

MetaGAI: A Large-Scale and High-Quality Benchmark for Generative AI Model and Data Card Generation

Haoxuan Zhang¹ Ruochi Li² Yang Zhang^{1,*} Zhenni Liang¹

Junhua Ding¹ Ting Xiao¹ Haihua Chen^{1,*}

¹University of North Texas, Denton, TX, USA

²North Carolina State University, Raleigh, NC, USA

{haoxuanzhang, zhenniliang}@my.unt.edu

{yang.zhang, junhua.ding, ting.xiao, haihua.chen}@unt.edu

rli14@ncsu.edu

*Corresponding authors

Abstract

The rapid proliferation of Generative AI has created a pressing need for rigorous documentation standards that support transparency and governance. Manual creation of Model and Data Cards, however, is difficult to scale, and automated approaches lack the large-scale, high-fidelity benchmarks required for systematic evaluation. We introduce MetaGAI, a benchmark of 2,541 semantically matched document triplets constructed from academic papers, GitHub repositories, and Hugging Face artifacts. Reference cards are produced by a multi-agent pipeline of specialized Retriever, Generator, and Editor agents, and the pipeline is assessed through a four-part validation protocol. We further propose an evaluation framework that combines reference-based automated metrics with a multi-model LLM-as-a-Judge protocol, together with explicit agreement and sensitivity analyses. Among the evaluated baselines, Qwen3-30B-A3B-Instruct achieves the best observed cost-quality trade-off, and the results reveal a recurring tension between faithfulness and completeness. MetaGAI provides a testbed for benchmarking, training, and analyzing automated Model and Data Card generation at scale. Our data and code are available at <https://github.com/haoxuan-unt2024/MetaGAI-Benchmark>.

1 Introduction

The rapid proliferation of Generative AI (GenAI) has transformed machine learning deployment: community platforms such as Hugging Face now host more than two million models and datasets (Wolf et al., 2020; Horwitz et al., 2025). As these systems move from research artifacts (Zhang et al., 2023b, 2025a) to critical infrastructure, documentation standards have evolved from foundational Model Cards (Mitchell et al., 2019) and Data Cards (Pushkarna et al., 2022) to comprehensive System Cards (OpenAI et al., 2024;

Comanici et al., 2025; OpenAI et al., 2025). Such documentation supports longitudinal model tracking, compliance auditing, and risk assessment in high-stakes domains (Longpre et al., 2024; Castaño et al., 2024). Without high-quality documentation, the AI ecosystem lacks the interoperability needed to benchmark capabilities and trace data provenance across complex supply chains (Rahman et al., 2025; Zhang et al., 2025c).

A significant bottleneck nevertheless impedes the widespread adoption of such documentation. The GenAI ecosystem is increasingly driven by the “long tail” of open-source contributions on community platforms (Wolf et al., 2020; Horwitz et al., 2025), and unlike well-resourced industry laboratories, many developers in this ecosystem lack the capacity to maintain rigorous documentation. Manual card creation faces severe scalability constraints and often yields incomplete, inconsistent, or subjective documentation (Yang et al., 2024; Liang et al., 2024). As a result, many papers and repositories still lack the structured metadata needed for reproducibility (Olmo et al., 2025). Automated documentation generation has therefore emerged as an important direction (Liu et al., 2024a; Zhang et al., 2026), but current approaches face substantial challenges: zero-shot methods may hallucinate details when summarizing lengthy documents, whereas retrieval-augmented strategies must align heterogeneous paper structures with rigid schemas (Zhang et al., 2026). Progress is further constrained by the absence of large-scale benchmarks with well-supported reference cards for systematic evaluation.

To address this gap, we introduce **MetaGAI**, a large-scale benchmark designed to evaluate automated Model and Data Card generation systematically. Unlike prior datasets that treat documentation as a simple summarization task, MetaGAI formulates automated documentation as structured generation evaluated against multi-source refer-

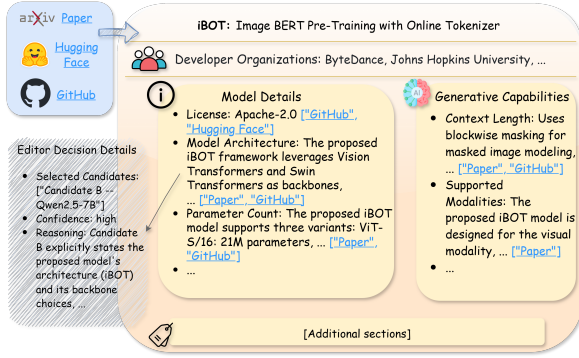


Figure 1: **MetaGAI Benchmark Construction Example.** Automated GenAI card generation for the iBOT model (Zhou et al., 2022). Multi-source triangulation combines architectural concepts from the paper, hyperparameters from GitHub, and licensing information from Hugging Face; editor-based synthesis then produces an editor-refined reference card.

ences, reflecting the practical need to verify scientific claims against implementation details.

As illustrated in Figure 1, complete documentation cards require evidence from heterogeneous sources. For the iBOT model (Zhou et al., 2022), for example, architectural concepts come from the academic paper, implementation details from the GitHub repository, and deployment constraints from Hugging Face. Our construction pipeline uses specialized Retriever, Generator, and Editor agents to synthesize these signals into editor-refined reference cards. We assess the pipeline through four complementary components: (1) retrieval-strategy selection on a sampled, human-annotated chunk-field validation set; (2) automated generator-divergence analysis; (3) source-grounded human and LLM evaluation of a stratified sample of editor-refined cards; and (4) editor-architecture comparison.

The main contributions of this study are as follows:

- We construct MetaGAI, a large-scale benchmark for GenAI documentation comprising 2,541 semantically matched triplets, and we assess its construction pipeline through sampled human annotation, automated diagnostics, and source-grounded human and LLM evaluation of a stratified card sample.
- We propose a multi-metric evaluation framework that combines reference-based automated metrics with a multi-model LLM-as-a-Judge protocol and explicit agreement and

sensitivity analyses.

- We provide an extensive empirical analysis showing that Qwen3-30B-A3B-Instruct offers the best observed cost-quality trade-off among the evaluated baselines, and that a recurring tension persists between faithfulness and completeness in generated metadata.

2 Related Work

Model Documentation. The lack of standardized documentation for trained machine learning models limits transparency, reproducibility, and responsible use, including in natural language processing (NLP). Model Cards were introduced to provide structured summaries of trained models, covering intended use, evaluation settings, performance, and limitations (Mitchell et al., 2019). Dataset-level documentation developed in parallel through Data Cards, which emphasize data provenance, collection processes, representativeness, and ethical considerations (Pushkarna et al., 2022). Together, Model Cards and Data Cards provide complementary documentation of model-level and data-level sources of uncertainty. Subsequent work extended model documentation to interactive and machine-readable formats: interactive Model Cards were shown to improve user understanding of model behavior (Crisan et al., 2022), and linked, semantic representations were proposed to improve traceability and reuse (Donald et al., 2023). Large-scale empirical analyses nevertheless report substantial variation in documentation quality, with evaluation and limitation sections frequently missing or incomplete (Liang et al., 2024).

Automatic Model and Data Card Generation.

To reduce the manual effort that documentation requires, recent work has explored the automated generation of Model and Data Cards. Liu et al. (2024a) introduced CARDBENCH, a benchmark of human-written cards, together with CARDGEN, an LLM-based system that generates structured documentation by retrieving information from model repositories and associated papers. Their results showed that automated generation can improve documentation completeness. Existing benchmarks nevertheless remain limited in scale and coverage (Zhang et al., 2026), which constrains the systematic evaluation of generation methods. MetaGAI addresses this limitation by providing a large-scale benchmark for GenAI Model and Data Card generation.

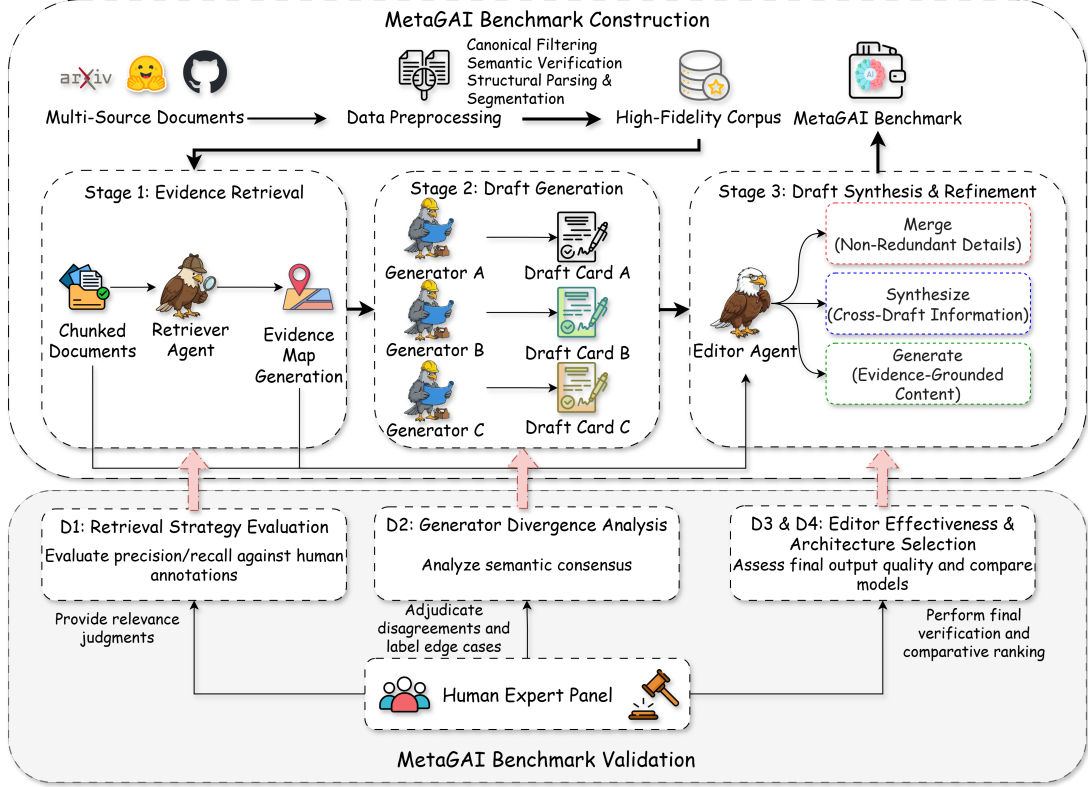


Figure 2: **MetaGAI Benchmark Construction and Validation Framework.** The pipeline integrates multi-source document preprocessing, a three-stage multi-agent framework (Evidence Retrieval, Draft Generation, and Draft Synthesis and Refinement), and a four-part validation protocol (D1–D4) covering retrieval-strategy validation, generator-divergence analysis, editor-effectiveness evaluation, and editor-architecture selection.

3 MetaGAI Benchmark Construction

Figure 2 illustrates the MetaGAI benchmark construction and validation pipeline, which comprises three sequential stages: (1) evidence retrieval from multi-source documents, (2) multi-generator draft synthesis, and (3) editor-based consolidation and refinement. Compared with traditional single-model approaches (Liu et al., 2024a), this architecture adds ensemble generation and a sample-based quality-assessment protocol for the resulting reference cards.

3.1 Task Definition

We formulate **MetaGAI** as a structured generation task that maps unstructured scientific papers $\mathcal{P}$ to standardized cards $\mathcal{C}$. The benchmark incorporates a deliberate asymmetry between reference construction and evaluation input. The reference card $\mathcal{C}_{GT}$ is derived from a document triplet $(\mathcal{P}, \mathcal{G}, \mathcal{H})$, where $\mathcal{G}$ denotes the GitHub repository and $\mathcal{H}$ the Hugging Face artifact, so that it captures implementation details frequently omitted from manuscripts (Mitchell et al., 2019; Pushkarna et al.,

2022). The evaluated generator, by contrast, receives only the paper input $\mathcal{P}$. MetaGAI therefore measures how much multi-source documentation quality can be recovered from paper-only evidence.

We adopt and extend the taxonomies of Mitchell et al. (2019) and Pushkarna et al. (2022) with GenAI-specific fields (Appendix A, Table 4). Each card $\mathcal{C}$ comprises M field-value pairs $\{(k_i, v_i)\}_{i=1}^M$, where k_i is a schema attribute and v_i its content. The objective is to learn a mapping f_θ that predicts $\hat{\mathcal{C}}$ from the paper alone:

$$\hat{\mathcal{C}} = f_\theta(\mathcal{P}) \quad (1)$$

while minimizing divergence from the multi-source reference $\mathcal{C}_{GT}(\mathcal{P}, \mathcal{G}, \mathcal{H})$, thereby quantifying how much documentation quality can be recovered under paper-only constraints.

3.2 Data Acquisition and Filtering

Corpus Construction. We built the benchmark by systematically triangulating arXiv, GitHub, and Hugging Face metadata in order to capture implementation details absent from published papers.

Unlike previous work that aggregates existing documentation (Liu et al., 2024a), we constructed a new corpus from scratch and applied automated cross-source semantic matching before card generation. We first collected computer science papers published on arXiv between 2019 and 2025 and retrieved their corresponding GitHub repositories and Hugging Face model or dataset artifacts, which produced an initial pool of **15,727** candidates. We then removed candidates with empty GitHub repositories or Hugging Face artifacts, together with cases in which a many-to-many paper–repository–artifact relation did not yield a unique linkage. This deterministic filtering left **4,068** uniquely linked candidates for semantic verification.

Semantic Verification. To eliminate spurious citations and improve cross-source alignment, we used Qwen3-30B-A3B-Instruct (Yang et al., 2025) for automated semantic consistency verification based on the paper introduction and the GitHub and Hugging Face README files (Prompt F.1). We assessed the consistency of the correspondence criterion on a pilot sample of 100 triplets (50 model and 50 dataset). Two students with AI/ML backgrounds independently annotated whether the paper, GitHub repository, and Hugging Face artifact referred to the same entity, and they reached perfect agreement on this sampled binary task. The pilot therefore provides evidence that the correspondence criterion can be applied consistently within the sampled triplets. After automated semantic filtering, the final corpus contains **2,541** semantically matched triplets (Appendix B). The provenance analysis in Table 10 shows that 22.2–22.4% of Model Card fields and 26.7–30.7% of Data Card fields are labeled Multi-Source, and that a further 3.4–3.5% and 3.3–3.9%, respectively, are labeled GitHub Only. A detailed provenance analysis appears in Appendix G.1.

3.3 Benchmark Generation Pipeline

We developed an automated pipeline that synthesizes reference cards through document preprocessing and a multi-agent framework.

3.3.1 Pre-processing

Raw PDF documents are converted to structured Markdown using OLMoCR-2, a state-of-the-art OCR model optimized for academic layouts (Poznanski et al., 2025). To accommodate context-window constraints, the converted papers and README files are segmented into 1024-token

chunks with 10% overlap between adjacent chunks, which preserves semantic continuity. Chunk boundaries are extended to the nearest sentence boundary so that semantically coherent passages are less likely to be split.

3.3.2 Multi-Agent Framework

We employ a multi-agent architecture (Algorithm 1) to maximize factual grounding and minimize hallucinations (Du et al., 2024). Whereas traditional approaches rely on single-model generation following retrieval (Liu et al., 2024a), our framework adds ensemble diversity and editor-based cross-validation. The framework processes the full context $\mathcal{P} \cup \mathcal{S}$, where $\mathcal{S} = \{\mathcal{G}, \mathcal{H}\}$ denotes supplementary documentation, through three specialized agents:

$$v_i = \text{Retriever}(\mathcal{P} \cup \mathcal{S}, k_i) \quad (2)$$

$$\tilde{\mathcal{C}}^{(j)} = \text{Generator}_j(\{(k_i, v_i)\}_{i=1}^M), \quad j \in \{1, 2, 3\} \quad (3)$$

$$\hat{\mathcal{C}} = \text{Editor}(\mathcal{P} \cup \mathcal{S}, \tilde{\mathcal{C}}^{(1:3)}) \quad (4)$$

Retriever Agent. The Retriever Agent aligns unstructured text with schema fields through exhaustive chunk-level classification (Zhang et al., 2023a). Each document segment receives a relevance score (0–4) together with matched sub-fields drawn from a predefined ontology (Prompt F.2). The construction pipeline is recall-oriented: it retains every chunk assigned to a valid schema field with relevance ≥ 3 rather than selecting a fixed top- k set. On the sampled validation set (Section 4.1), generative models outperformed the two discriminative rerankers, which motivated our selection of Qwen3-30B-A3B-Instruct as the retrieval backbone. Generative reasoning allows the retriever to interpret implicit metadata requirements that embedding-based similarity scores do not capture.

Generator Agent. To mitigate architectural bias and capture diverse interpretations, we use ensemble generation (Wang et al., 2023) with three architecturally distinct LLMs (OLMo-3-7B, Llama-3.1-8B, and Qwen2.5-7B). Each model independently synthesizes draft content, evidence quotations, and confidence scores. A strict “Subject Focus” constraint (Prompt F.3) ensures that the agents synthesize information only about the proposed model or dataset and exclude baselines and prior work. This multi-model design reduces dependence on any

single architecture when evidence about complex GenAI systems is captured.

Editor Agent. A “Chief Editor” consolidates the candidate drafts using a larger, cross-family model, which reduces self-enhancement bias (Zheng et al., 2023). The editor checks semantic alignment between the drafts and the raw evidence, filters attribution errors, and merges non-redundant details into concise final entries (Prompt F.4). This explicit verification stage is designed to mitigate hallucinations in technical specifications; concrete examples appear in Appendix H. We ran the Editor stage separately with Ministral-3-14B-Instruct and GPT-OSS-20B, which yielded the Bench_Mistral and Bench_GPT-OSS reference sets evaluated in Section 4.3.

4 MetaGAI Validation and Analysis

We assessed the benchmark construction pipeline through four complementary components. D1 compared retrieval strategies on a sampled human-annotated validation set; D2 quantified semantic variation between ensemble generators through automated analysis; D3 evaluated editor effectiveness on a stratified sample of reference cards using two human evaluators and three LLM judges; and D4 compared two editor architectures. Together, these components provide sample-based evidence about pipeline behavior and reference quality rather than an exhaustive human annotation of the benchmark.

4.1 Retrieval Strategy Validation (D1)

We compared five retrieval models from two paradigms on a sampled human-annotated validation set of 276 positive chunk-field pairs that both annotators agreed were relevant (Appendix D.3). As Table 1 shows, the discriminative rerankers achieve $P@1 < 0.14$, which indicates that semantic similarity alone is insufficient for schema mapping. Qwen3-30B-A3B-Instruct achieves the best performance ($P@1$: 0.635, $F1@5$: 0.342), supporting its selection as the retrieval backbone. In the construction pipeline, all chunks with relevance ≥ 3 are retained for the matched field rather than only the top five.

To separate evidence-retention failures from downstream omissions, we further conducted a construction-stage propagation diagnostic (Appendix D.4; Table 6). Among 3,216 unpopulated items, 92.2% lack sufficient evidence in the retained context, whereas downstream population

Model	P@1	R@5	F1@5
<i>Discriminative Rerankers</i>			
BGE-Reranker-v2-m3	0.135	0.114	0.101
Qwen3-Reranker-8B	0.115	0.137	0.091
<i>Generative LLMs</i>			
Llama-3.1-8B-Instruct	0.615	0.406	0.326
Qwen2.5-7B-Instruct	0.538	0.282	0.232
Qwen3-30B-A3B-Instruct	0.635	0.469	0.342

Table 1: Comparative validation of retrieval models (D1). The best result in each column is shown in bold.

succeeds in 95.9% of cases in which sufficient evidence is retained. Benchmark incompleteness is therefore associated primarily with evidence availability before final generation rather than with a failure to use evidence that has already been retained.

4.2 Generator Divergence Analysis (D2)

We analyzed semantic consensus within the generator ensemble (OLMo-3-7B, Llama-3.1-8B, and Qwen2.5-7B) by computing pairwise BERTScore (Zhang et al., 2020) similarity across all 2,541 samples. High consensus ($\mu > 0.8$) in fields such as *ethical considerations* ($\mu = 0.848$) predominantly reflects evidence scarcity rather than genuine agreement (Appendix G.4). Fields with pronounced divergence ($\sigma > 0.4$), by contrast, exhibit heterogeneous generator outputs. This variation motivates editor-based consolidation, but it does not by itself establish that the differing outputs are complementary. The full mean-standard-deviation analysis appears in Appendix G.3.

4.3 Sample-Based Reference Quality Assessment

A benchmark is only as trustworthy as its references. To obtain sample-based evidence about their quality, we conducted a controlled evaluation that compared the two final editor-refined reference sets with an unedited generation baseline. For each entity in a stratified random sample of 100 entities (50 models and 50 datasets), we compared three outputs, which yielded 300 card outputs in total: (1) **Raw Baseline**, the unedited generator draft with the highest token count; (2) **Bench_Mistral**, the output of the Ministral-3-14B-Instruct editor; and (3) **Bench_GPT-OSS**, the output of the GPT-OSS-20B editor. Two human evaluators and three LLM judges scored the same sampled outputs. The complete human-evaluation protocol, judge con-

System	Ann-1	Ann-2	Llama	GPT-OSS	Qwen
Bench_GPT-OSS	4.475	3.669	4.101	3.019	4.303
Bench_Mistral	4.355	3.799	4.634	2.897	3.990
Raw Baseline	3.709	2.966	3.130	2.951	3.447

Table 2: Mean Likert scores (1–5) for reference-card quality from two human evaluators and three LLM judges on 300 card outputs drawn from a stratified sample of 100 entities. **Bold** denotes the best score per column among the editor-refined systems.

figuration, and agreement analysis are reported in Appendix D.5.

Effectiveness of Editor Refinement (D3). Table 2 reports mean Likert scores (1–5) from two human evaluators and three LLM judges. Averaged across the two human evaluators, Bench_GPT-OSS and Bench_Mistral score 4.072 and 4.077, respectively, compared with 3.338 for the Raw Baseline, an improvement of approximately 0.74 points. Four of the five individual evaluators score both editor-refined references above the Raw Baseline; GPT-OSS-120B scores Bench_GPT-OSS slightly above the Raw Baseline and Bench_Mistral slightly below it. These results provide sample-based evidence that editor refinement improves reference-card quality, and they also motivate the judge-sensitivity analysis in Appendix E.

Editor Architecture Selection (D4). Bench_GPT-OSS and Bench_Mistral obtain nearly identical average human scores (4.072 vs. 4.077), but their relative ordering varies across individual evaluators: Bench_GPT-OSS leads under Ann-1 (4.48 vs. 4.36), whereas Bench_Mistral leads under Llama-3.3-70B-Instruct (4.63 vs. 4.10). We therefore retained them as two independent reference sets rather than claiming equivalence or averaging their card text. For every reference-based automated metric, each system output was scored separately against Bench_GPT-OSS and Bench_Mistral, and the two resulting scores were averaged only at reporting time. This procedure reduces dependence on a single editor architecture while preserving reproducible references.

5 Experiments

5.1 Experimental Setup

We evaluated all models zero-shot, generating cards $\hat{\mathcal{C}}$ exclusively from paper text $\mathcal{P}$, consistent with the task definition in Section 3.1. This setting reflects realistic deployment constraints, because

many published artifacts lack repositories or Hugging Face pages entirely (Liang et al., 2024; Yang et al., 2024). To balance statistical rigor against computational feasibility, we computed automated metrics on the **full benchmark** ($N = 2,541$) and conducted LLM-as-a-Judge evaluation on a **stratified random sample** of 500 entries (250 Model Card and 250 Data Card entries).

5.2 Baselines

We examined models grouped by access mode and architectural paradigm. Preliminary experiments with smaller models (≤ 14 B) revealed consistent generation failures; the reported evaluation therefore focuses on models with at least 20B parameters.

Open-Weight Models. We evaluated two architectural classes. **Dense Models** use standard transformer architectures (20B–32B parameters) and include Mistral-Small-24B, Gemma-3-27B (Team et al., 2025), and Qwen3-32B (Yang et al., 2025). **MoE Models** use Mixture-of-Experts architectures with high total parameter counts but efficient active-parameter usage, and include GPT-OSS-20B (OpenAI et al., 2025), Nemotron-Nano-30B-A3B (NVIDIA et al., 2025), and Qwen3-30B-A3B-Instruct (Yang et al., 2025). **Closed-Source Models.** We also included proprietary models accessed via API: the GPT-5 series (Mini and Nano) and the Gemini-2.5 series (Flash and Flash-Lite) (Cormanici et al., 2025).

5.3 Evaluation Metrics

Our evaluation framework has two layers, combining quantitative structural metrics with source-grounded human and LLM judgments.

Automated Metrics. We measured field coverage and semantic alignment using three metrics:

- **Completeness:** quantifies field-level coverage relative to the reference card. Let $\mathcal{K}(\mathcal{C})$ denote the set of populated keys in card $\mathcal{C}$:

$$\text{Completeness} = \frac{|\mathcal{K}(\hat{\mathcal{C}}) \cap \mathcal{K}(\mathcal{C}_{GT})|}{|\mathcal{K}(\mathcal{C}_{GT})|} \quad (5)$$

- **Semantic Similarity:** we used **ROUGE-L** (Lin, 2004) for lexical overlap and **BERTScore (F1)** (Zhang et al., 2020) for semantic alignment.

For each generated card, we computed Completeness, ROUGE-L, and BERTScore separately

against the corresponding Bench_GPT-OSS and Bench_Mistral reference cards and averaged the two scores only after evaluation; no textual averaging of the two reference cards was performed.

LLM-as-a-Judge Evaluation. To capture nuances beyond n-gram matching, we implemented an ensemble framework using GPT-OSS-120B (OpenAI et al., 2025), Llama-3.3-70B-Instruct (Grattafiori et al., 2024), and Qwen3-235B-A22B-2507 (Yang et al., 2025). Following the protocol in Appendix D (Prompt F.5), the judges evaluated the generated fields directly against the triangulated source context on a 1–5 Likert scale. We averaged their scores to reduce dependence on any single judge.

Standardized Inference Cost. To assess economic feasibility at scale, we define a Cost Index that compares inference prices under a fixed token workload. If p_{in} and p_{out} denote the USD prices per one million input and output tokens, respectively, then $\text{Cost Index} = p_{\text{in}} + 0.2p_{\text{out}}$. The index therefore corresponds to a fixed workload of one million input tokens and 0.2 million output tokens across models. Pricing is derived from OpenRouter¹. This index is a standardized price proxy rather than a measure of per-card spending or computational energy use.

5.4 Experimental Results

We computed automated metrics on all 2,541 entries and conducted LLM-as-a-Judge evaluation on 500 sampled entries. Beyond overall quality, we analyzed standardized cost–quality trade-offs and the relationship between reference-based similarity metrics and judged quality. Table 3 presents the results for all baseline models. The analysis below is organized around architectural efficiency, evaluation-metric validity, and systematic generation limitations.

5.4.1 Architectural Efficiency and Economic Viability

MoE Performance Among Evaluated Baselines. Within the Qwen family, the MoE-based Qwen3-30B-A3B-Instruct (4.55 Model Card quality) outperforms the dense Qwen3-32B (3.60) by 0.95 points despite comparable total parameter counts. This result is consistent with a benefit from sparse

activation at this scale, although differences in training and model design may also contribute. GPT-OSS-20B achieves substantially lower quality (3.06 and 3.38), which shows that an MoE architecture alone does not guarantee strong performance. The evaluated models are insufficient to establish a universal minimum parameter threshold for MoE benefits.

Open-Weight Cost Dominance. Qwen3-30B-A3B-Instruct occupies a Pareto-optimal position on the cost–quality frontier: it achieves the best observed Model Card quality (4.55) at a relatively low standardized cost (Cost Index: 0.15). GPT-5-Mini attains 4.28 quality at a Cost Index of 0.65, approximately 4.3 times the standardized cost of Qwen3-30B-A3B-Instruct, and Gemini-2.5-Flash combines lower quality (4.00) with the highest Cost Index (0.80). The detailed analysis in Appendix G.6 confirms the favorable standardized price–quality position of Qwen3-30B-A3B-Instruct among the evaluated baselines.

5.4.2 Evaluation Metric Validity

Limitations of Reference-Based Similarity Metrics. Reference-based similarity scores do not consistently track judged quality. Mistral-Small-24B, for example, obtains the highest Data Card ROUGE-L (0.270) but a relatively low judged quality score (3.26), whereas Qwen3-30B-A3B-Instruct obtains a lower ROUGE-L (0.246) but a higher judged quality score (4.30). BERTScore values also occupy a narrow 0.10–0.20 range even though judged quality ranges from 3.06 to 4.55. These results show that ROUGE-L and BERTScore alone are insufficient for ranking generated card quality.

Completeness and Judged Quality Can Diverge. Structural coverage and judged quality capture different aspects of performance. On Model Cards, Gemma-3-27B achieves high Completeness (0.734, second-best) yet scores only 3.94 in judged quality, below Qwen3-30B-A3B-Instruct (4.55), which also achieves higher Completeness (0.786). Similarly, on Data Cards, Nemotron-Nano-30B-A3B achieves 0.557 Completeness yet scores only 3.37 in quality, below Qwen3-30B-A3B-Instruct (4.30), which reaches 0.702 Completeness. These examples show that high field coverage does not guarantee high judged quality. Figure 3 illustrates the corresponding field-level patterns: averaged across all baselines, models exceed 0.7 Completeness on explicit, signal-rich fields (*model details*,

¹<https://openrouter.ai/> Pricing was retrieved from OpenRouter in February 2026.

Model	Faithfulness		Relevance		Accuracy		Consistency		Usefulness		Qual. Avg		Comp.		BScore		RL		Cost
	D	M	D	M	D	M	D	M	D	M	D	M	D	M	D	M	D	M	
Dense Models																			
Mistral-Small-24B	3.39	3.75	3.28	3.59	3.33	3.70	3.64	3.97	2.67	3.14	3.26	3.63	.280	.327	.193	<u>.194</u>	.270	<u>.255</u>	0.10
Gemma-3-27B	3.81	3.97	3.76	3.94	3.76	3.96	4.03	4.23	3.26	3.58	3.72	3.94	.464	<u>.734</u>	.151	.151	.214	.203	<u>0.07</u>
Qwen3-32B	3.40	3.72	3.29	3.58	3.32	3.69	3.66	3.94	2.65	3.10	3.26	3.60	.386	.513	<u>.184</u>	.198	<u>.267</u>	.265	0.13
MoE Models																			
GPT-OSS-20B	3.18	3.49	3.09	3.36	3.11	3.45	3.45	3.74	2.49	2.87	3.06	3.38	.394	.492	.146	.133	.235	.220	0.06
Nemotron-Nano-30B-A3B	3.45	3.80	3.41	3.76	3.39	3.77	3.71	4.03	2.88	3.35	3.37	3.74	<u>.557</u>	.644	.124	.133	.199	.206	0.11
Qwen3-30B-A3B-Instruct	4.33	4.50	4.36	4.57	4.31	4.51	4.46	4.67	4.06	4.49	4.30	4.55	.702	.786	.169	.174	.246	.243	0.15
Closed-Source Models																			
GPT-5-Mini	<u>4.14</u>	<u>4.32</u>	<u>4.09</u>	<u>4.26</u>	<u>4.10</u>	<u>4.32</u>	<u>4.25</u>	<u>4.43</u>	<u>3.74</u>	<u>4.07</u>	<u>4.06</u>	<u>4.28</u>	.556	.216	.102	.117	.185	.207	0.65
GPT-5-Nano	3.18	3.38	3.06	3.18	3.10	3.32	3.46	3.57	2.53	2.73	3.06	3.23	.383	.127	.113	.149	.188	.228	0.13
Gemini-2.5-Flash	3.88	4.08	3.80	3.97	3.82	4.06	4.04	4.19	3.27	3.69	3.76	4.00	.443	.181	.160	.170	.241	.246	0.80
Gemini-2.5-Flash-Lite	3.93	4.10	3.88	4.03	3.87	4.08	4.12	4.25	3.39	3.69	3.83	4.03	.511	.194	.140	.144	.207	.208	0.18

Table 3: **Comprehensive MetaGAI Benchmark Results.** Performance on Data Card (D) and Model Card (M) generation. Completeness, BERTScore, and ROUGE-L are reported on the full test set ($N = 2,541$) and averaged over separate evaluations against the two editor-derived reference sets. Qual. Avg (Qualitative Average) is based on source-grounded evaluation of a sample of 500 entries. Cost Index is a standardized price in USD for one million input tokens and 0.2 million output tokens. **Bold** and underlined values indicate the best and second-best performance, respectively.

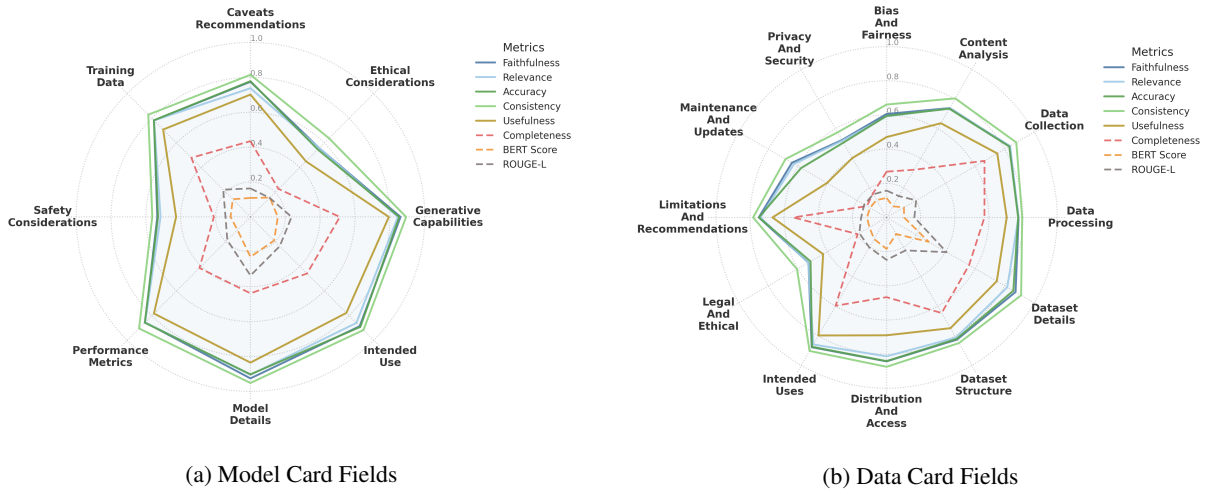


Figure 3: **Field-Level Performance Patterns Averaged Across All Baselines.** Colored lines show evaluation metrics across the card fields represented by the axes. Performance is stronger on signal-rich fields (*model details*) but degrades on abstract categories (*ethical considerations*), which reveals systematic generation difficulty when documentation is sparse.

performance metrics) but fall below 0.5 on abstract categories (*ethical considerations*, *maintenance*) that require inference from sparse contexts. Appendix G.4 shows that the reference population-rate proxy accounts for 31% of the observed BERTScore variance, which identifies source sparsity as one contributor to generation difficulty. These findings motivate our multi-metric framework, which combines structural diagnostics with LLM-as-a-Judge semantic evaluation.

5.4.3 Systematic Generation Limitations

Consistently Lower Data Card Performance.

All evaluated models score lower on Data Cards than on Model Cards, with gaps ranging from 0.17 to 0.37 points (e.g., GPT-OSS-20B: 3.06 vs. 3.38; Qwen3-30B-A3B-Instruct: 4.30 vs. 4.55). Data Card schemas demand high-granularity lifecycle documentation (privacy protocols, security measures, collection procedures, and maintenance plans), whereas papers often treat datasets as ancillary artifacts and describe them sparsely. When such lifecycle facts are absent, automated systems

cannot reliably recover them without additional input from data stewards. We therefore regard Data Card generation as a particularly suitable setting for human–AI collaboration: a model can produce and source an initial draft, while responsible humans verify its claims and supply the missing lifecycle information. MetaGAI standardizes the evaluation of that initial draft and reveals which fields most require human intervention, rather than assuming that documentation can be fully automated.

Faithfulness–Completeness Tension. On Model Cards, Qwen3-30B-A3B-Instruct achieves the highest observed Faithfulness (4.50) but only 0.786 Completeness, which leaves approximately 21% of the reference fields unpopulated. This quality–coverage gap is consistent with documented long-context retrieval limitations (Liu et al., 2024b; Li et al., 2026), under which models struggle to access dispersed information. To examine whether the gap is solely an artifact of paper-only input, we conducted an ablation study (Appendix G.2) in which the model received the full multi-source context during inference. The resulting score changes are small to moderate and inconsistent in direction across judges, with no consistent improvement from the additional sources. Source access alone therefore does not resolve the evaluation-time bottleneck: models must still locate and integrate dispersed evidence. This evaluation-time result complements the construction-stage diagnostic in Section 4.1: the former compares paper-only with raw multi-source input, whereas the latter measures whether the construction retriever retained sufficient field-specific evidence. In the two fields examined in Appendix G.5, baseline outputs use more narrative and hedging language than the editor-refined references.

6 Conclusion

We introduce MetaGAI, a large-scale benchmark of 2,541 semantically matched entries constructed through multi-source triangulation and a multi-agent framework. Unlike prior work that aggregates existing documentation, MetaGAI uses editor-refined references and assesses its construction pipeline through sampled human annotation, automated diagnostics, and source-grounded evaluation of a stratified sample. Among the evaluated baselines, Qwen3-30B-A3B-Instruct achieves the best observed standardized cost–quality trade-off, while reference-based similarity metrics do not consis-

tently track judged quality. The results also reveal a recurring tension between faithfulness and field coverage. MetaGAI thus provides infrastructure for evaluating automated first drafts and human–AI collaborative documentation at scale.

Limitations

Although MetaGAI provides a large-scale benchmark for automated Model and Data Card generation, its scope remains limited. Our current framework relies on text-only generation and therefore overlooks important information embedded in figures, tables, and other non-textual modalities. In addition, MetaGAI treats document generation as a set of isolated paper-level tasks and does not model the complex dependencies among papers, models, and datasets in the broader GenAI ecosystem. Beyond these technical limitations, automated documentation also introduces potential risks. Errors or omissions in generated cards may propagate misleading signals about model capabilities, data provenance, or licensing conditions, particularly when such artifacts are reused at scale. Without careful validation and human oversight, these inaccuracies could undermine transparency efforts or lead to misplaced trust. Multimodal understanding, ecosystem-level modeling, and robust verification mechanisms remain important directions for future work.

Ethics Statement

Data Provenance and Licensing. Our benchmark uses publicly available materials from arXiv, GitHub, and Hugging Face for scientific research and card generation. We did not intentionally collect non-public personal information.

Human Studies. The validation involved volunteer graduate students with backgrounds in NLP and machine learning. All evaluators were informed about the nature of the task, and the study did not involve exposure to harmful content.

Risks of Automated Documentation. LLM-based documentation may contain hallucinations or omissions. To mitigate these risks, the Generator Agents provide evidence and source attribution, and the Editor verifies candidate drafts against the source materials. Generated cards should therefore be treated as preliminary drafts that require human verification rather than as replacements for expert review.

Information About Use of AI Assistants

AI assistants were used only for language polishing and grammar correction, not for generating scientific content, analysis, or conclusions. All content was reviewed and approved by the authors, who take full responsibility for the manuscript.

References

- Joel Castaño, Silverio Martínez-Fernández, Xavier Franch, and Justus Bogner. 2024. [Analyzing the evolution and maintenance of ml models on hugging face](#). In *Proceedings of the 21st International Conference on Mining Software Repositories, MSR '24*, page 607–618, New York, NY, USA. Association for Computing Machinery.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2025. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, et al. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. 2022. [Interactive model cards: A human-centered approach to model documentation](#). In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22*, page 427–439, New York, NY, USA. Association for Computing Machinery.
- Andy Donald, Apostolos Galanopoulos, Edward Curry, Emir Muñoz, Ihsan Ullah, M. A. Waskow, Maciej Dabrowski, and Manan Kalra. 2023. [Towards a semantic approach for linked dataspace, model and data cards](#). In *Companion Proceedings of the ACM Web Conference 2023, WWW '23 Companion*, page 1468–1473, New York, NY, USA. Association for Computing Machinery.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. [Improving factuality and reasoning in language models through multiagent debate](#). In *Forty-first International Conference on Machine Learning*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, et al. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Xiaojie Guo and Qiming Hu. 2022. [Low-light image enhancement via breaking down the darkness](#). *Int. J. Comput. Vision*, 131(1):48–66.
- Eliahu Horwitz, Nitzan Kurer, Jonathan Kahana, Liel Amar, and Yedid Hoshen. 2025. [We should chart an atlas of all the world's models](#). *Preprint*, arXiv:2503.10633.
- Jinze Li, Yang Zhang, Xin Yang, Jiayi QU, Jinfeng Xu, Shuo Yang, Junhua Ding, and Edith Cheuk-Han Ngai. 2026. [OCR-memory: Optical context retrieval for long-horizon agent memory](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10409–10420, San Diego, California, United States. Association for Computational Linguistics.
- Weixin Liang, Nazneen Rajani, Xinyu Yang, Ezinwanne Ozoani, Eric Wu, Yiqun Chen, Daniel Scott Smith, and James Zou. 2024. [Systematic analysis of 32,111 ai model cards characterizes documentation practice in ai](#). *Nature Machine Intelligence*, 6(7):744–753.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Jiarui Liu, Wenkai Li, Zhijing Jin, and Mona Diab. 2024a. [Automatic generation of model and data cards: A step towards responsible AI](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1975–1997, Mexico City, Mexico. Association for Computational Linguistics.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Shayne Longpre, Robert Mahari, Anthony Chen, Naana Obeng-Marnu, Damien Sileo, William Brannon, Niklas Muennighoff, Nathan Khazam, Jad Kabbara, et al. 2024. [A large-scale audit of dataset licensing and attribution in ai](#). *Nature Machine Intelligence*, 6(8):975–987.
- Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. [Model cards for model reporting](#). In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19*, page 220–229, New York, NY, USA. Association for Computing Machinery.
- Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. [Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict](#). *Political Analysis*, 16(4):372–403.
- NVIDIA, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, et al. 2025. [Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning](#). *Preprint*, arXiv:2512.20848.

- Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, et al. 2025. [Olmo 3. Preprint](#), arXiv:2512.13961.
- OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, et al. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, et al. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Aman Rangapur, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. 2025. [olmocr: Unlocking trillions of tokens in pdfs with vision language models](#). *Preprint*, arXiv:2502.18443.
- Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjar-tansson. 2022. [Data cards: Purposeful and transparent dataset documentation for responsible ai](#). In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22*, page 1776–1826, New York, NY, USA. Association for Computing Machinery.
- Mohammad Shahedur Rahman, Peng Gao, and Yuede Ji. 2025. [Hugginggraph: Understanding the supply chain of llm ecosystem](#). In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM '25*, page 5997–6005, New York, NY, USA. Association for Computing Machinery.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, et al. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, et al. 2020. [Hugging-face’s transformers: State-of-the-art natural language processing](#). *Preprint*, arXiv:1910.03771.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, et al. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Xinyu Yang, Weixin Liang, and James Zou. 2024. [Navigating dataset documentations in AI: A large-scale analysis of dataset cards on huggingface](#). In *The Twelfth International Conference on Learning Representations*.
- Haoxuan Zhang, Ruochi Li, Zhenni Liang, Mehri Sattari, Phat Vo, Collin Qu, Ting Xiao, Junhua Ding, Yang Zhang, and Haihua Chen. 2026. [Adage-cg: Adaptive query expansion for web-scale generative ai model and data card generation](#). In *Proceedings of the ACM Web Conference 2026, WWW '26*, page 4360–4371, New York, NY, USA. Association for Computing Machinery.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.
- Yang Zhang, Yufei Wang, Quan Z. Sheng, Adnan Mahmood, Wei Emma Zhang, and Rongying Zhao. 2023a. [Hybrid data augmentation for citation function classification](#). In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Yang Zhang, Yufei Wang, Quan Z. Sheng, Lina Yao, Haihua Chen, Kai Wang, Adnan Mahmood, Wei Emma Zhang, Munazza Zaib, et al. 2025a. Deep learning meets bibliometrics: A survey of citation function classification. *Journal of Informetrics*, 19(1):101608.
- Yang Zhang, Yufei Wang, Kai Wang, Quan Z. Sheng, Lina Yao, Adnan Mahmood, Wei Emma Zhang, and Rongying Zhao. 2023b. When large language models meet citation: A survey. *arXiv preprint arXiv:2309.09727*.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. 2025b. [Qwen3 embedding: Advancing text embedding and reranking through foundation models](#). *Preprint*, arXiv:2506.05176.
- Zhaoyang Zhang, Yang Zhang, Quan Z. Sheng, Jia Wu, and Yaolin Zhou. 2025c. [From dispersed records to structured knowledge: A graph-based knowledge modeling framework for archival resources](#). In *2025 IEEE International Conference on Web Services (ICWS)*, pages 1053–1063.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, et al. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.
- Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. 2022. [Image BERT pre-training with online tokenizer](#). In *International Conference on Learning Representations*.

A Definition of Model and Data Card

Table 4 summarizes the Model Card and Data Card fields used in MetaGAI.

Card	Field	Description
Model Card	Model Details	Model architecture, parameter count, developer organization, license, release date, and other model metadata
	Intended Use	Primary applications, target audience, out-of-scope uses, domain restrictions, and other intended uses
	Generative Capabilities	Context length, inference latency, multilingual support, supported modalities, and other capabilities
	Safety Considerations	Safety alignment methods, red teaming results, jailbreak resistance, harm reduction strategies, and other safety analysis
	Training Data	Training corpus, data mixture, filtering pipeline, data provenance, and other training data information
	Performance Metrics	Benchmark results, accuracy metrics, reasoning scores, safety scores, robustness metrics, and other quantitative metrics
	Ethical Considerations	Environmental impact, intellectual property, dual-use risks, societal implications, and other ethical issues
	Caveats & Recommendations	Hardware requirements, deployment guidelines, operational constraints, model limitations, and other recommendations
Data Card	Dataset Details	Dataset name, version identifier, creators and curators, funding, license, text language, and other dataset metadata
	Dataset Structure	Instance count, field schema, data splits, dataset size, and other structural details
	Data Collection	Collection methodology, data sources, collection timeframe, consent process, and other collection details
	Data Processing	Preprocessing steps, cleaning procedures, labeling process, filtering criteria, deduplication, and other processing steps
	Intended Uses	Primary tasks, intended use cases, prohibited uses, commercial restrictions, and other usage information
	Bias & Fairness	Demographic distribution, geographic coverage, social bias analysis, fairness assessment, and other bias or fairness information
	Privacy & Security	Personally identifiable information (PII), anonymization methods, data security protocols, confidentiality measures, and other privacy measures
	Content Analysis	Content types, toxicity analysis, misinformation risks, offensive language, and other content risks
	Legal & Ethical	Copyright status, terms of use, ethical review, compliance requirements, and other legal details
	Maintenance & Updates	Maintenance plan, update frequency, versioning policy, deprecation plan, and other maintenance information
	Distribution & Access	Access mechanism, download instructions, repository link, citation requirements, and other access details
	Limitations & Recommendations	Known limitations, usage guidelines, quality caveats, and other recommendations

Table 4: Definitions of Model Cards and Data Cards for Generative AI

B Dataset Characteristics

Figure 4 (top) illustrates the temporal distribution of the corpus, which grows from 141 triplets in 2022 to 931 in 2025. The domain composition is dominated by Computer Vision (cs.CV) and Computational Linguistics (cs.CL), reflecting our filtering criteria, which target papers with verified GitHub and Hugging Face artifacts in generative modeling. The word-count analysis (bottom) reveals distinct content characteristics across sources: academic papers provide comprehensive context (median $\approx 10k$ words), whereas GitHub and Hugging Face README files offer concise implementation details (median $< 1k$ words). This complementary information structure provides empirical support for our multi-source triangulation approach.

C MetaGAI Benchmark Generation Algorithm

Algorithm 1 formalizes one run of the three-stage MetaGAI generation pipeline. For each schema key $k_i \in \mathcal{K}$, the Retriever Agent maps segmented source chunks $\mathcal{X}$ to relevant evidence spans, which are passed to three Generator Agents that synthesize independent structured drafts. The Editor Agent then consolidates the drafts by checking them against the full multi-source context $\mathcal{P} \cup \mathcal{S}$, verifying attribution, and merging non-redundant content to produce a final card $\hat{\mathcal{C}}$. The Editor stage is executed once per editor architecture to produce the two reference sets.

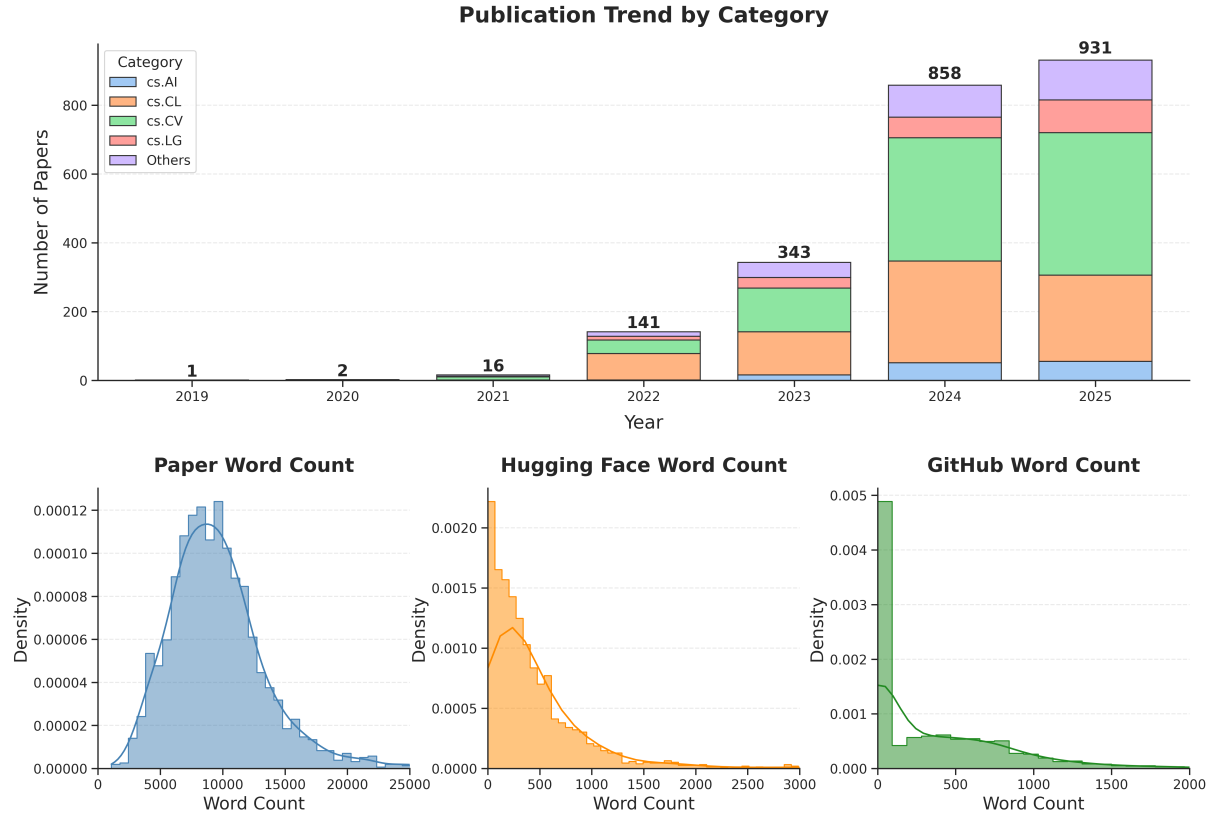


Figure 4: **Dataset Overview.** *Top:* Publication trends of the 2,541 collected triplets (2019–2025), categorized by primary arXiv domain. *Bottom:* Word-count distributions across the three data sources (academic papers, Hugging Face, and GitHub), which illustrate complementary levels of information granularity.

D MetaGAI Quality Validation Protocols

To assess generated card quality, we used a five-dimensional evaluation framework. The framework includes three established metrics (*Faithfulness*, *Relevance*, and *Accuracy*) from previous work (Liu et al., 2024a) and introduces two additional metrics (*Consistency* and *Usefulness*) that capture the structural and practical requirements of technical documentation.

D.1 Metric Definitions

Faithfulness (F). Measures the degree to which the generated content is grounded in the provided source materials (paper, GitHub, and Hugging Face), that is, the absence of unsupported claims or hallucinations.

Relevance (R). Evaluates information density and pertinence to the target field. High-scoring content adheres strictly to the field definition and avoids redundancy, verbosity, and out-of-scope information.

Accuracy (A). Assesses whether specific technical details, such as numerical values, licenses, and

entity names, match the provided source evidence.

Consistency (C). Evaluates internal logical coherence within the generated content supplied to the evaluator, detecting contradictions that compromise documentation integrity.

Usefulness (U). Measures practical value for downstream users. Useful content provides specific, actionable insights that support deployment decisions rather than generic descriptions.

D.2 Scoring Rubric

Each metric is evaluated on a 1–5 Likert scale. Table 5 presents the scoring criteria applied by both the human evaluators and the LLM judges.

D.3 Retrieval Strategy Validation: Experimental Details

Two researchers with AI/ML backgrounds independently annotated ten randomly sampled entries after segmentation, labeling whether each chunk was relevant to each schema field. Of the 6,116 candidate chunk–field pairs, the 276 pairs that both annotators agreed were relevant served as positive rele-

Metric	Score 1 (Poor)	Score 3 (Acceptable)	Score 5 (Excellent)
Faithfulness	Contains major hallucinations; claims unsupported by sources.	Mostly supported with minor extrapolations.	Fully grounded in source text; no unsupported claims.
Relevance	Largely off-topic, redundant, or contains significant noise.	Generally on-topic with some redundant details.	Concise and strictly focused on field definition.
Accuracy	Contains critical factual errors.	Core facts correct; minor imprecisions present.	Factually precise; all technical details correct.
Consistency	Contains direct logical contradictions.	Generally consistent; minor ambiguities present.	Logically coherent; no internal contradictions.
Usefulness	Vague or generic; provides no practical value.	Provides basic information but lacks depth.	Provides rich, actionable insights that support deployment.

Table 5: **Evaluation Scoring Rubric.** Criteria for assessing generated metadata cards on a 1–5 Likert scale across five quality dimensions.

vance labels for the retriever comparison. We compared BGE-Reranker-v2-m3 (Chen et al., 2025), Qwen3-Reranker-8B (Zhang et al., 2025b), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct, and Qwen3-30B-A3B-Instruct (Yang et al., 2025). We used precision at rank 1 (P@1) to measure top-result accuracy, together with recall and F1 at rank 5 (R@5, F1@5) to assess broader retrieval coverage. These fixed-cutoff measures were used only to compare candidate retrievers; the production pipeline instead retains every valid field assignment with relevance ≥ 3 . Table 1 reports the results.

D.4 Construction-Stage Propagation Diagnostic

To determine whether unpopulated benchmark fields arose from insufficient retained evidence or from later Generator or Editor failures, we performed a stage-wise diagnostic at the schema-item level. Three independent judge models assessed whether the retained context contained sufficient evidence to populate each item, and we crossed this assessment with whether the final editor-refined card populated that item. Table 6 reports four outcomes: Retrieved and Populated (R&P), Retrieved and Not Populated (R&NP), Not Retrieved and Not Populated (NR&NP), and Not Retrieved and Populated (NR&P).

Across the two reference sets, 3,216 items remain unpopulated. Of these, 2,964 (92.2%) lack sufficient evidence in the retained context (NR&NP), whereas 252 (7.8%) contain sufficient retained evidence but are missed downstream (R&NP). Conversely, when sufficient evidence is retained, downstream population succeeds for 5,901 of 6,153 items (95.9%). Benchmark incom-

pleteness is therefore associated primarily with evidence availability before final generation rather than with a failure to use evidence that has already been retained. Because evidence sufficiency is itself model-judged, this analysis provides supportive diagnostic evidence rather than a substitute for exhaustive human auditing.

D.5 Human Evaluation Protocol for Reference Quality Assessment

Evaluators and Sample. Two Ph.D. students specializing in NLP and machine learning (Ann-1 and Ann-2) independently evaluated the same stratified random sample of 100 entities (50 models and 50 datasets). For every sampled entity, they assessed the Raw Baseline, Bench_Mistral, and Bench_GPT-OSS outputs, which yielded 300 card outputs per evaluator. System identities were hidden, and cards and systems were shuffled independently to reduce ordering and expectation effects.

Evaluation Instrument. The evaluators were given the paper, GitHub repository, and Hugging Face artifact for source-grounded assessment. They checked each populated field against these materials and scored the full card on a 1–5 Likert scale along the five quality dimensions defined in Appendix D: Faithfulness, Relevance, Accuracy, Consistency, and Usefulness. The evaluators received a fixed rubric and calibration examples before the main study.

LLM Judge Configuration. Three LLM judges—Llama-3.3-70B-Instruct (Grattafiori et al., 2024), GPT-OSS-120B (OpenAI et al., 2025), and Qwen3-235B-A22B-2507 (Yang et al., 2025)—evaluated the same outputs under randomized presentation order. We queried each

Reference set	Card type	R&P	R&NP	NR&NP	NR&P
Bench_GPT-OSS	Data	1,648 (53.18%)	103 (3.32%)	958 (30.91%)	390 (12.58%)
Bench_GPT-OSS	Model	1,166 (55.60%)	52 (2.48%)	672 (32.05%)	207 (9.87%)
Bench_Mistral	Data	1,846 (59.55%)	43 (1.39%)	748 (24.13%)	463 (14.94%)
Bench_Mistral	Model	1,241 (59.15%)	54 (2.57%)	586 (27.93%)	217 (10.34%)

Table 6: Stage-wise propagation diagnostic by editor-derived reference set and card type. “Retrieved” denotes that the retained context was judged to contain sufficient supporting evidence for the schema item; “Populated” denotes that the final card contains a value for that item.

judge three times per instance with independently shuffled orderings and averaged the scores across runs to mitigate position bias.

Target Judge	Ann-1	Ann-2
Ann-2	0.439*	—
Llama-3.3-70B	0.675*	0.493*
GPT-OSS-120B	0.087 (n.s.)	0.191*
Qwen3-235B	0.703*	0.367*

Table 7: Pearson correlations across 300 card–system pairs between the human evaluators and the LLM judges. * $p < 0.001$; n.s. $p \geq 0.05$.

Inter-Annotator and Human–LLM Agreement.

Table 7 reports Pearson correlation coefficients across 300 card–system pairs. Agreement between the two human evaluators is moderate ($r = 0.44$, $p < 0.001$), consistent with the documented subjectivity of documentation-quality judgments. Among the LLM judges, Qwen3-235B and Llama-3.3-70B achieve strong alignment with Ann-1 ($r = 0.70$ and $r = 0.67$; both $p < 0.001$) and moderate alignment with Ann-2 ($r = 0.37$ and $r = 0.49$; both $p < 0.001$). GPT-OSS-120B is not significantly correlated with Ann-1 ($r = 0.09$, n.s.) and shows only weak agreement with Ann-2 ($r = 0.19$, $p < 0.001$), which argues against any single-judge design and supports our ensemble evaluation strategy.

E Inter-Judge Agreement Analysis

We computed Pearson correlations among the three judge models on the stratified evaluation set ($N = 500$). Table 8 shows substantial variation: Qwen3-235B-A22B-2507 and Llama-3.3-70B-Instruct demonstrate moderate alignment ($r = 0.540$), whereas GPT-OSS-120B exhibits markedly different scoring patterns (correlations ≤ 0.226). Although all inter-judge correlations are statistically significant ($p < 0.05$), their magnitudes show that ensemble averaging cannot by itself establish

Judge Model	Llama	GPT	Qwen
Llama-3.3-70B-Instruct	1.000	0.101 [†]	0.540*
GPT-OSS-120B	0.101 [†]	1.000	0.226*
Qwen3-235B-A22B-2507	0.540*	0.226*	1.000

Table 8: **Inter-Judge Correlation Matrix.** Pearson correlations computed on 500 samples. [†] $p < 0.05$; * $p < 0.001$. **Llama:** Llama-3.3-70B-Instruct; **GPT:** GPT-OSS-120B; **Qwen:** Qwen3-235B-A22B-2507.

judge invariance. We therefore used averaging as a bias-mitigation mechanism and explicitly tested whether the central conclusion changed when one judge was removed.

Judge-Sensitivity Analysis. Table 9 recomputes the overall quality scores from the 100-entity source-context ablation and benchmark-quality study after leaving out each LLM judge. The Raw Baseline remains the lowest-scoring system under the full three-LLM ensemble and under every leave-one-out aggregation, and the editor-refined references remain above the Raw Baseline, which supports the conclusion that editor refinement improves reference quality. The ordering among Bench_GPT-OSS, Bench_Mistral, and the Qwen ablations, however, changes under some judge subsets, particularly when Llama is omitted. We therefore base our claims on the broad editor-versus-raw pattern and do not interpret small differences among the refined systems as robust architectural rankings.

F Prompt Templates

This section presents the prompt templates used throughout the MetaGAI construction and evaluation pipeline. The templates specify the instructions, input structure, and output requirements for each agent and evaluation component, which supports consistent and reproducible implementation.

System	All three LLMs	Without GPT-OSS	Without Llama	Without Qwen
Bench_GPT-OSS	3.808	4.202	3.661	3.560
Bench_Mistral	3.840	4.312	3.444	3.766
Qwen Paper-only	3.712	4.037	3.683	3.416
Qwen All-source	3.633	4.086	3.476	3.336
Raw Baseline	3.176	3.289	3.199	3.041

Table 9: Leave-one-judge-out sensitivity on the shared 100-entity study. Each entry averages the available LLM-judge scores from Table 12; it is not a new evaluation sample.

F.1 Semantic Verification Prompt

F.2 Retriever Agent Prompt

Semantic Verification Prompt

System Instruction

You are an expert research evaluator specializing in [dataset/model] identification.

Task Description

Determine whether a research paper introduction and two README files (GitHub and Hugging Face) describe the **same** [dataset/model].

Evaluation Criteria

1. **Existence:** Does the paper explicitly introduce a specific named entity?
2. **Consistency:** Do BOTH the GitHub README and Hugging Face README reference the SAME entity as the paper?
3. **Alignment:** Compare names, domain, size metrics, methodology, and unique features.

Input Data

Paper Content: <Paper text>

GitHub README: <GitHub README text>

Hugging Face README: <HF README text>

Output Format

RELATED: [Yes/No]

CONFIDENCE: [High/Medium/Low]

EXPLANATION: [2–3 sentences citing specific evidence]

Retriever Agent Prompt

System Instruction

You are a metadata classification assistant. Output valid JSON only. STRICTLY select sub-fields from the provided ontology.

Task Description

Classify the provided text chunk into predefined metadata fields for [Dataset/Model] documentation.

Evaluation Criteria

1. **Relevance:** Assign a score (0–4) indicating how well the chunk describes the field.
2. **Sub-field Selection:** Select 1–5 sub-fields **strictly** from the provided “Selectable Sub-fields” list. Do not extract raw text.

Input Data

Source: <[PAPER] / [GITHUB] / [HUGGING-FACE]>

Content: <Text chunk content>

Output Format

```
{ "classifications": [ {
  "field": "field_name",
  "relevance": 3,
  "matched_sub_fields": ["sub_field1"] } ] }
```

Algorithm 1 MetaGAI Benchmark Generation Pipeline

Require: $\mathcal{P} \cup \mathcal{S}$: Full Context (Paper, GitHub, Hugging Face) $\mathcal{X}$: Segmented chunks of $\mathcal{P} \cup \mathcal{S}$ $\mathcal{K}$: Schema Keys $\{k_i\}_{i=1}^M$

Ensure: $\hat{\mathcal{C}}$: Final Card

Stage 1: Retriever Agent (Evidence Mapping)

- 1: $\mathcal{E} \leftarrow \emptyset$
- 2: **for** each $k_i \in \mathcal{K}$ **do**
- 3: $v_{evidence} \leftarrow \text{Retriever}(\mathcal{X}, k_i)$ $\triangleright$ Extract factual content
- 4: $\mathcal{E}[k_i] \leftarrow v_{evidence}$
- 5: **end for**

Stage 2: Generator Agents (Draft Synthesis)

- 6: $\tilde{\mathcal{C}}^{(j)} \leftarrow \emptyset$ for $j \in \{1, 2, 3\}$
- 7: **for** each $k_i \in \mathcal{K}$ **do**
- 8: **for** each generator $j \in \{1, 2, 3\}$ **do**
- 9: $\tilde{v}_i^{(j)} \leftarrow \text{Generator}_j(\mathcal{E}[k_i], \text{Prompt}_{\text{Gen}})$
- 10: $\tilde{\mathcal{C}}^{(j)}[k_i] \leftarrow \tilde{v}_i^{(j)}$
- 11: **end for**
- 12: **end for**

Stage 3: Editor Agent (Consolidation)

- 13: **for** each $k_i \in \mathcal{K}$ **do**
- 14: $\triangleright$ Editor verifies three drafts against full context
- 15: $\mathcal{D}_i \leftarrow \{\tilde{\mathcal{C}}^{(j)}[k_i]\}_{j=1}^3$
- 16: $\hat{v}_i \leftarrow \text{Editor}(\mathcal{P} \cup \mathcal{S}, \mathcal{D}_i, \text{Prompt}_{\text{Edit}})$
- 17: $\hat{\mathcal{C}}[k_i] \leftarrow \hat{v}_i$
- 18: **end for**
- 19: **return** $\hat{\mathcal{C}}$

F.3 Generator Agent Prompt

Generator Agent Prompt

System Instruction

You are an expert AI researcher. Generate specific metadata sections for a [Model/Dataset] Card. Output valid JSON only.

Critical Constraint: Subject Focus

- **Focus:** Focus solely on the entity **introduced** in this paper.
- **Subject Exclusion:** Do not generate metadata about baselines or pre-trained models as the target entity; retain comparative evidence about the proposed entity when relevant.

Input Data

Target: $\langle \text{Field Name} \rangle$ (e.g., *Training Data*) | **Context:** $\langle \text{Retrieved chunks} \rangle$

Output Requirements (Per Sub-field)

1. **Content:** Summarized factual answer.
2. **Evidence Quote:** Direct verbatim quote supporting the answer.
3. **Confidence:** [low, medium, high, certain].
4. **Source:** Provenance (e.g., "Paper+GitHub").

Output Format

```
{ "sub_field": {
  "content": "...",
  "evidence_quote": "...",
  "confidence": "high",
  "source": "Paper" } }
```

F.4 Editor Agent Prompt

Editor Agent Prompt

System Instruction

You are the Chief Editor. Consolidate three candidate drafts into ONE **concise** entry describing the **proposed** entity.

Task Description

Review three independent drafts (Candidates A, B, C) against the source evidence. Identify the correct **proposed** entity, filter out unsupported claims, and merge valid details.

Evaluation Logic

1. **Identify Proposed Entity:** Describe **ONLY** the entity introduced in **THIS** paper. Discard candidates describing baselines.
2. **Verify Evidence:** Check attribution.
3. **Conciseness:** Remove fluff; use direct facts (1–3 sentences).

Input Data

Target: $\langle \text{Field Name} \rangle$ | **Source Context:** $\langle \text{Raw chunks} \rangle$

Candidates: $\langle \text{Draft A} \rangle, \langle \text{Draft B} \rangle, \langle \text{Draft C} \rangle$

Output Format

```
{
  "selected_candidates": ["Candidate B"],
  "final_content": "...",
  "final_evidence": "...",
  "reasoning": "..." }
```

F.5 Evaluation Judge Prompt

Evaluation Judge Prompt

System Instruction

You are an expert evaluator for Generative AI documentation. Your task is to compare the generated metadata field against the provided triangulated sources (Paper, GitHub, Hugging Face).

Task Description

Evaluate the candidate text based on the five dimensions below. Assign a score from 1 (Poor) to 5 (Excellent) for each dimension.

Evaluation Metrics

- Faithfulness (F)
- Relevance (R)
- Accuracy (A)
- Consistency (C)
- Usefulness (U)

Analysis Considerations

Classify the content as one of the following: Accurate facts (verifiable), Vague facts, Logical reasoning, Illogical reasoning, or Acceptable inferences.

Input Data

Sources: <Context>; Target Field: <Field Name>; Candidate: <Text>

Output Format

```
{"scores": {"F": 5, "R": 5, "A": 5, "C": 5, "U": 5}, "reasoning": "..."}

```

G Comprehensive Analysis of Benchmark Characteristics and Baseline Behaviors

To characterize the challenges posed by MetaGAI and the behavior of the evaluated models, we analyzed information provenance, information sparsity, lexical differences, and standardized cost–quality trade-offs.

G.1 Information Provenance and Editor Dynamics

A central design principle of MetaGAI is the use of multi-source evidence where relevant. We characterize this design through a provenance analysis of the final reference fields.

Source Distribution. Table 10 shows that academic papers provide the largest share of attributed information (45–66%), that 22–31% of fields are labeled Multi-Source, and that a further 3–4% are labeled GitHub Only. Data Cards have a higher Multi-Source share (27–31%) than Model Cards (22%), which indicates that dataset documentation is more fragmented across papers and repositories in these reference sets.

Editor Strategy. Table 11 shows different consolidation patterns. The Best Match strategy dominates (80–95%), whereas the Merge strategy ac-

Source Category	Bench_Mistral		Bench_GPT-OSS	
	Model	Data	Model	Data
Paper Only	65.9%	61.9%	57.1%	45.1%
Multi-Source	22.4%	26.7%	22.2%	30.7%
GitHub Only	3.5%	3.3%	3.4%	3.9%
Not Provided	8.2%	8.1%	17.4%	20.3%

Table 10: **Information Provenance Distribution.** Percentage of metadata fields by attributed source category across the two editor-derived reference sets.

	Bench_Mistral	Bench_GPT-OSS
<i>Consolidation Strategy</i>		
Best Match	80.0%	94.9%
Merge	20.0%	5.1%
<i>Winning Candidate Source</i>		
Qwen2.5-7B	64.6%	63.8%
OLMo-3-7B	27.8%	30.3%
Llama-3.1-8B	7.6%	5.9%

Table 11: **Editor Agent Dynamics.** *Top:* The logic used by the Editor to finalize content. *Bottom:* The distribution of Generator drafts selected for the final reference.

counts for 5–20% depending on the editor architecture. The editors therefore usually select a single candidate but sometimes synthesize information from multiple generator drafts.

G.2 Ablation Study: Source Completeness vs. Model Inferential Limits

The faithfulness–completeness tension raises an important question: does the gap in field coverage arise from the model’s inferential capacity, or is it an artifact of restricting inference to paper-only context while the references were constructed from all three sources? To examine these two factors, we evaluated Qwen3-30B-A3B-Instruct, the top-performing model on the main benchmark, under two conditions on the same 100 entities used in Section 4.3:

- **Paper-only:** the standard evaluation protocol, conditioned solely on paper text $\mathcal{P}$ and consistent with the main benchmark conditions.
- **All-source:** the full multi-source context $\mathcal{P} \cup \mathcal{S}$, where $\mathcal{S}$ comprises the GitHub README and the Hugging Face artifact, that is, the same source types available during reference construction.

Results and Implications. Table 12 presents mean Likert scores under both conditions alongside

	Ann-1	Ann-2	Llama	GPT-OSS	Qwen
<i>Editor-Refined Benchmarks (Reference)</i>					
Bench_GPT-OSS	4.475	3.669	4.101	3.019	4.303
Bench_Mistral	4.355	3.799	4.634	2.897	3.990
<i>Qwen3-30B-A3B-Instruct (Ablation)</i>					
Paper-only	4.358	3.456	3.770	3.061	4.304
All-source	4.376	3.435	3.946	2.726	4.226
Δ	+0.018	-0.021	+0.176	-0.335	-0.078
Raw Baseline	3.709	2.966	3.130	2.951	3.447

Table 12: Ablation on source context. Qwen3-30B-A3B-Instruct is evaluated under paper-only and full multi-source conditions on the same 100-entity test set used in Section 4.3. The editor references and the Raw Baseline are shown for context. Score changes are inconsistent across evaluators and do not indicate a consistent benefit from adding raw multi-source context.

the editor-refined references. Providing the full multi-source context produces small to moderate but inconsistent changes: Ann-1 shifts by +0.018, Ann-2 by -0.021, Llama by +0.176, GPT-OSS by -0.335, and Qwen by -0.078. The two human evaluators and the Qwen judge show essentially no improvement, whereas the two remaining judges move in opposite directions. The all-source condition also does not consistently exceed either editor-refined reference under the human evaluators, despite receiving equivalent source types during inference.

These results show that source access alone does not resolve the evaluation-time difficulty. Even when all raw sources are supplied, the model must still locate, integrate, and populate sparse or implicitly stated metadata across heterogeneous long documents. This ablation should therefore be interpreted as evidence about input access rather than as proof that retrieval is irrelevant. The construction-stage propagation diagnostic in Table 6 addresses a different question—whether the dedicated Retriever retained sufficient field-specific evidence—and identifies evidence retention as the dominant source of unpopulated benchmark fields. Together, the two analyses distinguish construction-time evidence mapping from evaluation-time long-context inference. Under this interpretation, the paper-only protocol measures how much multi-source documentation quality can be recovered from paper input rather than performance in a fully source-matched setting.

G.3 Generator Divergence: Full Mean-Standard-Deviation Analysis

Figure 5 presents the full mean-standard-deviation distribution across all card fields. Fields such as *ethical considerations* ($\mu = 0.848$) cluster in the

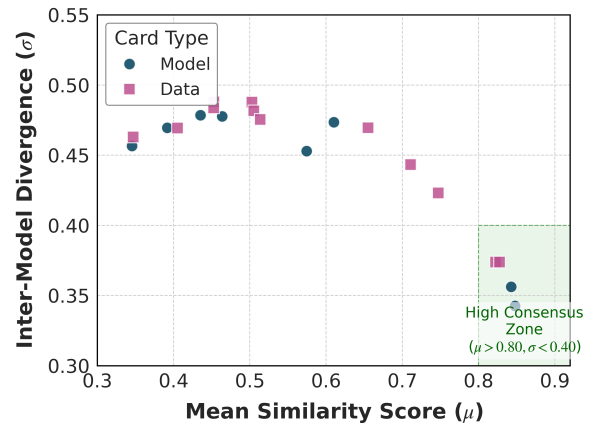


Figure 5: Mean BERTScore similarity (μ , X-axis) versus standard deviation (σ , Y-axis) across card fields. High-consensus zones reflect evidence scarcity, where generators converge on placeholder responses.

high-consensus zone, which predominantly reflects evidence scarcity rather than genuine agreement: when documentation is absent from the source materials, the generators converge on similar placeholder responses (Appendix G.4). Fields with pronounced divergence ($\sigma > 0.4$) show heterogeneous generator outputs, although divergence alone does not establish whether those differences are complementary or erroneous.

G.4 Impact of Information Sparsity

To explain the field-level performance stratification observed in Figure 3, we analyzed 20 card fields and used the reference population rate (the proportion of schema items populated in the editor-refined references) as a proxy for source information density. BERTScore served as the generation-performance indicator.

We hypothesize that the difficulty of generating abstract fields (*safety considerations*, *bias & fair-*

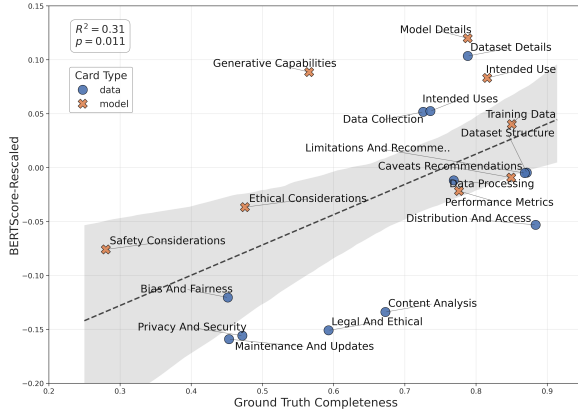


Figure 6: **Information Sparsity and Generation Performance.** Across 20 card fields, the reference population-rate proxy (X-axis) is positively associated with semantic similarity (Y-axis, BERTScore; OLS $R^2 = 0.31$, $p = 0.011$).

ness) is partly attributable to information sparsity in the sources. Higher-density fields such as *model details* provide more explicit signals, whereas abstract fields often contain sparse or implicit information.

Figure 6 shows a statistically significant positive relationship ($R^2 = 0.31$, $p = 0.011$) between the reference population-rate proxy and BERTScore across the 20 fields. The proxy accounts for 31% of the observed BERTScore variance, which leaves substantial variation attributable to other factors.

G.5 Comparative Lexical Analysis

We performed a lexical analysis using the Log-Odds Ratio (Monroe et al., 2008) to characterize lexical differences between the baseline outputs and the editor-refined references. We focused on the *model details* and *dataset details* fields, which contain open-ended, high-density technical information.

Figure 7 shows that the baseline outputs (red) use narrative and hedging terms (*suggests*, *implying*) more frequently, whereas the MetaGAI references (blue) more frequently contain technical entities (*CDLA*, *OpenRAIL*) and licensing terms. This two-field analysis characterizes lexical emphasis; it does not by itself measure factual accuracy.

G.6 Cost–Quality Analysis

To compare standardized inference prices, we analyzed the trade-off between generation quality and the Cost Index. As defined in Section 5, the index represents the OpenRouter price for one million input tokens and 0.2 million output tokens; it is not an observed per-card cost.

Figure 8 visualizes this cost–quality landscape. The observed Pareto frontier consists entirely of open-weight models. Qwen3-30B-A3B-Instruct in particular achieves the highest judged quality at a Cost Index of 0.15, whereas Gemini-2.5-Flash and GPT-5-Mini have Cost Indices of 0.80 and 0.65, respectively, without exceeding Qwen3-30B-A3B-Instruct in judged quality. Among the evaluated baselines, Qwen3-30B-A3B-Instruct therefore offers the most favorable standardized price–quality trade-off.

H Case Study

We present two qualitative case studies that illustrate how multi-source triangulation and editor-based synthesis can address information omissions and incomplete evidence coverage.

H.1 Multi-Source Triangulation

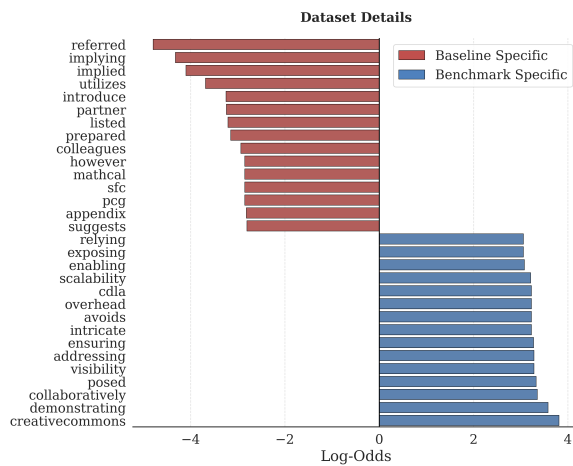
We analyze the generation of the *model architecture* field for the paper “Low-light Image Enhancement via Breaking Down the Darkness” (Guo and Hu, 2022). As Figure 9 illustrates, high-fidelity output requires information from three distinct sources: the paper defines the macro-level architectural topology, GitHub specifies implementation hyperparameters that are typically omitted from manuscripts, and Hugging Face confirms entity alignment and deployment availability.

This case illustrates how the pipeline can bridge conceptual descriptions and implementation details. The paper provides the architectural framework, GitHub provides implementation-specific hyperparameters, and Hugging Face metadata supports entity alignment. The Editor Agent (GPT-OSS-20B) selects Candidate B because it incorporates the relevant evidence shown in the example.

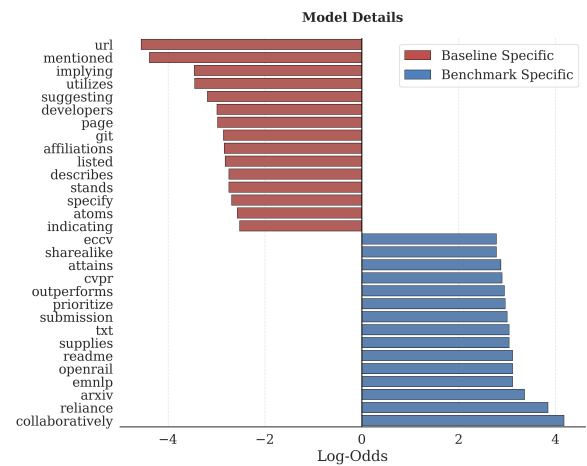
H.2 Editor-Driven Information Synthesis

Even when relevant evidence is retrieved, individual Generator Agents may focus on different parts of the source context, and the Editor Agent can merge non-redundant content from their drafts.

Continuing with the same paper, we analyze the *robustness metrics* field for the “Bread” model. As Figure 10 shows, two of the three Generator drafts provide complementary evidence: Candidate A (OLMo-3-7B) reports comparative benchmark results, whereas Candidate B (Qwen2.5-7B) reports findings from a component ablation. Rather than selecting a single candidate, the Editor Agent (GPT-OSS-20B) merges the non-redundant information.

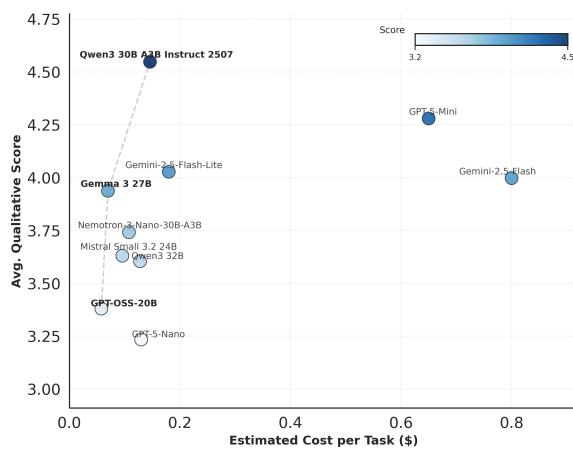


(a) Dataset Details Divergence

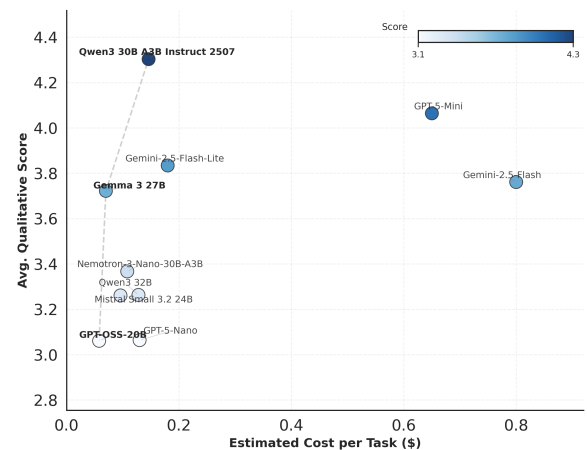


(b) Model Details Divergence

Figure 7: **Lexical Divergence (Log-Odds Ratio)**. **Left (Red)**: words over-represented in the baseline outputs, consistent with more narrative-oriented phrasing. **Right (Blue)**: words over-represented in the MetaGAI references, including technical entities and specifications.



(a) Model Card Generation



(b) Data Card Generation

Figure 8: **Standardized Cost–Quality Analysis**. The Pareto frontier (dashed grey line) highlights the models that offer the highest observed quality at a given Cost Index.

This case illustrates a potential benefit of editor-based synthesis; it does not establish a general advantage over all single-model approaches.

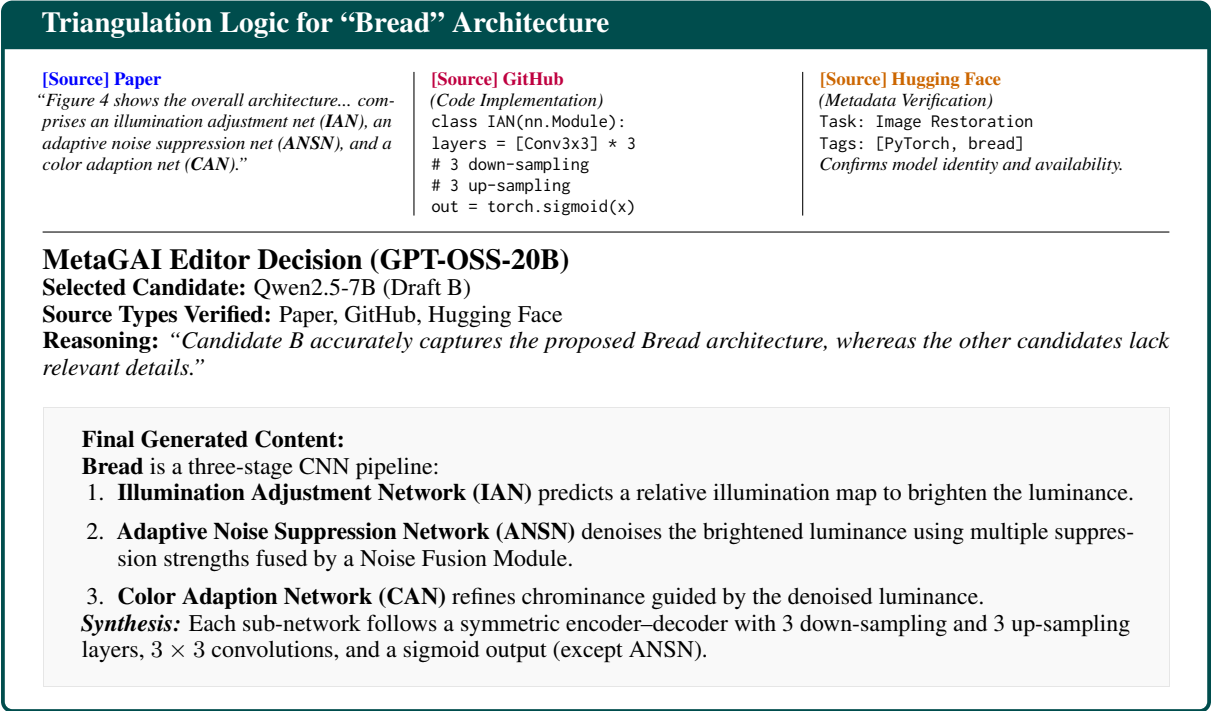


Figure 9: **Multi-Source Triangulation.** The Editor Agent combines high-level topology from the paper with implementation details from GitHub, while Hugging Face metadata supports entity alignment.

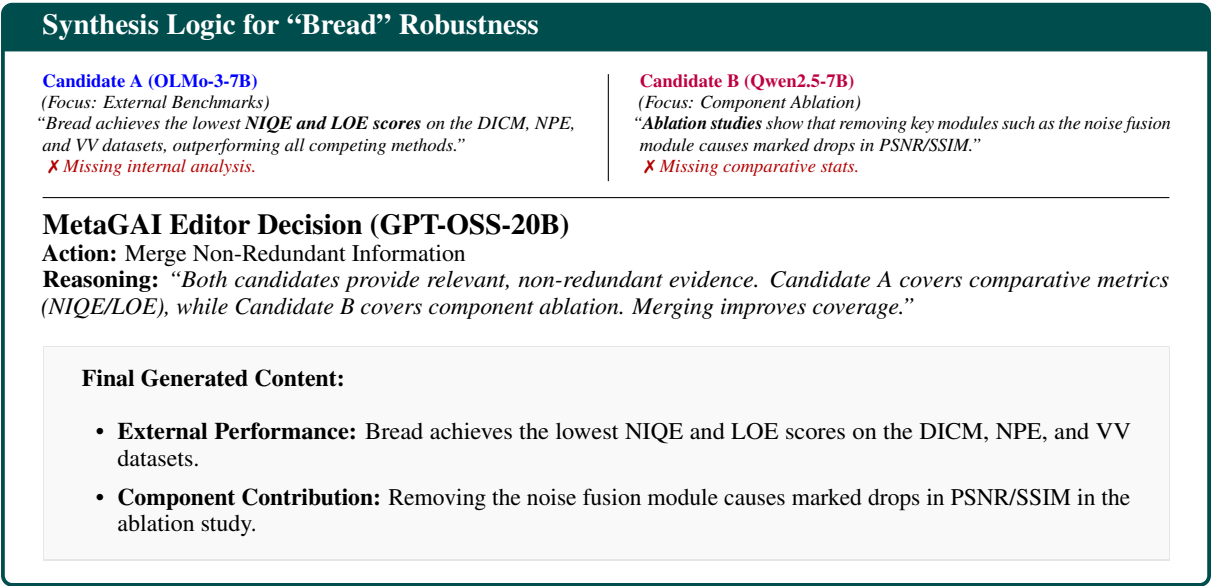


Figure 10: **Resolving Incompleteness via Synthesis.** Candidates A and B provide non-redundant evidence from comparative benchmarks and a component ablation. The Editor merges them into a single entry.