM juries. Given the reliability limitations of individual judges, recent work increasingly turns to multiple evaluators, or LLM juries. PoLL shows that panels composed of models from different families can outperform a single large judge while reducing cost and model-specific bias (Verga et al., 2024). RoPoLL studies robustness of such panels and replaces standard aggregation with a geometric-median estimator to reduce the effect of biased or corrupted judges (Acharya et al., 2026). SkillAggregation learns how to combine predictions from multiple LLM judges without reference labels (Sun et al., 2025). CARE explicitly models shared latent confounders among judges, separating latent response quality from correlated effects such as verbosity or stylistic preferences (Zhao et al., 2026).

Other work models heterogeneous judge reliability in comparative evaluation, where judges compare candidate responses and indicate which one is preferred. BT-σ extends the Bradley–Terry model (Hunter, 2004) with a judge-specific discriminator and jointly infers response rankings and judge reliability from pairwise comparisons without human labels (Qian et al., 2026). Related judge-aware ranking formulations similarly introduce judge-specific discrimination parameters to account for heterogeneous reliability (Xu et al., 2026b). MAPLE extends multi-evaluator aggregation to multiple evaluation aspects by estimating both criterion importance and evaluator reliability from criterion-wise pairwise judgments (Jinji et al., 2026).

The primary objective of these methods is aggregation or latent reliability estimation after the judges have been selected. That is, they study how to combine judge outputs once the jury is already formed, or assume that the entire candidate pool is queried on every evaluation item. PoLL

additionally motivates cross-family panel diversity, but does not infer a criterion-specific competence representation and optimize panel composition in that space in an unsupervised manner. Judge2Vec instead focuses on the practical deployment decision: it learns a multi-dimensional competence embedding for each candidate from an unlabeled pilot and selects a compact jury before test-time evaluation, balancing weakest-criterion coverage against redundancy among competence profiles. Because only this small but competent and complementary jury is queried on subsequent evaluation items, rather than the full candidate pool, Judge2Vec reduces deployment cost while maintaining high evaluation quality.

Rating-based and preference-based judging. LLM evaluation commonly uses either *rating-based judging*, which assigns scalar or Likert-style scores to individual outputs (Liu et al., 2023), or *preference-based judging*, which compares outputs and identifies the preferred one (Zheng et al., 2023). Judge2Vec uses multi-criterion ratings to characterize criterion-specific judge strengths for complementary jury construction. The same ratings can also induce criterion-wise preferences between responses to the same input without additional judge calls, allowing a selected jury to support both scoring and comparison while preserving criterion-level information.

Crowdsourcing and annotator reliability. The LLM-jury setting is also related to classical work on learning from noisy human annotators. Dawid–Skene jointly estimates latent labels and annotator-specific error rates using maximum likelihood (Dawid & Skene, 1979). GLAD jointly models annotator expertise and item difficulty (Whitehill et al., 2009), while MACE learns annotator competence and latent labels without requiring gold annotations (Hovy et al., 2013). Subsequent work provides stronger statistical foundations and estimation procedures for truth inference from multiple noisy annotators (Zhang et al., 2016).

These approaches address the important question of how much annotators should be trusted when their judgments are combined. However, their central objective is truth inference or reliability-weighted aggregation from a set of collected annotations, rather than constructing a small, multi-dimensional subset for future deployment. As a result, they generally still require judgments from the full annotator or judge pool on each evaluation item before aggregation, which can be costly when evaluation is performed at scale. Judge2Vec instead constructs a compact, reliable jury from an unlabeled pilot and subsequently queries only the selected judges on new items, substantially reducing deployment cost.

Multi-criterion evaluation. Several evaluation frameworks emphasize that output quality is inherently multi-dimensional (Zhang et al., 2026a). For instance, G-Eval evaluates generated outputs under explicit task-specific criteria (Liu et al., 2023). FLASK decomposes coarse overall evaluation into fine-grained alignment skills and demonstrates the value of skill-level assessment (Ye et al., 2024). Prometheus 2 supports customized evaluation rubrics for both direct scoring and pairwise comparison (Kim et al., 2024), while MAPLE integrates judgments across multiple criteria and multiple evaluators (Jinji et al., 2026).

Building on this perspective, Judge2Vec treats not only output quality but also judge reliability as multi-dimensional. Rather than collapsing each evaluator to a single overall reliability score, we represent it by criterion-specific competence. Evaluation criteria therefore serve not only as dimensions for scoring responses, but also as coordinates of the judge representation itself. As a result, two judges with similar overall performance can still be distinguished by different strengths across criteria. This representation enables complementary jury construction based on heterogeneous criterion-specific capabilities.

Expert team formation and complementary selection. A broader analogue of our selection problem appears in expert team formation. Classical formulations assume a pool of individuals with heterogeneous skills and seek a team whose combined skills satisfy the requirements of a task. Lappas et al. additionally model compatibility through a social network and optimize communication cost (Lappas et al., 2009). Anagnostopoulos et al. study online team formation, jointly considering required-skill coverage, communication overhead, and workload balance (Anagnostopoulos et al., 2012). Kargar et al. further consider both communication and personnel costs (Kargar et al., 2013). USTF provides a unified platform for comparing representative team-formation algorithms (Wang et al., 2016), and Saeedi et al. organize the broader literature by problem formulation, optimization objectives, and search- versus learning-based approaches (Saeedi et al., 2025).

A central distinction lies in what is known before selection. Expert-team-formation methods generally assume that candidates' relevant skills, and often their collaboration relationships, are already available as input. In LLM jury construction, however, criterion-specific judge competence is initially unknown. Judge2Vec therefore first infers a competence embedding from unlabeled judging behavior and then performs complementary selection in this learned representation space. The setting also differs operationally: expert teams are formed to collaborate on a task, whereas LLM judges independently evaluate the same items. Accordingly, Judge2Vec measures redundancy through similarity between competence profiles rather than through communication or collaboration costs.

C Shared Criterion Space

For zero-shot cross-task jury transfer (Section 4.2), inspired by the task-agnostic evaluation dimensions in FLASK (Ye et al., 2024), we use a fixed shared criterion space with $D_U = 10$. Its dimensions are *coherence*, *correctness*, *relevance*, *engagement*, *fluency*, *complexity*, *specificity*, *completeness*, *overall*, and *empathy*.

These dimensions provide common coordinates for aligning and transferring judge representations across tasks.

D Evaluation Criteria

Qualitative datasets. Each qualitative dataset provides multiple built-in evaluation criteria, each with human annotations as the reference. For embedding construction, we use only judge ratings on these criteria; for held-out evaluation, we compare the selected jury's predictions against the human annotations across all criteria. Specifically, HANNA uses six criteria (D=6): *relevance*, *coherence*, *empathy*, *surprise*, *engagement*, and *complexity*; SummEval uses four (D=4): *coherence*, *consistency*, *fluency*, and *relevance*; HelpSteer2 uses five (D=5): *helpfulness*, *correctness*, *coherence*, *complexity*, and *verbosity*; and FED uses eight (D=8): *interesting*, *engaging*, *specific*, *relevant*, *correct*, *semantically appropriate*, *fluent*, and *overall*.

Quantitative datasets. Each quantitative dataset instead provides a single target criterion for held-out evaluation. To construct multi-dimensional competence embeddings, we supplement this target with task-relevant auxiliary criteria drawn from prior work. These auxiliary criteria are used only for embedding construction and jury selection; final evaluation remains exclusively on the dataset's original target criterion against its ground-truth reference. We preserve the criterion definitions and scoring rubrics from their respective sources as follows:

- **MATH and MathVista (D=4).** The target criterion is binary *correctness*. We add three auxiliary *Logical Thinking* criteria from FLASK (Ye et al., 2024): *logical_correctness*, *logical_robustness*, and *logical_efficiency*.
- **ExpertQA (D=11).** The target criterion is *attribution*. We add ten auxiliary criteria: *answer_relevance* and *faithfulness* (Es et al., 2024); *knowledge_coverage* (Tan et al., 2024); *coherence* and *conciseness* (Bai et al., 2025); *completeness* (Min et al., 2023); *helpfulness* and *correctness* (Wang et al., 2024c); *instruction_following* (Zhou et al., 2023); and *reasoning_soundness* (Ye et al., 2024).
- **HAGRID (D=4).** The target criterion is *attribution*. We add three auxiliary criteria: *answer_relevance* (Es et al., 2024), *knowledge_coverage* (Tan et al., 2024), and *coherence* (Bai et al., 2025).
- **TheAgentCompany (D=4).** The target criterion is *task_completion*. We add three auxiliary criteria: *tool_appropriateness* and *parameter_accuracy* from MCP-Bench (Wang et al., 2026), together with *plan_optimality* from TRAIL (Deshpande et al., 2025).

E Test-Set Sizes

Table 5 reports the number of held-out evaluation items used for each dataset.

Table 5: **Held-out test-set sizes.**

Dataset	Test size
HANNA	656
SummEval	1,200
HelpSteer2	1,038
FED	175
MATH	225
ExpertQA	754
HAGRID	359
TheAgentCompany	861
MathVista	465

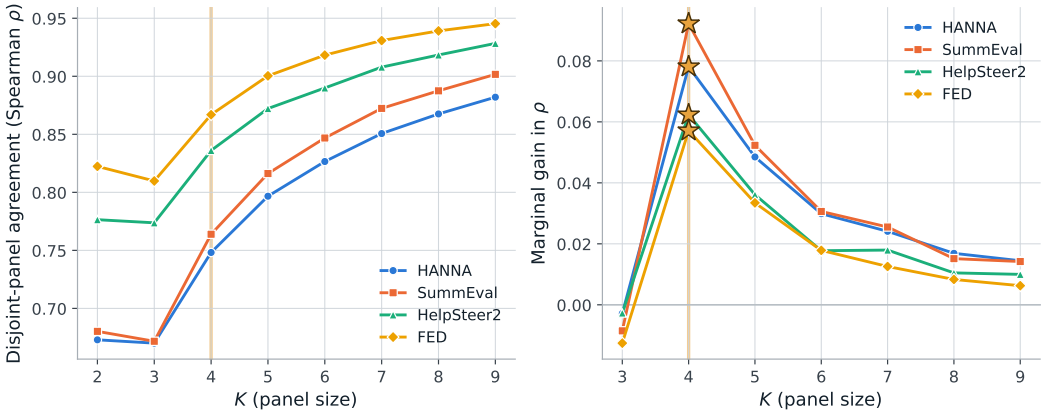


Figure 4: **Unsupervised jury-size selection.** Left: disjoint-panel agreement as a function of jury size K, measured by Spearman correlation between the aggregated predictions of two randomly sampled disjoint K-judge panels on the unlabeled pilot set. Right: the corresponding marginal increase in agreement. The largest marginal increase occurs at $K = 4$ for all datasets, motivating our use of $K = 4$.

F Additional Implementation Details

Multimodal judge pool. The multimodal study uses a separate pool of seven judges from four model families: Claude (Opus 4.8, Fable 5), Llama (4 Maverick, 4 Scout), Mistral (Ministral 14B, Ministral 8B), and Kimi (K2.5). All seven accept interleaved image and text inputs, receive the same scoring prompt and evaluation criteria, and are queried at temperature zero.

Jury size. We choose K in an unsupervised manner using only the unlabeled pilot set. The intuition is simple: if a jury size is sufficiently reliable, two independently sampled K-judge panels should produce similar aggregated predictions on the same pilot examples. Accordingly, for each candidate K, we repeatedly sample two disjoint K-judge panels, aggregate each panel's predictions using the same trim-one rule used at deployment, and measure the agreement between the two aggregated outputs. We then examine the marginal increase in this *disjoint-panel agreement* as K grows. As shown in Figure 4, the largest marginal increase occurs at $K = 4$ in all five splits of every dataset. We therefore use $K = 4$ as the common jury size, allowing K to be selected without using any labeled evaluation data.

Pilot size. We determine the pilot size from the stability of the learned judge representation. We consider

$$p \in \{15, 25, 40, 60, 80, 100, 125, 150, 200, 250, 300, 400\},$$

truncated at 400. For each candidate p, we split the p pilot examples into two halves, independently reconstruct the Judge2Vec competence representation from each half, and measure their agreement. We apply the Spearman–Brown correction

$$\rho_{\mathrm{SB}} = \frac{2\rho}{1+\rho}$$

to the split-half correlation and select the smallest p for which $\rho_{\mathrm{SB}} \geq 0.95$; if the threshold is not reached, we use 400.