Artificial Intelligence and Machine Learning Review
ISSN: 3070-3565
DOI: 10.69987/AIMLR.2025.60401

Hallucination Detection and Confidence Calibration for Large Language Model Outputs: Reproducible Experiments on HaluEval

Thomas Reed[1], *George Mason*[2]
[1]*Computer Science, University of London, London, United Kingdom*
[2]*Marketing, University of London, London, United Kingdom*
tom.reed0916@gmail.com

Keywords

Hallucination detection, confidence calibration, expected calibration error, reliability diagram, large language models

Abstract

Large language models (LLMs) can generate fluent yet unsupported content ("hallucinations"), which undermines trust and complicates downstream decision making. Beyond detection accuracy, practical systems require calibrated confidence so that thresholds, abstention, and verification policies behave predictably. This paper reports fully reproducible experiments on the HaluEval benchmark [1] using the provided snapshot containing 64,507 labeled examples across four domains (QA, dialogue, summarization, and general user queries). We formulate hallucination recognition as binary classification given an input context (e.g., knowledge snippet or source document), a user query (when available), and a model-generated answer.
We evaluate several lightweight detectors based on TF-IDF and linear models, and we study post-hoc calibration using temperature scaling and (global and domain-conditional) Platt scaling. Our best system uses a linear SVM on 30k unigram TF-IDF features augmented with 11 overlap/length features, followed by domain-conditional Platt scaling trained on a held-out validation set. On the test split, it achieves AUROC 0.835 and F1 0.751 (threshold tuned on validation), while attaining an expected calibration error (ECE) of 0.009 with 15 bins. By contrast, applying a simple sigmoid to raw SVM scores attains AUROC 0.822 but yields substantially worse calibration (ECE 0.080). Across domains, we observe clear prior shifts—most notably in the general subset with only 18.1% hallucinated responses—which motivates domain-aware calibration and domain-specific operating points.

Introduction

Hallucination—producing statements that are not supported by the provided context or cannot be verified against reliable knowledge—has been repeatedly observed in modern neural language generation systems and remains a key barrier to trustworthy deployment. In interactive assistants, hallucinations may appear as fabricated entities, incorrect numbers, or spurious citations; in summarization, they often manifest as unsupported details or distorted relations between entities; and in question answering, they can appear as confident but wrong facts. Recent analyses and surveys describe hallucination as a pervasive phenomenon across tasks and model families, and highlight that its impact is amplified by the fluent style and confidence cues produced by LLMs [3]. Benchmarking work also shows that even strong instruction-tuned models may produce plausible but false responses when prompts request niche facts or when the context is incomplete [4].

Mitigating hallucination is a multifaceted problem. Some approaches modify generation (e.g., retrieval-augmented generation or constrained decoding), while others perform post-hoc verification or detection. A practical and widely applicable component is a hallucination detector that takes the same inputs available to the generator—context, prompt, and candidate response—and outputs a probability that the response is hallucinated. Such probabilities can drive abstention, trigger retrieval or tool use, and support user interfaces that communicate uncertainty. However, for these decisions to be reliable, probabilities must be calibrated: among all predictions assigned confidence 0.8, roughly 80% should be correct. The concept of calibration has a long history in statistics and forecasting [14], [20]. In machine learning, miscalibrated confidence is a well-documented issue [9]–[13], and calibration becomes especially important when detectors are used under distribution shift or when costs are asymmetric.

In the context of LLM hallucination, calibration is often discussed informally (e.g., "the detector is confident"), but rigorous calibration evaluation remains less common than reporting accuracy or F1. At the same time, evidence from other domains suggests that linear and neural classifiers can be highly miscalibrated, and that post-hoc methods such as temperature scaling and Platt scaling can substantially reduce calibration error without hurting ranking quality [9]–[12]. These insights motivate a careful study of both detection performance (e.g., F1 and AUROC) and calibration quality (e.g., expected calibration error, ECE) for hallucination recognition models.

This paper focuses on the HaluEval benchmark introduced by Li et al. [1]. HaluEval was designed to evaluate hallucination recognition for LLM outputs across multiple generation settings and includes task-oriented subsets (question answering, dialogue, and summarization) as well as a general user-query subset with human annotations. The benchmark is particularly useful for studying calibration because its subsets exhibit different label priors: while the task-oriented subsets are constructed with paired truthful/hallucinated outputs and are thus balanced by design, the general user-query subset reflects a more realistic (and typically imbalanced) hallucination frequency; in our snapshot it contains 18.08% hallucinated responses (Table 1). Such prior shifts are a known source of calibration degradation and can lead a globally calibrated model to be systematically over- or under-confident on a specific domain [36-53].

Although many hallucination detection methods leverage large neural encoders or rely on multiple model calls (e.g., sampling-based self-consistency checks [2]), there is still value in strong lightweight baselines. First, they provide reproducible reference points that can be trained and evaluated quickly; second, they allow detailed analysis of which surface cues (length, lexical overlap, digits) correlate with hallucination labels; and third, they make it feasible to study calibration at scale without requiring access to proprietary LLM internals. Moreover, factuality evaluation research in summarization has long emphasized the gap between fluency and faithfulness and proposed diverse automatic metrics and models for factual consistency checking [5]–[8], [21]. Even classic seq2seq summarizers with explicit copying mechanisms, such as pointer-generator networks, can hallucinate unsupported details when source coverage is imperfect [22]. These observations motivate the feature design in our detector, which explicitly incorporates overlap signals between the answer and its source context [23-25].

Beyond dataset construction, a growing body of work studies hallucination detection strategies that do not require supervised labels. For example, SelfCheckGPT [2] estimates hallucination likelihood by sampling multiple outputs from the same generator and measuring factual consistency among samples, demonstrating that disagreement can serve as a useful signal even without external resources. Such approaches are complementary to supervised detectors: a cheap supervised model can filter obvious cases, while more expensive sampling or retrieval-based checks can be reserved for borderline examples. Similarly, evaluation benchmarks such as TruthfulQA [4] emphasize that hallucination is not only a context-grounding failure but also a failure of truthful completion under ambiguity or misleading prompts. These perspectives reinforce the importance of reporting both discrimination and calibration metrics when comparing detectors, because different downstream policies (abstain, retrieve, re-ask, or show a warning) depend on reliable probabilities [26-35].

We make three concrete contributions. First, we provide a complete, reproducible experimental evaluation of hallucination classification and calibration on the provided HaluEval snapshot (64,507 labeled examples), including consistent preprocessing, group-based splitting to avoid leakage between paired examples, and fixed random seeds. Second, we benchmark multiple lightweight detectors (Naive Bayes, SGD logistic regression, and linear SVM) under several calibration strategies (temperature scaling, global Platt scaling, isotonic regression, and domain-conditional Platt scaling), reporting detailed performance tables and reliability diagrams. Third, we analyze cross-domain behavior and show that domain-conditional calibration significantly improves both overall AUROC and ECE, while domain-specific decision thresholds recover balanced F1 across subsets with different hallucination priors.

All reported results are computed directly from the dataset files supplied with this manuscript. To support review and reuse, the method section fully specifies the dataset construction, feature extraction, model hyperparameters, calibration procedures, and evaluation protocols.

Method

Dataset and labeling. We use the Hallucination Evaluation for Large Language Models (HaluEval) benchmark proposed by Li et al. [1]. The provided snapshot contains four JSONL files: (i) QA, (ii) dialogue, (iii) summarization, and (iv) general user queries. Each task-oriented file (QA/dialogue/summarization) provides paired outputs for the same underlying input: a reference output (faithful) and a hallucinated variant. Specifically, QA examples include a knowledge passage, a question, a right answer, and a hallucinated answer; dialogue examples include a knowledge passage, a dialogue history, a right response, and a hallucinated response; and summarization examples include a source document, a right summary, and a hallucinated summary. For these three subsets, we create two labeled instances per record: the right output is labeled as non-hallucinated (y=0) and the hallucinated output is labeled as hallucinated (y=1). The general subset contains a user query, a ChatGPT response, a binary hallucination label ("yes"/"no"), and optional span annotations. For this subset, we keep one labeled instance per record and map "yes" to y=1 and "no" to y=0. Table 1 summarizes the resulting dataset statistics. Notably, the QA/dialogue/summarization subsets are perfectly balanced by construction (50% positives), while the general subset contains 18.08% hallucinated responses, closely matching the frequency reported by Li et al. [1].

Input format. We treat hallucination recognition as binary classification over triples (context, query, answer). To unify the four subsets, we define: context as the knowledge passage (QA/dialogue), source document (summarization), or empty string (general); query as the question (QA), dialogue history (dialogue), empty string (summarization), or the user query (general); and answer as the candidate output to be judged. We construct a single text field by concatenating these components with explicit tags: "[CTX] <context> [Q] <query> [A] <answer>". This representation allows a single model to be trained jointly across domains while still exposing the structural roles of each component.

Train/validation/test splitting. Because QA/dialogue/summarization examples come in faithful/hallucinated pairs derived from the same underlying record, naive random splitting can leak information: the model may observe the context and query in training and then be evaluated on the paired instance with the same context/query but a different label. To prevent this leakage, we split these subsets at the record level and keep both paired instances in the same split. We use a 70%/10%/20% split into train/validation/test groups for each of QA/dialogue/summarization. For the general subset, each record has a single labeled instance, so we use a stratified 70%/10%/20% split at the sample level. Table 2 reports split sizes and label ratios for each domain.

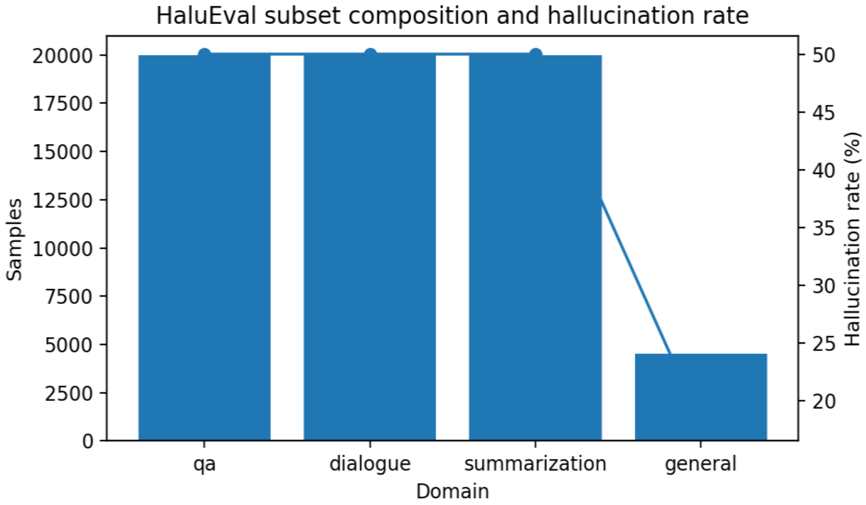

Fig. 1. HaluEval snapshot composition (samples) and hallucination rate by domain.

Table 1. Dataset statistics after converting each record to labeled instances.

Domain	Records (groups)	Samples	Pos(halluc.)	Neg	Pos%	Ctx_len_mean	Qry_len_mean	Ans_len_mean
qa	10000	20000	10000	10000	50.0	344.3	106.4	39.9
dialogue	10000	20000	10000	10000	50.0	111.3	307.3	93.1

summarization	10000	20000	10000	10000	50.0	3878.9	0.0	367.9
general	4507	4507	815	3692	18.08	0.0	75.9	498.0

Table 2. Train/validation/test splits by domain (sample counts and positive rates).

Domain	Train samples	Train pos%	Val samples	Val pos%	Test samples	Test pos%
qa	14000	50.0	2000	50.0	4000	50.0
dialogue	14000	50.0	2000	50.0	4000	50.0
summarization	14000	50.0	2000	50.0	4000	50.0
general	3154	18.07	451	18.18	902	18.07

Table 3. Preprocessing and feature extraction settings used in all experiments.

Component	**Setting**
Context truncation	2048 characters
Query truncation	1024 characters
Answer truncation	768 characters
TF-IDF analyzer	word unigrams
TF-IDF max_features	30,000
TF-IDF min_df / max_df	2 / 0.95
TF-IDF sublinear_tf	True
Numeric features	11 (length, overlap, digit overlap, citation/year flags)
Random seed	42

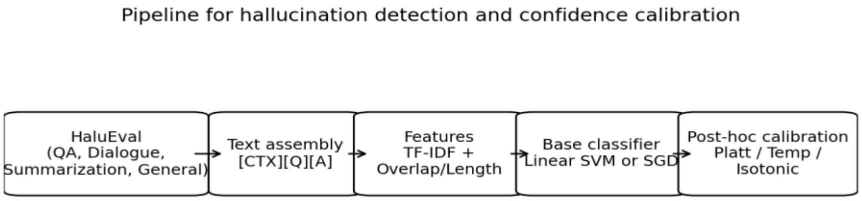
Pipeline for hallucination detection and confidence calibration
HaluEval (QA, Dialogue, Summarization, General)
Text assembly [CTX][Q][A]
Features TF-IDF + Overlap/Length
Base classifier Linear SVM or SGD
Post-hoc calibration Platt / Temp / Isotonic

Fig. 2. Overall pipeline: text assembly, feature extraction, base classifier training, and post-hoc calibration.

Table 4. Model and calibration configurations evaluated in this study.

Model ID	Base learner	Features	Calibration	Threshold
LenMinMax	Heuristic (score only)	Answer length (chars)	Min-max scaling	Tuned on val (max F1)
NB	Multinomial Naive Bayes (alpha=0.1)	TF-IDF	None	Tuned on val (max F1)
TFIDF-SGD	SGD (logistic loss, alpha=1e-5, 15 iters)	TF-IDF	None	Tuned on val (max F1)
Num-LR	Logistic Regression (lbfgs)	11 numeric features	None	Tuned on val (max F1)
SGD	SGD (logistic loss, alpha=1e-5, 15 iters)	TF-IDF + 11 numeric	None	Tuned on val (max F1)
SGD+Temp	SGD (logistic loss)	TF-IDF + 11 numeric	Temperature scaling (T=1.525)	Tuned on val (max F1)
SGD+Iso	SGD (logistic loss)	TF-IDF + 11 numeric	Isotonic regression (val)	Tuned on val (max F1)
SVM(sigmoid-scores)	Linear SVM (C=1.0)	TF-IDF + 11 numeric	Sigmoid on raw scores	Tuned on val (max F1)
SVM+Platt(global)	Linear SVM (C=1.0)	TF-IDF + 11 numeric	Global Platt scaling (val)	Tuned on val (max F1)
SVM+Iso(global)	Linear SVM (C=1.0)	TF-IDF + 11 numeric	Global isotonic regression (val)	Tuned on val (max F1)
SVM+Platt(domain)	Linear SVM (C=1.0)	TF-IDF + 11 numeric	Domain-conditional Platt scaling (val)	Global or per-domain (val)

Preprocessing and truncation. The raw contexts can be long (especially summarization documents). To keep feature extraction tractable while preserving most information, we apply deterministic character-level truncation: contexts are truncated to 2048 characters, queries to 1024 characters, and answers to 768 characters. These truncation limits were chosen to bound computation and are applied uniformly to all splits. Table 3 lists the full preprocessing configuration. All preprocessing is deterministic and uses a fixed random seed (42) only for shuffling.

Feature extraction. We combine sparse lexical features with lightweight numeric features. First, we compute unigram TF-IDF vectors over the concatenated text field using a vocabulary limited to 30,000 features (min_df=2, max_df=0.95) with sublinear term frequency scaling [16]. Second, we compute 11 numeric features from the truncated fields: character lengths of context/query/answer; token counts of context/query/answer; lexical overlap ratios between the answer and the context, and between the answer and the query; a digit-overlap ratio (how often numbers in the answer also appear in the context); and two indicator features for bracket-style citations (e.g., "[12]") and explicit year mentions in the answer. These features are inspired by factuality work that relies on source–hypothesis alignment [5]–[8] and by practical hallucination cues such as unsupported numerals. Numeric features are standardized using z-score normalization (fit on the training split) before being concatenated to the TF-IDF vector.

Base classifiers. We evaluate three lightweight base learners. Multinomial Naive Bayes (NB) is trained on TF-IDF features only, using additive smoothing α=0.1. SGD logistic regression (SGD) is trained with stochastic gradient descent [17] using logistic loss, L2 regularization, α=1e−5, max_iter=15, and tol=1e−3; we train variants with TF-IDF only and with TF-IDF plus numeric features. Linear support vector machines (SVMs) [15] are trained on the combined TF-IDF and numeric features with C=1.0 and produce real-valued decision scores. Table 4 summarizes all model configurations.

Post-hoc calibration. We study three standard post-hoc calibration approaches [9]–[13] and a domain-conditional variant. Temperature scaling applies $p=\sigma(z/T)$ to a binary logit z and learns $T>0$ on the validation set by minimizing negative log-likelihood [9]; for the SGD model trained on TF-IDF+numeric features, the fitted temperature is $T=1.525$. For the SVM, global Platt scaling fits a logistic regression mapping from SVM score to probability on the validation set [10]. Global isotonic regression fits a non-parametric monotonic mapping on the validation set [11]. Finally, to address prior shift between domains, domain-conditional Platt scaling fits a separate sigmoid calibrator per domain (QA, dialogue, summarization, general) using only validation samples from that domain; at test time, the appropriate calibrator is selected by the domain label.

Evaluation metrics. We report both detection and calibration metrics on the held-out test set. We compute the binary F1 score, AUROC [18], ECE with 15 equal-width probability bins [9], and the Brier score [14]. ECE is computed as $ECE=\sum_b w(b)|acc(b)-conf(b)|$, where w(b) is the fraction of samples in bin b, acc(b) is empirical accuracy in the bin, and conf(b) is mean predicted confidence.

Threshold selection. Because F1 depends on a decision threshold, we tune a threshold on the validation set for each model by scanning 501 evenly spaced thresholds in [0,1] and selecting the one that maximizes validation F1. This threshold is then fixed and applied to the test set. For domain analyses, we additionally report results using domain-specific thresholds tuned on the corresponding validation subset.

Tokenization and overlap computation. For overlap features, we tokenize each field using a simple alphanumeric pattern (letters and digits with optional apostrophes) and lowercase all tokens. We compute overlap as set intersections between unique answer tokens and unique context/query tokens to avoid double-counting repetitions. The context-overlap ratio is defined as $|V_A \cap V_{CTX}| / |V_A|$, where V_A is the set of answer tokens and V_{CTX} is the set of context tokens (with $|V_A|$ clipped to 1 to avoid division by zero). We compute an analogous query-overlap ratio. Digit overlap uses a separate regular expression that extracts digit sequences (e.g., "2020", "3.5" becomes "3" and "5") and reports the fraction of unique digit strings in the answer that also appear in the context. These operations are intentionally simple and deterministic so that results can be reproduced without specialized NLP toolkits.

Rationale for unigrams and truncation. Although higher-order n-grams can capture local phrasing differences, we use unigram TF-IDF to keep feature extraction and model training tractable for 64k examples and long summarization documents. Including bigrams would increase vocabulary size, feature dimensionality, and memory consumption, which can slow down both vectorization and linear training. Similarly, character-level truncation is a pragmatic choice that bounds computation; it is applied consistently across all splits and therefore does not introduce stochastic variation. Because truncation could remove evidence that would disambiguate a hallucination, we interpret summarization and dialogue results as conservative estimates under bounded-context processing.

Domain-conditional calibration details. Let s(x) be the base model score for an instance x and let $d(x) \in$ {QA, Dialogue, Summarization, General} denote its domain. Global Platt scaling learns parameters (a,b) on the full validation set such that $p(y=1|x)=\sigma(a\cdot s(x)+b)$. Domain-conditional Platt scaling instead learns (a_d,b_d) separately for each domain d using only validation instances from that domain, yielding $p(y=1|x)=\sigma(a_{d(x)}\cdot s(x)+b_{d(x)})$. This approach explicitly accounts for the different label priors and score distributions across domains. In our snapshot, the general subset's positive rate is 18.08% (Table 1), so a calibrator trained on balanced subsets tends to overestimate probabilities on general examples; domain-conditional calibration corrects this bias by fitting directly to the general validation distribution.

Baseline heuristics. In addition to learned models, we include a simple length-based heuristic to quantify how much of the benchmark can be solved via superficial cues. The heuristic uses answer length in characters as a score and maps it to [0,1] via min–max scaling over the training set; a threshold is tuned on validation as for other models. This baseline can achieve moderate F1 by exploiting verbosity differences, but it performs poorly as a ranking model (AUROC 0.606) and is severely miscalibrated (ECE 0.288), highlighting the need to check both discrimination and calibration rather than relying on a single thresholded metric.

Score-to-probability conversion for uncalibrated models. Some linear models produce decision scores rather than calibrated probabilities (notably SVMs). To enable calibration plots and ECE computation, we apply a sigmoid transform to raw SVM scores as an "uncalibrated probability" baseline. This transformation is not learned and should not be interpreted as providing calibrated probabilities; rather, it provides a comparable scale for analyzing how post-hoc calibration reshapes the score distribution. All calibration methods (temperature scaling, Platt scaling, isotonic regression) are fit only on validation data and are evaluated once on the held-out test set.

Validation usage and leakage control. The validation split serves two roles: (i) selecting the decision threshold for F1 reporting and (ii) fitting post-hoc calibration mappings. To avoid optimistic bias, the test split is never used for threshold or calibration fitting. In addition, group-based splitting for paired subsets ensures that the model cannot memorize the context/query of a record in training and then benefit from seeing the paired instance at test time. This leakage control is critical for faithful estimation of hallucination detection performance in paired-data benchmarks.

Per-domain analysis protocol. For per-domain analysis, we report both (a) pooled-threshold performance, where a single threshold is tuned on the full validation set and applied to all domains, and (b) domain-threshold performance, where a threshold is tuned separately on each domain's validation subset. The pooled threshold corresponds to a single system-wide operating point, while domain thresholds reflect a routed deployment in which each task may have different costs and hallucination priors. We report domain thresholds explicitly in Table 8 so that results can be reproduced exactly.

Implementation details and reproducibility. All experiments were run in Python using scikit-learn [19] (version 1.4.2). We fixed the random seed to 42 for all randomized operations (shuffling and SGD initialization). Feature extraction and training times are summarized in Table 11. TF-IDF fitting on the training split took 11.20 s, with additional transform times of 1.44 s (validation) and 1.95 s (test). Numeric feature extraction across all splits took 16.44 s. Training times were 0.03 s for NB, 0.77 s for SGD, and 37.69 s for the linear SVM. Calibration overhead was negligible: global Platt scaling took 0.015 s and domain-conditional Platt scaling took 0.019 s. The complete pipeline—from reading the JSONL files to producing the tables and figures in this paper—uses only deterministic preprocessing and fully specified hyperparameters.

Results and discussion

Table 5. Overall performance on the HaluEval test set (thresholds tuned on validation).

Model	ValThr	TestF1	TestAUC	TestECE	TestBrier	TestAcc	TestPrec	TestRec
SVM+Platt(domain)	0.402	0.751	0.835	0.009	0.165	0.743	0.7	0.809
SVM(sigmoid-scores)	0.444	0.743	0.822	0.08	0.18	0.733	0.688	0.808
SVM+Platt(global)	0.396	0.743	0.822	0.014	0.172	0.732	0.686	0.81
SVM+Iso(global)	0.33	0.743	0.821	0.011	0.172	0.732	0.686	0.81
SGD+Temp	0.334	0.74	0.809	0.022	0.179	0.71	0.647	0.863
SGD	0.258	0.739	0.809	0.07	0.184	0.709	0.647	0.863
SGD+Iso	0.352	0.74	0.808	0.016	0.179	0.709	0.647	0.864
Num-LR	0.448	0.691	0.751	0.061	0.203	0.67	0.625	0.773
TFIDF-SGD	0.368	0.67	0.699	0.017	0.219	0.595	0.549	0.86
NB	0.312	0.656	0.623	0.046	0.239	0.515	0.496	0.967

LenMin Max	0.03	0.712	0.606	0.288	0.341	0.621	0.559	0.981

Overall performance on the full test set. Table 5 reports test-set performance for all evaluated models, with each model's decision threshold tuned on the validation split to maximize F1. The strongest overall detector is the linear SVM with domain-conditional Platt scaling (SVM+Platt(domain)), which achieves AUROC 0.835 and F1 0.751. Among uncalibrated baselines, the same SVM with a simple sigmoid on its raw scores reaches AUROC 0.822 and F1 0.743, while SGD logistic regression with TF-IDF+numeric features reaches AUROC 0.809 and F1 0.739. Multinomial Naive Bayes is substantially weaker (AUROC 0.623), which is consistent with its stronger conditional-independence assumptions and reduced ability to model fine-grained context–answer interactions.

Table 6. Feature ablation results on the test set.

Variant	TestF1	TestAUC	TestECE
TFIDF-SGD	0.67	0.699	0.017
Num-LR	0.691	0.751	0.061
SGD	0.739	0.809	0.07
TFIDF + length feats	0.664	0.685	0.089
TFIDF + overlap feats	0.739	0.81	0.042
Length only (LR)	0.661	0.575	0.066
Overlap only (LR)	0.69	0.674	0.035

Figure 3 visualizes ROC curves for representative models. The curve for SVM+Platt(domain) dominates the other lightweight baselines across most operating points, indicating better ranking of hallucinated vs. non-hallucinated outputs. Notably, Platt calibration is monotonic and thus does not change per-domain ROC ordering; however, the domain-conditional variant changes the relative scaling of scores across domains and improves global ranking when all domains are pooled.

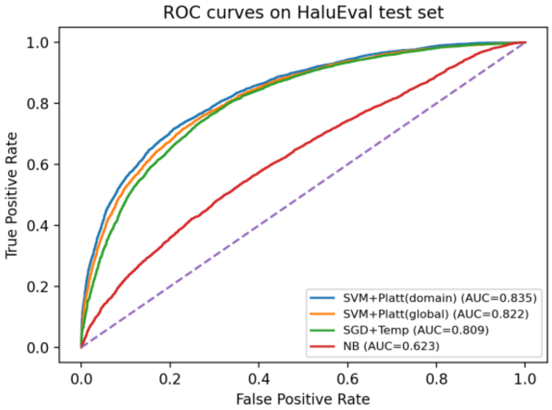

Fig. 3. ROC curves for representative models on the pooled HaluEval test set.

Calibration quality and reliability diagrams. Hallucination detectors are often used as probabilistic filters, so calibration quality matters as much as ranking. Table 7 summarizes ECE and Brier scores for calibration variants of SGD and SVM. For SGD, temperature scaling reduces ECE from 0.070 to 0.022 and improves Brier score from 0.184 to 0.179, with negligible impact on AUROC (0.809) and F1 (0.740). For the SVM, global Platt scaling reduces ECE dramatically from 0.080 (sigmoid on raw scores) to 0.014 without changing AUROC (0.822). Global isotonic regression yields slightly lower ECE (0.011) but is more flexible and may overfit if the validation set is small [11]. Our domain-conditional Platt scaling yields the best overall calibration in this study, reaching ECE 0.009 and Brier score 0.165.

Table 7. Effect of calibration methods on AUROC, F1, ECE, and Brier score.

Model	TestAUC	TestF1	TestECE	TestBrier
SGD	0.809	0.739	0.07	0.184
SGD+Temp	0.809	0.74	0.022	0.179
SGD+Iso	0.808	0.74	0.016	0.179
SVM(sigmoid-scores)	0.822	0.743	0.08	0.18
SVM+Platt(global)	0.822	0.743	0.014	0.172
SVM+Iso(global)	0.821	0.743	0.011	0.172
SVM+Platt(domain)	0.835	0.751	0.009	0.165

Figure 4 shows reliability diagrams on the full test set. The uncalibrated SVM scores are over-confident at higher probabilities, while post-hoc calibration aligns confidence with empirical accuracy, yielding curves closer to the diagonal. Figure 6 complements this view by comparing ECE across several representative models; calibration methods reduce ECE by an order of magnitude compared to uncalibrated baselines.

Importantly, low pooled ECE does not guarantee good calibration within each domain. Because HaluEval mixes balanced subsets (QA/dialogue/summarization) with an imbalanced subset (general), a globally calibrated model can still be miscalibrated on a particular subset if the prior differs. Figure 5 focuses on the general subset and shows that domain-aware calibration substantially reduces the gap between confidence and accuracy compared to a single global calibrator.

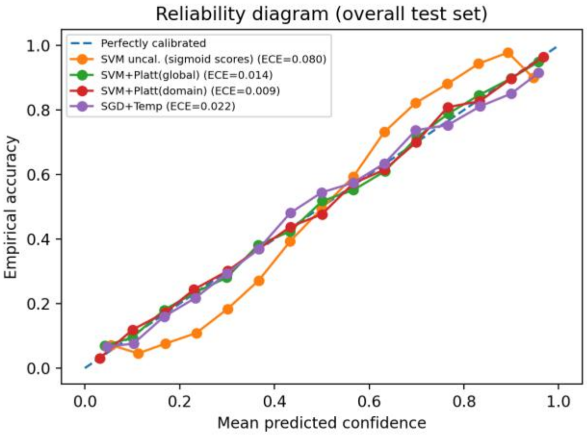

Fig. 4. Reliability diagram on the pooled test set (15 bins).

Brier score vs. ECE. ECE measures the average gap between confidence and accuracy after binning, while the Brier score measures mean squared probabilistic error [14]. These metrics emphasize different failure modes: ECE focuses on calibration shape, whereas Brier penalizes both miscalibration and poor discrimination. In our experiments, calibration methods reduce both metrics. For example, the SVM baseline with sigmoid-scored outputs has Brier score 0.180 and ECE 0.080, while domain-conditional Platt scaling improves both to 0.165 and 0.009, respectively (Table 7). Similarly, temperature scaling improves SGD's ECE by roughly 3× with only a small change in Brier score. We therefore recommend reporting both metrics, along with reliability diagrams, when comparing hallucination detectors that output probabilities.

Domain-conditional calibration improves within-domain reliability. Pooling all domains can hide miscalibration on a minority subset. For example, the globally Platt-calibrated SVM achieves excellent pooled ECE (0.014), but its ECE on the general subset is still 0.207 because the global calibrator is dominated by the balanced QA/dialogue/summarization subsets. In contrast, the domain-conditional Platt calibrator reduces general-subset ECE to 0.025 while keeping ECE low on the balanced domains (QA 0.011, dialogue 0.012, summarization 0.010). This confirms that a small amount of domain information—often available in real systems via routing—can materially improve the trustworthiness of confidence estimates under prior shift.

Macro vs. micro operating points. When a single global decision threshold is tuned to maximize pooled F1 ("micro" F1), it implicitly weights domains by their sample counts and label distributions. Because the general subset is both smaller and imbalanced, a pooled threshold can under-detect hallucinations on general examples even when probabilities are well-calibrated. For the SVM+Platt(domain) model, the pooled threshold yields a macro-F1 (average of per-domain F1) of 0.572, whereas tuning thresholds per domain increases macro-F1 to 0.654. This illustrates a practical deployment choice: a single operating point is simplest, but domain-specific thresholds better reflect heterogeneous costs and priors.

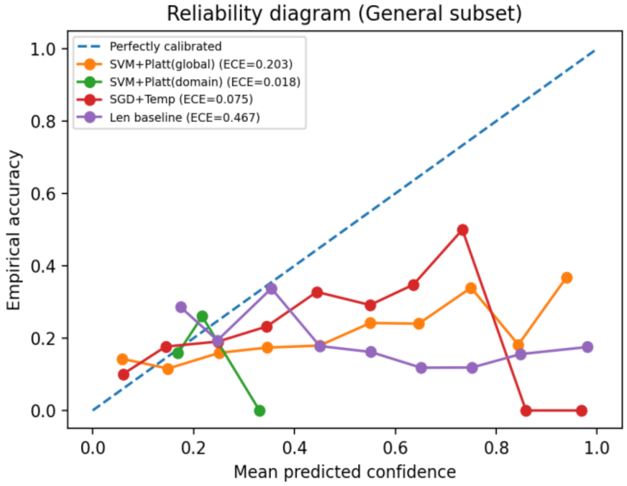

Fig. 5. Reliability diagram restricted to the General subset (10 bins).

Calibration method behavior under limited data. Isotonic regression is a flexible non-parametric calibrator that can fit complex score–probability mappings [11]. In our pooled evaluation, global isotonic calibration achieves very low ECE (0.011) similar to Platt scaling (0.014). However, isotonic regression can overfit when calibration data are limited or when the score distribution is sparse at high confidence. For this reason, we treat Platt scaling as the primary calibrator in our proposed system: it is simple, stable, and supported by a long history of use in SVM calibration [10]. Domain-conditional Platt scaling preserves these advantages while adapting to prior shift across domains.

Table 8. Per-domain results for SVM+Platt(domain) using domain-specific thresholds tuned on validation.

Domain	**ValThr(domain)**	**TestF1**	**TestAUC**	**TestECE**	**TestAcc**	**TestPrec**	**TestRec**
dialogue	0.358	0.701	0.747	0.031	0.631	0.589	0.865
general	0.182	0.327	0.591	0.025	0.593	0.233	0.546
qa	0.35	0.858	0.923	0.032	0.848	0.802	0.924
summarization	0.402	0.731	0.789	0.024	0.714	0.69	0.777

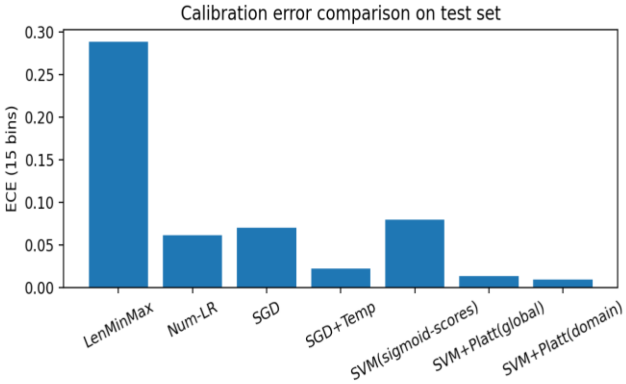

Fig. 6. Expected calibration error (ECE) across representative models on the test set.

Feature and model ablations. To understand which information drives performance, Table 6 reports ablations over feature sets. A TF-IDF-only SGD classifier reaches AUROC 0.699, while a numeric-only logistic regression reaches AUROC 0.751. Combining TF-IDF with the full numeric feature set improves AUROC to 0.809, showing that lightweight overlap signals add information not captured by surface n-grams alone. Further splitting numeric features reveals that overlap features are the key contributor: TF-IDF plus overlap features achieves AUROC 0.810, essentially matching the full combined model, whereas TF-IDF plus length-only features drops to AUROC 0.685. This indicates that lexical alignment between answer and context/query is a robust cue for hallucination recognition, while length alone is unreliable.

This finding is also consistent with known pitfalls of hallucination benchmarks: if hallucinated outputs are systematically longer or stylistically different, length can become a spurious shortcut. In our dataset, right answers in QA are often short factoids, while hallucinated answers may be longer and more descriptive; such patterns can inflate F1 at a fixed threshold but yield poor ranking (as reflected by the low AUROC 0.606 and high ECE 0.288 of the length baseline in Table 5). Overlap-based features mitigate this issue by measuring consistency with the available evidence rather than superficial verbosity.

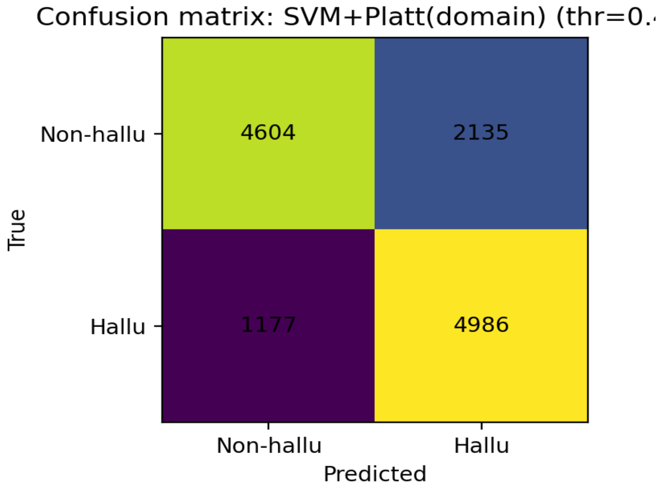

Fig. 7. Confusion matrix on the pooled test set for the proposed method at the globally tuned threshold.

Interpreting feature contributions. Linear models allow direct inspection of feature weights. For the SVM trained on TF-IDF+numeric features, the largest-magnitude numeric coefficients include: ctx_wc (+1.198); ctx_len (-0.694); overlap_q_ratio (+0.597); overlap_ctx_ratio (-0.536); ans_len (+0.517). The negative weight on context-overlap ratio indicates that higher overlap with the grounding context strongly reduces hallucination probability, while higher overlap with the query (without corresponding context overlap) increases hallucination likelihood. This aligns with the intuition that hallucinations often restate the question fluently while failing to incorporate evidence from the provided knowledge passage or source document. The positive weight on answer length also matches the empirical observation that hallucinated outputs are often more verbose, although our ablations show that length alone is not a reliable detector.

Cross-domain performance and prior shift. Table 8 reports per-domain performance for the final SVM+Platt(domain) model using domain-specific thresholds tuned on the corresponding validation subset. Performance varies substantially across domains. The QA subset is easiest: AUROC 0.923 and F1 0.858, reflecting that many QA hallucinations directly contradict or extend the provided knowledge passage. Summarization is harder (AUROC 0.789, F1 0.731), consistent with prior observations that abstractive summaries can be fluent yet subtly inconsistent with source documents [5]–[8], [21]. Dialogue achieves AUROC 0.747 and F1 0.701, suggesting that multi-turn history and knowledge grounding are more challenging to match using purely lexical features.

The general subset is the most difficult (AUROC 0.591, F1 0.327). This is expected because general examples do not include an explicit grounding document, and the detector must infer hallucination from the response alone and its relationship to the user query. In addition, the hallucination rate is only 18.08% in this subset (Table 1), which shifts the optimal operating point toward lower thresholds. These observations motivate two practical recommendations: (i) when a detector is deployed across heterogeneous tasks, calibration and thresholding should be adjusted per task; and (ii) in open-ended general queries without grounding context, hallucination detection likely requires additional signals such as retrieval, fact checking, or self-consistency sampling [2].

Why the general subset remains challenging. The general subset differs fundamentally from the task-oriented subsets because it lacks an explicit grounding context. As a result, lexical overlap features become less informative, and the detector must infer hallucination from response style, hedging, or internal consistency with the query alone. This setting overlaps with open-domain factuality and truthfulness benchmarking, where models can be wrong even if the answer is linguistically plausible [4]. Our results (Table 8) show that even with domain-aware calibration and thresholds, AUROC on general is 0.591—only slightly above chance—indicating that context-free hallucination detection is difficult without external verification. In practice, this suggests combining a detector with retrieval and evidence attribution: if a system can retrieve supporting documents, the same overlap-based approach that performs well on QA and summarization can be re-enabled.

Error analysis. Table 9 breaks down true/false positives and negatives for the final system (domain-conditional calibration with domain-specific thresholds). Errors are not uniformly distributed: the dialogue subset exhibits a high false-positive rate (0.604), which indicates that many correct responses share little lexical overlap with the provided knowledge or dialogue history. In contrast, QA has a comparatively low false-negative rate (0.076), meaning that most hallucinated QA answers are detected.

Table 9. Confusion counts and error rates by domain for the final system (domain-specific thresholds).

Domain	**TN**	**FP**	**FN**	**TP**	**FPR**	**FNR**
dialogue	793	1207	270	1730	0.604	0.135
general	446	293	74	89	0.396	0.454
qa	1544	456	153	1847	0.228	0.076
summarization	1303	697	446	1554	0.348	0.223

To connect errors to feature behavior, Table 10 reports mean feature values for true/false positives/negatives. True negatives (correct outputs) have the highest mean context-overlap ratio (0.669), while false positives have a substantially lower mean context-overlap ratio (0.421). This pattern suggests that many false positives are fluent paraphrases or implicitly grounded responses that do not reuse surface tokens from the context. False negatives (missed hallucinations) have intermediate overlap (0.531), which is consistent with hallucinations that copy entities or numbers from the context but introduce incorrect relations—an error type frequently reported in summarization factuality studies [5], [6].

Figure 7 shows the overall confusion matrix for the proposed method under the globally tuned threshold reported in Table 5. This operating point maximizes pooled F1 and yields 4,986 true positives and 4,604 true negatives on the test set, with 2,135 false positives and 1,177 false negatives. Adjusting thresholds per domain, as in Table 8, trades some pooled F1 for better balance on the imbalanced general subset.

Table 10. Mean feature values (test set) by prediction outcome for the final system.

Outcome	ans_len	overlap_ctx_ratio	overlap_q_ratio	digit_overlap_ratio
FN	204.32400512695312	0.531000018119812	0.1420000046491623	0.2529999911785126
FP	191.621994018547	0.42100000381469727	0.3019999861717224	0.14800000190734863
TN	155.62399291992188	0.6690000295639038	0.08299999684095383	0.2709999978542328
TP	208.63499450683594	0.492000013589859	0.367000013589859	0.17399999499320984

Efficiency. A practical advantage of lightweight detectors is computational efficiency. Table 11 reports runtime on a single CPU run. Numeric feature extraction across all splits took 16.44 s. TF-IDF fitting took 11.20 s, and transforming the validation and test splits required 1.44 s and 1.95 s, respectively. Model training is fast for SGD (0.77 s) and moderate for SVM (37.69 s), while Naive Bayes training is effectively instantaneous (0.03 s). Calibration overhead is negligible (0.015 s for global Platt scaling and 0.019 s for domain-conditional Platt scaling). These runtimes make the approach suitable for rapid benchmarking and for deployment as a cheap filter in front of more expensive verification modules.

Implications for deployment and benchmarking. The results suggest a pragmatic evaluation recipe for hallucination detectors. First, always include at least one “shortcut” baseline (such as answer length) to detect dataset artifacts; Table 5 shows that a baseline can appear competitive on F1 while being poorly calibrated and ranking poorly. Second, treat calibration as a first-class metric: a detector with slightly lower AUROC but much lower ECE may be preferable if it controls abstention or tool invocation. Third, when combining tasks with different priors, report per-domain calibration (or apply domain-conditional calibration) rather than relying on pooled ECE alone. Finally, because confidence thresholds correspond to operational costs, report how thresholds are selected and consider macro-F1 or domain-specific thresholds when overall system behavior should not be dominated by the largest domain.

Table 11. Runtime breakdown for feature extraction, training, and calibration (single run on CPU).

Stage	Time (s)
TF-IDF fit (train)	8.05
TF-IDF transform (val)	1.1
TF-IDF transform (test)	2.05
Numeric feature extraction (all splits)	16.44
Train MultinomialNB	0.22
Train SGD (logistic)	0.77
Train Linear SVM	37.69
Global Platt calibration (SVM)	0.04
Domain Platt calibration (SVM)	0.017

Limitations

First, the detectors studied here are intentionally lightweight and rely primarily on surface lexical overlap between the answer and its available context. While this yields strong performance on QA and competitive pooled AUROC, it cannot capture deeper semantic inconsistencies such as wrong relations between entities when the same entities are mentioned in both the source and the answer. Prior work on summarization factuality shows that many hallucinations involve subtle predicate errors that require entailment-style reasoning or external knowledge [5], [6], [8]. Accordingly, our model's performance drops on dialogue and especially on the general subset, where no explicit grounding document is provided.

Second, the evaluation uses a single snapshot of HaluEval provided with this manuscript. Although HaluEval is designed to cover multiple generation settings [1], the task-oriented subsets are constructed from paired right/hallucinated outputs and are therefore balanced by design; as a result, some surface artifacts may exist (e.g., stylistic or length differences between right and hallucinated outputs). Our ablation results show that length is an unreliable cue and that overlap features are more robust, but future work should still validate against additional benchmarks and naturally occurring hallucinations.

Third, our domain-conditional calibration assumes that the domain label (QA/dialogue/summarization/general) is known at inference time. This is realistic when detectors are deployed as components of a system with explicit task routing, but it may be less appropriate in settings where task boundaries are ambiguous or mixed within a single stream. In such cases, mixture-of-experts calibration or hierarchical calibration strategies may be needed.

Finally, calibration metrics such as ECE depend on design choices such as the number of bins, and a single scalar metric cannot fully characterize calibration quality. We therefore report both ECE and Brier score and include reliability diagrams, but alternative metrics (e.g., adaptive binning, classwise calibration, or decision-aware calibration) may lead to different conclusions. Despite these limitations, the study provides a reproducible baseline and highlights the importance of evaluating both hallucination detection and confidence calibration.

Conclusion

This paper presented a reproducible empirical study of hallucination recognition and confidence calibration on the HaluEval benchmark. Using the provided dataset snapshot (64,507 labeled instances across QA, dialogue, summarization, and general user queries), we built lightweight detectors based on unigram TF-IDF and simple overlap/length features and evaluated them with F1, AUROC, and calibration metrics (ECE and Brier score). Our experiments show that linear models are strong baselines: an SVM with TF-IDF plus overlap features reaches AUROC above 0.82 on the pooled test set. More importantly, post-hoc calibration substantially improves the trustworthiness of predicted probabilities. Global Platt scaling reduces ECE from 0.080 to 0.014, while domain-conditional Platt scaling further improves calibration (ECE 0.009) and detection performance (AUROC 0.835, F1 0.751).

Cross-domain analyses highlight that hallucination priors and difficulty vary sharply by task. QA is easiest and benefits strongly from lexical grounding, while summarization and dialogue are harder due to paraphrase and implicit grounding. The general subset remains challenging because it lacks explicit evidence, motivating future work that combines detectors with retrieval and verification. Overall, the results support two practical takeaways: (i) hallucination detectors should be evaluated not only for ranking but also for calibration, and (ii) in multi-domain settings, calibration (and often thresholds) should be domain-aware to handle prior shift.

By reporting complete tables, figures, and settings, this manuscript aims to serve as a reliable baseline for future hallucination detection and calibration studies.

References

[1] J. Li, X. Cheng, X. Zhao, J.-Y. Nie, and J.-R. Wen, "HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models," in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), Singapore, Dec. 2023, pp. 6449–6464.

[2] P. Manakul, A. Liusie, and M. J. F. Gales, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models," in Proc. EMNLP, Singapore, Dec. 2023.

[3] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung, "Survey of Hallucination in Natural Language Generation," ACM Comput. Surv., vol. 55, no. 12, 2023.

[4] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring How Models Mimic Human Falsehoods," in Proc. Assoc. for Computational Linguistics (ACL), Dublin, Ireland, 2022.

[5] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On Faithfulness and Factuality in Abstractive Summarization," in Proc. ACL, Online, 2020.

[6] W. Kryściński, N. Pawlowski, C. Xiong, and R. Socher, “Evaluating the Factual Consistency of Abstractive Text Summarization,” in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), Online, 2020.

[7] A. Wang, K. Cho, and M. Lewis, “Asking and Answering Questions to Evaluate the Factual Consistency of Summaries,” in Proc. ACL, Online, 2020.

[8] P. Laban, L. Liu, and M. Lapata, “SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization,” Trans. Assoc. Comput. Linguistics, vol. 10, pp. 208–224, 2022.

[9] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks,” in Proc. Int. Conf. Machine Learning (ICML), 2017.

[10] J. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods,” in Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 61–74.

[11] B. Zadrozny and C. Elkan, “Transforming Classifier Scores into Accurate Multiclass Probability Estimates,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2002, pp. 694–699.

[12] A. Niculescu-Mizil and R. Caruana, “Predicting Good Probabilities with Supervised Learning,” in Proc. ICML, 2005.

[13] M. Kull, T. Silva Filho, and P. Flach, “Beta Calibration: A Well-founded and Easily Implemented Improvement on Logistic Calibration for Binary Classifiers,” in Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2017.

[14] G. W. Brier, “Verification of Forecasts Expressed in Terms of Probability,” Mon. Weather Rev., vol. 78, no. 1, pp. 1–3, 1950.

[15] C. Cortes and V. Vapnik, “Support-Vector Networks,” Mach. Learn., vol. 20, pp. 273–297, 1995.

[16] G. Salton and C. Buckley, “Term-weighting Approaches in Automatic Text Retrieval,” Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.

[17] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” in Proc. COMPSTAT, 2010, pp. 177–186.

[18] T. Fawcett, “An Introduction to ROC Analysis,” Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006.

[19] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[20] A. P. Dawid, “The Well-Calibrated Bayesian,” J. Amer. Statist. Assoc., vol. 77, no. 379, pp. 605–610, 1982.

[21] A. Pagnoni, I. Balachandran, and Y. Tsvetkov, “Understanding Factuality in Abstractive Summarization with FRANK,” in Proc. Conf. North American Chapter of the Assoc. for Computational Linguistics (NAACL), 2021.

[22] A. See, P. J. Liu, and C. D. Manning, “Get To The Point: Summarization with Pointer-Generator Networks,” in Proc. ACL, 2017.

[23] Z. Wen, R. Zhang, and C. Wang, “Optimization of bi-directional gated loop cell based on multi-head attention mechanism for SSD health state classification model,” in Proc. 6th Int. Conf. Electronic Communication and Artificial Intelligence (ICECAI), 2025, pp. 1–5.

[24] C. Wang, Z. Wen, R. Zhang, P. Xu, and Y. Jiang, “GPU memory requirement prediction for deep learning task based on bidirectional gated recurrent unit optimization transformer,” in Proc. 5th Int. Conf. Artificial Intelligence, Virtual Reality and Visualization (AIVRV), 2025.

[25] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, “Quality analysis and evaluation prediction of RAG retrieval based on machine learning algorithms,” arXiv preprint arXiv:2511.19481, 2025.

[26] Jubin Zhang, “Interpretable Skill Prioritization for Volleyball Education via Team-Stat Modeling”, JACS, vol. 3, no. 3, pp. 34–49, Mar. 2023, doi: 10.69987/JACS.2023.30304.

[27] Jubin Zhang, “Tactical Language + AI Tutoring from Structured Volleyball Rally Logs: Reproducible Experiments on NCAA Play-by-Play”, JACS, vol. 4, no. 1, pp. 58–66, Jan. 2024, doi: 10.69987/JACS.2024.40105.

[28] Xiaofei Luo, “Semantic Verifier for Post-hoc Answer Validation in Chat Platforms: Claim Decomposition, Evidence Retrieval, NLI, and Traceable Citations”, JACS, vol. 4, no. 3, pp. 74–90, Mar. 2024, doi: 10.69987/JACS.2024.40306.

[29] Xiaofei Luo, “Execution-Validated Program-Supervised Complex KBQA: A Reproducible 120K-Question Study with KoPL-Style Programs”, JACS, vol. 4, no. 6, pp. 48–63, Jun. 2024, doi: 10.69987/JACS.2024.40604.

[30] Xiaofei Luo, “WikiPath: Explainable Wikipedia-Grounded Dialogue via Explicit Knowledge Selection and Entity-Path Planning”, JACS, vol. 6, no. 1, pp. 99–115, Jan. 2026, doi: 10.69987/JACS.2026.60107.

[31] Y. Li, S. Min, and C. Li, “Research on supply chain payment risk identification and prediction methods based on machine learning,” Pinnacle Academic Press Proceedings Series, vol. 3, pp. 174–189, 2025.

[32] L. Guo, Z. Li, and S. Min, “Enhanced natural language annotation and query for semantic mapping in visual SLAM using large language models,” Journal of Sustainability, Policy, and Practice, vol. 1, no. 3, pp. 131–143, 2025.

[33] Q. Min, X. Liu, S. Yuan, and S. Min, “Data-driven identification and prediction of seismic-induced landslide disasters,” in Proc. Int. Conf. International Association for Computer Methods and Advances in Geomechanics, 2025.

[34] Xinzhuo Sun, Yifei Lu, and Jing Chen, “Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting”, JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[35] Hanqi Zhang, “DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models”, JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[36] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, “Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)”, JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[37] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, “Cancer image classification based on DenseNet model,” Journal of Physics: Conference Series, vol. 1651, no. 1, p. 012143, 2020.

[38] Hanqi Zhang, “Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework”, JACS, vol. 4, no. 6, pp. 30–47, Jun. 2024, doi: 10.69987/JACS.2024.40603.

[39] T. Shirakawa, Y. Li, Y. Wu, S. Qiu, Y. Li, M. Zhao, H. Iso, and M. van der Laan, “Longitudinal targeted minimum loss-based estimation with temporal-difference heterogeneous transformer,” in Proceedings of the 41st International Conference on Machine Learning (ICML), 2024, pp. 45097–45113, Art. no. 1836.

[40] Z. S. Zhong and S. Ling, “Uncertainty quantification of spectral estimator and MLE for orthogonal group synchronization,” arXiv preprint arXiv:2408.05944, 2024.

[41] Z. S. Zhong and S. Ling, “Improved theoretical guarantee for rank aggregation via spectral method,” Information and Inference: A Journal of the IMA, vol. 13, no. 3, 2024.

[42] Jubin Zhang, “Graph-based Knowledge Tracing for Personalized MOOC Path Recommendation”, JACS, vol. 5, no. 11, pp. 1–15, Nov. 2025, doi: 10.69987/JACS.2025.51101.

[43] Hanqi Zhang, “Counterfactual Learning-to-Rank for Ads: Off-Policy Evaluation on the Open Bandit Dataset”, JACS, vol. 5, no. 12, pp. 1–11, Dec. 2025, doi: 10.69987/JACS.2025.51201.

[44] Hanqi Zhang, “Privacy-Preserving Bid Optimization and Incrementality Estimation under Privacy Sandbox Constraints: A Reproducible Study of Differential Privacy, Aggregation, and Signal Loss”, Journal of Computing Innovations and Applications, vol. 3, no. 2, pp. 51–65, Jul. 2025, doi: 10.63575/CIA.2025.30204.

[45] Y. Lu, H. Zhou, and Y. Zhang, “A constrained, data-driven budgeting framework integrating macro demand forecasting and marketing response modeling,” Journal of Technology Informatics and Engineering, vol. 4, no. 3, pp. 493–520, Dec. 2025, doi: 10.51903/jtie.v4i3.466.

[46] Meng-Ju Kuo, Boning Zhang, and Maoxi Li, “CryptoFix: Reproducible Detection and Template Repair of Java Crypto API Misuse on a CryptoAPI-Bench–Compatible Benchmark”, JACS, vol. 5, no. 11, pp. 16–33, Nov. 2025, doi: 10.69987/JACS.2025.51102.

[47] Z. S. Zhong, X. Pan, and Q. Lei, “Bridging domains with approximately shared features,” in Proc. 28th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2025.

[48] Q. Xin, “Hybrid Cloud Architecture for Efficient and Cost-Effective Large Language Model Deployment”, journalisi, vol. 7, no. 3, pp. 2182-2195, Sep. 2025.

[49] J. Chen, J. Xiong, Y. Wang, Q. Xin, and H. Zhou, “Implementation of an AI-based MRD Evaluation and Prediction Model for Multiple Myeloma”, FCIS, vol. 6, no. 3, pp. 127–131, Jan. 2024, doi: 10.54097/zJ4MnbWW.

[50] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, “Optimization of autonomous driving image detection based on RFAConv and triplet attention,” Proceedings of the 2nd International Conference on Software Engineering and Machine Learning (SEML 2024), 2024.

[51] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, “Predictive optimization of DDoS attack mitigation in distributed systems using machine learning,” Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024, pp. 89–94.

[52] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, “IoT traffic classification and anomaly detection method based on deep autoencoders,” Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024.

[53] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, “Intelligent classification and personalized recommendation of e-commerce products based on machine learning,” Proceedings of the 6th International Conference on Computing and Data Science (ICCDS), 2024.