Dual-Model Language Systems for Reliable Knowledge Generation and Hallucination Detection in Semiconductor Applications

Anonymous ACL submission

Abstract

Large language models (LLMs) show promise as intelligent assistants in semiconductor engineering, yet hallucinations—fabricated or incorrect facts—make them risky for high-stakes workflows. We present a dual-model framework with a domain-adapted *generator* and a recall-oriented *classifier* that filters unreliable outputs. The generator is Qwen2.5-14B continued-pretrained on a ∼200M-token semiconductor corpus and augmented with a Chat Vector (model merging) to strengthen reasoning; the classifier is fine-tuned on ∼8k QA triplets (factual vs. realistic hallucinations). On a 1,000-QA benchmark, our pipeline lifts binary factual accuracy from 82.0% to 87.13% with only 200–350 ms latency overhead. Ablations show complementary gains from continued pretraining (CP), reasoning alignment (CV), and classifier filtering (CLF). The framework provides a practical path to safer, non-RAG deployment of LLMs in semiconductor tasks.

1 Introduction

The semiconductor industry is a cornerstone of the global economy, enabling advances in computing, communications, transportation, and artificial intelligence. As the foundation of modern electronics, it underpins devices ranging from smartphones and datacenters to medical systems and aerospace technologies. Fabrication processes are extraordinarily precise: even minor errors in process parameters—such as misreporting a plasma etching ratio, misidentifying the order of lithography steps, or confusing etch/deposition selectivity—can cascade into wasted wafers, yield reduction, or multimillion-dollar losses. Given that leading-edge manufacturing hubs such as Taiwan, Korea, and the United States anchor worldwide supply chains, reliable knowledge assistance in semiconductor workflows is not only a national priority but also a global imperative.

Large language models (LLMs) promise an attractive interface for semiconductor engineers: they can answer process-design questions, summarize long technical documents, and explain mechanisms in natural language. However, hallucinations—fabricated, contradictory, or overly confident but wrong statements—remain a key obstacle to adoption in such high-stakes environments. An LLM that suggests the wrong gas chemistry or an invalid recipe step is not merely inaccurate; it is potentially catastrophic in an industrial setting.

Recent advances in instruction tuning and reasoning transfer have improved fluency, but factual reliability is not guaranteed. Generic benchmarks such as TruthfulQA or FactCC measure broad tendencies toward hallucination, yet they fail to capture the unique risks in semiconductor contexts. Unlike open-domain QA, this domain requires *fine-grained factuality*: exact gas ratios, temperature ranges, or layer orderings must be correct to the digit or step. Slight deviations are unacceptable.

Ensuring reliability of LLMs in high-stakes scientific and engineering settings has emerged as a central challenge. While medicine, law, and finance have begun to see domain-adapted models, semiconductor engineering remains underexplored despite its economic importance. This work closes that gap by introducing both a large-scale semiconductor corpus and a factuality benchmark, enabling the first systematic study of hallucination mitigation in this domain.

Unlike many recent approaches that rely on retrieval-augmented generation (RAG), our study deliberately excludes retrieval. This allows us to directly test whether a simple generator–classifier pipeline, without external search infrastructure, can already deliver strong factual reliability. By isolating the effect of our core components, the analysis shows whether hallucination reduction can be achieved without the added complexity of retrieval mechanisms.

Contributions. This work addresses these challenges with a dual-model framework that explicitly separates generation and verification. Our contributions are:

1. **Framework.** A generator–classifier pipeline where a domain-adapted LLM produces answers and a hallucination detector filters unreliable outputs.

2. **Data resources.** A ∼200M-token continued-pretraining corpus of semiconductor texts, and a hallucination-detection dataset of ∼8k QA triplets with realistic but incorrect answers.

3. **Evaluation.** A 1,000-QA benchmark with metrics including binary factual accuracy, classifier recall/false positives, and latency overhead. Results show accuracy improves from 82.0% to 87.13% with modest latency cost (200–350 ms).

2 Related Work

Domain-adaptive pretraining. Adapting foundation models to specific domains has been shown to improve both factuality and representation learning. For instance, SciBERT (Beltagy et al., 2019) improves scientific NLP, while BioGPT (Luo et al., 2022) and PubMedBERT (Gu et al., 2021) advance biomedical tasks. FinBERT (Araci, 2019) adapts models to finance, and similar approaches have been explored for legal (Chalkidis et al., 2020), chemical (Gupta et al., 2022), and materials-science corpora (Tshitoyan et al., 2019). These works highlight the value of domain grounding in high-stakes settings. Our work extends this tradition to the semiconductor domain, where no dedicated corpora or benchmarks previously existed. By continuing pretraining on ∼200M tokens, we provide the first large-scale domain resource, grounding terminology such as etch chemistries, deposition precursors, and lithography parameters. We also evaluate against Semikong-8B/70B (Nguyen et al., 2024), which are industry-specific LLMs trained on semiconductor corpora. We follow their released inference configuration (temperature 0, greedy decoding) for fair comparison.

Reasoning and alignment. Instruction-tuned models such as GPT-3.5, LLaMA-2 (Touvron et al., 2023), and Mistral (Jiang et al., 2023) demonstrate improved reasoning with chain-of-thought prompting (Wei et al., 2023) or distilled reasoning (DeepSeek-AI Team, 2025). Reasoning transfer has also been studied via lightweight model merging, where capability vectors are injected without full-scale fine-tuning. This has been applied for multilingual transfer (Hu et al., 2021; Muennighoff et al., 2022) and arithmetic reasoning (Magister et al., 2022). We adopt this *Chat Vector* approach (Huang et al., 2024) to import reasoning ability from DeepSeek-R1-Distill-Qwen-14B while preserving our domain pretraining, showing that reasoning and factuality can be composed rather than learned jointly from scratch.

Hallucination detection. Hallucination mitigation has been approached through verifier models (Kadavath et al., 2022), self-consistency (Wang et al., 2023), and LLM-as-a-judge frameworks (Zheng et al., 2023). Other strategies include retrieval grounding (Lewis et al., 2020), reranking (RARR (Gao et al., 2023)), or self-checking (SelfCheckGPT (Manakul et al., 2023)). These approaches often require multiple generations, external corpora, or expensive post-hoc filtering. In contrast, our classifier is domain-specific and recall-oriented: it directly predicts whether a candidate answer is factually supported by the question, rejecting unsafe outputs instead of repairing them. This recall-first policy prioritizes safety, which is critical in semiconductor workflows where incorrect guidance cannot be tolerated.

Positioning within factuality research. Domain-specific risks vary: incorrect drug–disease associations in biomedicine (Jimenez-Gutierrez et al., 2022), misreported numbers in finance (Cachola et al., 2020), or flawed reasoning in legal settings (Holzenberger et al., 2023). In semiconductor contexts, wrong process parameters can translate directly into wafer loss. Our study is the first to address factuality control in this setting, introducing both a large-scale domain corpus and a hallucination-specific evaluation benchmark.

Unlike retrieval-heavy solutions, this work deliberately avoids external search to test whether a generator–classifier pipeline alone can ensure safety while keeping latency practical.

Broader scope. Our work aligns with emerging research on safe deployment of LLMs in high-stakes domains such as healthcare (Miotto et al., 2023), law (Bommarito and Katz, 2022), and engineering (Rao et al., 2023). These studies emphasize

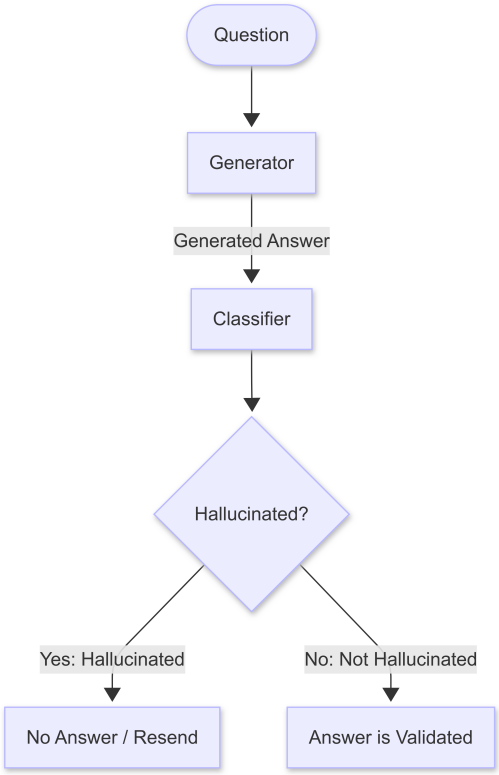

Figure 1: Overview of the dual-model pipeline: continued-pretrained generator with reasoning vector + recall-oriented classifier.

that hallucination tolerance is domain-dependent: while creative text can accept some errors, engineering and scientific domains cannot. By framing hallucination detection as a recall-oriented classification problem, our approach contributes to this broader movement toward risk-sensitive LLMs.

3 Method

3.1 Overview

Our framework (Figure 1) adopts a *dual-model* architecture consisting of a domain-adapted **generator** G and a recall-oriented **classifier** C. Given a question q, the generator proposes an answer $\hat{a} = G(q)$. The classifier then predicts $y = C(q, \hat{a}) \in \{0, 1\}$, where 1 indicates a factual response and 0 a hallucination:

$$\text{out}(q) = \begin{cases} \hat{a}, & C(q, \hat{a}) = 1 \\ \textsc{Refuse}, & \text{otherwise.} \end{cases}$$

This enforces a safety-first policy without retrieval, minimizing latency and deployment complexity. Unlike retrieval-augmented generation (RAG), which adds evidence but increases infrastructure cost, our design maintains simplicity by relying entirely on internal knowledge and a strong factuality gate.

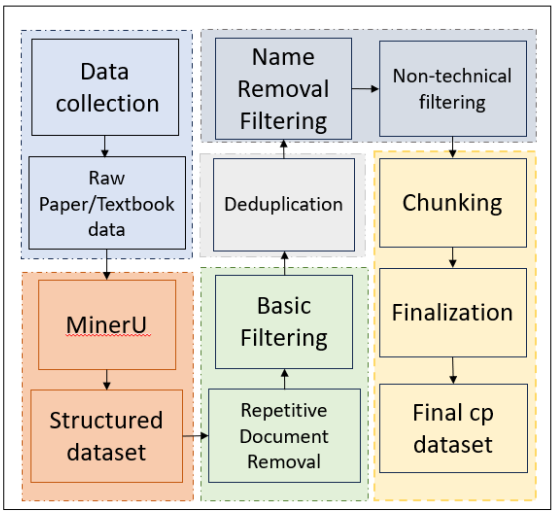

Figure 2: Preprocessing pipeline for the continued-pretraining corpus, including document structuring, filtering, deduplication, and chunking.

3.2 Continued Pretraining of the Generator

Corpus. This study assembled a semiconductor corpus comprising ∼44k research papers (1995–2024), 12 textbooks, and SemiWiki/industry documents. Boilerplate content (TOCs, references) was removed, and near-duplicate paragraphs were removed through standard hash-based deduplication. Low-information pages (figures-only, references-only) were discarded. The resulting text amounts to ∼200M tokens. A schematic of the preprocessing stages is shown in Figure 2.Licensing sources and release policies for all corpus components are summarized in Table 7.

Training setup. Following the pipeline in Figure 2, we continue pretraining Qwen2.5-14B with a standard causal LM objective. Training used a sequence length of 8,192, the AdamW optimizer, cosine learning-rate decay, and mixed precision (bf16). This step improved coverage of technical terminology (e.g., plasma chemistries, deposition precursors) and enhanced long-context stability, ensuring the generator retained domain-specific phrasing and data-rich descriptions. Full hyperparameter and compute details are provided in Appendix A.

3.3 Reasoning Transfer with Chat Vector

Instruction-following and reasoning were injected by merging a distilled reasoning *Chat Vector* (from DeepSeek-R1-Distill-Qwen-14B). Let θ_{cp} denote

continued-pretrained weights and θ_{cv} the reasoning delta; the merged parameters are:

$$\theta_G = \theta_{\text{cp}} + \lambda \cdot \theta_{\text{cv}}, \quad \lambda = 0.7.$$

This lightweight interpolation equips the generator with stepwise explanation and alignment capabilities without full supervised fine-tuning, while preserving CP knowledge when λ is modest. In practice, λ values between 0.6–0.8 gave stable behavior; below 0.5 the reasoning ability weakened, while higher values risked overwriting CP knowledge. We selected $\lambda = 0.7$ as a balanced point that preserved domain-specific terminology while retaining reasoning strength, and found it consistently stable across validation runs.

3.4 Classifier Dataset Construction

We created ∼8k QA triplets tailored to hallucination detection:

1. **Seed extraction:** atomic factual statements were extracted from CP passages and expert-provided seeds. Example: "Adding O_2 reduces polymer and improves Si/SiO_2 selectivity."
2. **Question expansion:** each fact was paraphrased into 2–3 natural questions, varying surface form and specificity.
3. **Answer pairing:** each question was paired with (i) a factual short answer and (ii) a *realistic but wrong* answer, created by perturbing details such as gas ratios, selectivity directions, step ordering, or numeric ranges (see Appendix C for the full generation template).

This approach ensures the classifier is trained on subtle errors that mirror real failure modes in semiconductor reasoning. Compared to naive negative sampling, these perturbations are realistic enough to fool a generator but wrong enough to be unsafe in practice.

3.5 Classifier Model and Training

The classifier shares the same backbone as the generator (Qwen2.5-14B + CP + CV). Each input pair (q, a) is concatenated, the EOS hidden state pooled, and passed through a linear head $h(\cdot)$ to produce a logit z. Training minimizes:

$$\mathcal{L} = \text{BCEWithLogits}(z, y),$$

with class-balanced batches and mild label smoothing (0.05) for stability. Training converged in ∼160 GPU-hours on 8×H100 with batch size 128.

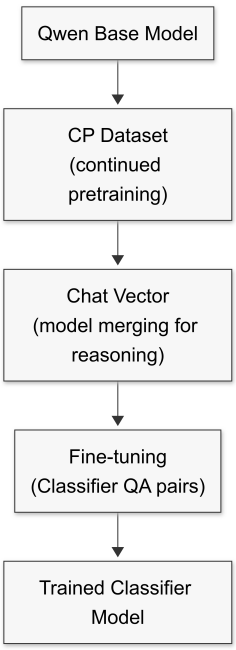

Figure 3: Classifier dataset construction: factual answers paired with realistic but incorrect hallucinations.

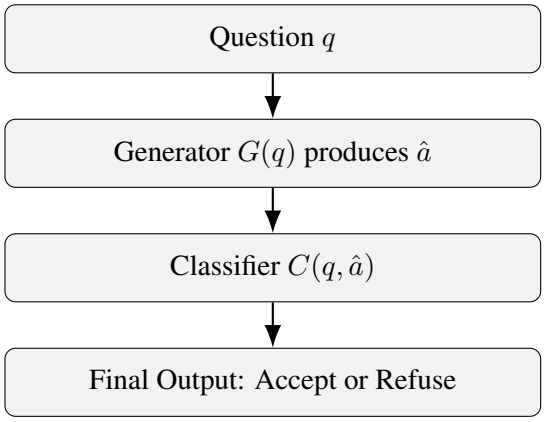

Figure 4: Evaluation flow: single-pass generation gated by a recall-oriented classifier.

3.6 Evaluation Flow

To clarify how the generator and classifier interact, Figure 4 illustrates the vertical flow: a question is answered by G, filtered by C, and either accepted or refused.

3.7 Inference Policy

At inference time, the system is single-pass by default, adding only 200–350 ms latency. Optionally, if C rejects an output, we allow up to three generator retries with stricter prompts (e.g., numeric tolerance hints, "cite-from-question" reminders). In ablations, these retries occasionally recovered borderline cases but increased latency by 2–3×.

Results reported in Section 4 use the single-pass setting.

3.8 Practical Considerations

We enforce standardized prompt templates for G and C.The classifier runs with temperature 0 and a short max-token budget to reduce variance. A known failure mode is *over-filtering* rare-but-correct answers; to mitigate this, we add a small *uncertainty band* near the decision boundary: when the classifier's confidence is close to indeterminate, we trigger a brief clarification prompt rather than issuing a hard rejection.

Deployment note: Because the classifier is lightweight, it can be colocated with the generator in a serving pipeline. Batching 8–16 queries amortizes latency spikes, making the system practical for interactive use.

4 Experiments

4.1 Evaluation Benchmark

Unless otherwise noted, all quantitative results are reported on the 1,000-QA benchmark.

Primary: 1,000-QA benchmark. A balanced evaluation set of 1,000 questions covering etching, deposition, and lithography topics. Each question has a concise gold reference, and accuracy is evaluated in a binary manner (factual vs. non-factual).

Diagnostic: expert sub-fact set. A small, expert-curated subset (12 questions decomposed into 2–6 atomic claims each) used only for fine-grained and human-agreement analyses. This set is diagnostic only and does not contribute to reported aggregate scores.

Training data note. The classifier fine-tuning corpus described in Section 3.4 was used solely for training; no items from it appear in evaluation benchmarks.

4.2 Experimental Setup

Splits and leakage control. All datasets are stratified into train/dev/test (80/10/10) with balancing by subdomain and question type (numeric vs. conceptual). To avoid information leakage from continued pretraining (CP), evaluation documents were excluded from the CP crawl by DOI/URL hashing, and near-duplicate paragraphs were removed at the paragraph level through standard hash-based deduplication (see Figure 2).

Prompting and decoding. All generators use a shared instruction template (Appendix C). Generation employs greedy or nucleus (top-p) decoding depending on the baseline, with a maximum output length of 128 tokens. Classification runs deterministically (T=0), processing up to 2,048 input tokens (question + answer) and generating short outputs of 16–32 tokens. Unless otherwise noted, no retrieval augmentation is used.

Evaluation protocol. For the 1,000-QA benchmark, factual accuracy is scored by GPT-4o acting as an LLM-as-a-judge: an output is counted as correct only if *all* factual claims match the gold answer. The benchmark itself is derived from 12 expert-written seed questions, expanded into 1,000 diverse QA pairs using GPT and Gemini. While the seed set provides human grounding, large-scale scoring is conducted entirely by GPT-4o in a binary factuality setting (see Appendix C for the full evaluation prompt).A subset of 100 QA items was additionally human-verified to assess alignment between GPT-4o and expert judgments (Section 4.7).

4.3 Baselines and Implementation

We compare our approach against four representative models: **Qwen2.5-14B Instruct** (open instruction-tuned baseline), **DeepSeek-R1-Distill-Qwen-14B** (reasoning-oriented distillation), **Semikong-8B/70B** (semiconductor-pretrained models without hallucination filtering), and **GPT-4o** (a commercial frontier model). Each generator is evaluated both standalone and when paired with our classifier.

Inference is conducted using `vLLM` (v0.9.2) on 8× V100 (32 GB) GPUs with fp16/bfloat16 precision and temperature 0.0 for deterministic decoding. The maximum context length is 16,384 tokens; output lengths are capped at 8,192 tokens for Qwen-based models and 4,096 for LLaMA-based ones (e.g., Semikong). Classifier passes are lightweight and batched to minimize latency variance. Each pass processes up to 2,048 input tokens (question + answer) and generates a short binary output ($\leq$32 tokens).

4.4 Main Results and Ablations

Table 1 isolates the contributions of CP, CV, and CLF. CP provides the largest single improvement (+6.6 points) by grounding terminology; CV injects reasoning (+7.1 points); and CLF delivers the final accuracy lift (+5.1 points). The effects are comple-

Variant	CP	CV	CLF	Acc. (%)
Base Qwen2.5-14B	–	–	–	68.3
+ CP	✓	–	–	74.9
+ CP + CV	✓	✓	–	82.0
+ CP + CV + CLF (Ours)	✓	✓	✓	**87.13**

Table 1: Ablation on 1,000 QA pairs.

Model	Before	After	Lat. (ms)
Ours (Dual-Model)	82.00	**87.13**	249.6
GPT-4o	77.40	85.71	264.8
Qwen2.5-14B Instruct	68.30	75.64	348.6
DeepSeek-R1-Distill-Qwen-14B	70.00	74.24	226.4
Semikong-8B	52.30	59.42	237.5
Semikong-70B	49.40	58.53	307.6

Table 2: Pipeline binary accuracy (%) before/after classifier; added latency.

mentary: performance degrades substantially if any component is removed.

Table 2 reports end-to-end pipeline accuracy and latency. Our dual-model system improves from 82.0% to 87.13% accuracy with only 249.6 ms added latency. Importantly, all baselines also gain when paired with the classifier, confirming its generality across architectures. For example, GPT-4o improves +8.3 points and Semikong-70B +9.1 points, but both remain below our 14B pipeline despite their scale.

Analysis. These results highlight that scale alone is insufficient for safety in semiconductor QA. Even the 70B Semikong model remains far below our smaller 14B dual-model when evaluated under the same binary criteria. The gains from classifier gating are consistent across families, underscoring that factual verification is a model-agnostic requirement. Latency overhead (200–350 ms) is negligible compared to human decision cycles in semiconductor engineering, where process design steps typically span minutes to hours.

The classifier also generalized well across architectures: despite being trained only on Qwen2.5 outputs, it consistently improved models as diverse as GPT-4o and Semikong-70B. This suggests that the decision boundary it learns is not model-specific but rather tied to domain-appropriate factual grounding. Taken together, these findings highlight that continued pretraining and reasoning transfer provide coverage and fluency, while classifier gating supplies the final layer of factual safety.

Setting	Value
CP sequence length	8192
CP optimizer / LR	AdamW / 2×10^{-5} (cosine)
Precision	bf16
CV merge λ	**0.7**
Classifier max tokens	16–32 (temp 0)
Retries	0 (single-pass reported)
Hardware	8×H100 80GB

Table 3: Key hyperparameters for CP, CV merge, and classifier inference.

Model	TP	FP	TN	FN	Acc.	Rec.	FPR
Ours (Dual)	85	177	643	95	72.8	47.2	21.6
DeepSeek-R1	104	135	565	196	66.9	34.7	19.3
GPT-4o	144	282	492	82	63.6	63.7	36.4
Qwen2.5 Inst.	175	242	441	142	61.6	55.2	35.4
Semikong-8B	210	132	391	267	60.1	44.0	25.2
Semikong-70B	307	199	325	169	63.2	64.5	38.0

Table 4: Classifier metrics on held-out judged answers. Counts correspond to ∼1,000 evaluated answers. The same trained classifier is applied across all generators.

4.5 Classifier Effectiveness and Error Analysis

Table 3 summarizes key hyperparameters for CP, CV, and the classifier. The classifier was calibrated for recall-first behavior, prioritizing the rejection of unsafe answers even at the cost of additional false positives.

All results in Table 4 use the same trained classifier (Qwen2.5-14B + CP + CV, fine-tuned on our hallucination dataset), applied unchanged across all generators. This design demonstrates that the classifier generalizes beyond its training distribution and is not specialized to a single model.

The recall-first operating point leads to a relatively high false-positive rate (e.g., 21.6% for our dual system), but this is intentional: it ensures unsafe answers are consistently blocked, even at the expense of rejecting some safe ones. This tradeoff is particularly important in semiconductor contexts where the cost of allowing an unsafe answer far outweighs the inconvenience of an extra rejection.

Error modes. Residual errors fall into three main categories: (1) *Semantic Reversals*, where mechanisms are flipped (e.g., whether O_2 suppresses or increases polymerization); (2) *Partially Incorrect ranges*, where numeric values are shifted outside acceptable bounds; and (3) *Rare-but-Correct outputs*, where the answer is factually valid but conservatively rejected. These rare-but-correct cases can be further divided into two subtypes: (i) *Numeric variants*, where the model provides values within

Type	Example (abridged)
Semantic Reversal	*Q:* Does adding O_2 increase or suppress polymer in HBr+O_2 etch? *Gen:* "Increases polymer." *Gold:* "Suppresses polymer." *Classifier:* Reject.
Partially Incorrect	*Q:* Typical chamber pressure for step X? *Gen:* "40–60 mTorr." *Gold:* "30–50 mTorr." *Classifier:* Reject (out-of-band).
RBC-N	*Q:* Selectivity when adding gas Y? *Gen:* "Si:SiO_2 $\approx$ 7:1." *Gold:* "5–7:1." *Classifier:* (FP) Reject.
RBC-C	*Q:* How can LER be reduced in EUV resists? *Gen:* "Use a double-layer resist system." *Gold:* "Increase absorption and reduce acid diffusion." *Classifier:* (FP) Reject (alt. valid mechanism).

Table 5: Representative classifier outcomes. RBC = Rare-but-Correct; N = numeric variant, C = conceptual variant.

a plausible range but slightly deviating from the reference (e.g., reporting 510 nm/min instead of 500 nm/min), and (ii) *Conceptual variants*, where the model offers an alternative yet valid mechanistic explanation (e.g., describing a double-layer resist structure instead of reduced acid diffusion for LER mitigation). Representative examples are shown in Table 5.

Overall, because the classifier parameters are fixed, its consistent improvements across diverse generators (including GPT-4o and Semikong) suggest it captures domain-grounded factual cues rather than overfitting to a single model's style.

4.6 Latency and Throughput

The classifier adds 200–350 ms per query (median $\approx$250 ms, p90 in appendix). Because C is a short forward pass with small max tokens, batching 8–16 queries amortizes the cost. In practice, this overhead is negligible compared to the time costs of wafer processing, yet prevents costly unsafe guidance.

4.7 Human–LLM Agreement Validation

To assess the reliability of GPT-4o as the automatic factuality judge, we sampled 100 items from the 1,000-QA benchmark and obtained independent annotations from two researchers experienced in large language model (LLM) evaluation. Each rater labeled model outputs as *factual* or *hallucinated* following the same binary guideline used by the LLM judge, focusing on factual consistency rather than domain-specific technical accuracy.

Comparison	Cohen's κ	Gwet's AC1
Human A vs Human B	0.64	0.82
Human A vs GPT-4o	0.71	0.85
Human B vs GPT-4o	0.79	0.92
Human majority vs GPT-4o	0.92	0.97

Table 6: Inter-judge agreement on a 100-QA subset. Values show agreement between two human factuality raters and the GPT-4o factuality judge.

Table 6 summarizes inter-judge agreement. Inter-annotator consistency between the two human raters reached Cohen's $\kappa = 0.64$ and Gwet's AC1 $= 0.82$, indicating substantial agreement. Pairwise agreement between GPT-4o and each human annotator yielded $\kappa = 0.71$ / 0.79 (AC1 = 0.85 / 0.92), and comparing the human majority vote with GPT-4o's labels produced $\kappa = 0.92$, AC1 $= 0.97$. Disagreements primarily involved borderline numeric tolerances (±5%) or alternative yet valid mechanism descriptions.

These results confirm that GPT-4o's automatic judgments are highly consistent with human factuality assessments, validating its use as an efficient large-scale factuality judge while emphasizing the value of multi-rater verification for high-precision subsets.

5 Reproducibility

All models were trained with `LLaMA-Factory` (v0.9.1.dev0) and evaluated with `vLLM` (v0.9.2). Experiments ran on 8×H100 80GB GPUs. Continued pretraining used a context length of 8,192, the AdamW optimizer with cosine decay, a learning rate of 2×10^{-5}, and bf16 precision. Classifier training used fixed random seeds (42 for CP and 123 for the classifier), temperature 0, and a maximum output length of 16–32 tokens.

To support reproducibility, the 1,000-QA benchmark, preprocessing scripts, prompt templates, and code for reproducing Tables 2, 1, and 4 will be released upon publication.

6 Conclusion

This study presented a dual-model pipeline designed to mitigate hallucinations in semiconductor QA. The framework integrates three complementary components: continued pretraining on a 200M-token domain corpus, reasoning transfer via

a Chat Vector, and a recall-oriented classifier. Together, these yield a significant accuracy gain on a 1,000-QA benchmark, raising binary factual accuracy from 82.0% to 87.13% with only 200–350 ms latency overhead. The improvements are consistent across baseline generators, confirming the classifier's generality, and analysis shows that it suppresses common failure modes such as semantic reversals and numeric drift—errors that in practice can translate into costly wafer losses. Even modest accuracy gains (+5.1% over CP+CV) rescue roughly 50 otherwise hallucinated answers per 1,000 queries, illustrating practical value. By combining a recall-oriented classifier in a single-pass pipeline, the framework strikes a strong balance between factual reliability, latency, and deployment feasibility.

Limitations

This study has several limitations. First, the evaluation benchmark covers only etching, deposition, and lithography, leaving packaging, metrology, and design automation for future work. Second, the recall-first classifier can over-filter rare but correct answers—a deliberate trade-off suited to high-stakes applications but less ideal for exploratory use. Third, the classifier depends on a curated hallucination dataset; extending it to other domains will require additional expert input. Finally, evaluation relies on LLM-as-a-judge scoring for the 1,000-QA benchmark. Although based on expert-authored seeds, its large-scale expansion and automated scoring may introduce bias relative to full human annotation.

Ethical Considerations and Risks

This work aims to improve factual reliability of LLMs in semiconductor QA, where incorrect process guidance can cause serious material or financial losses. Recall-oriented filtering may over-reject valid answers, but in this domain safety outweighs coverage. Future work may quantify this trade-off in terms of misguidance versus abstention cost.

All data were drawn from public academic and industrial sources. Proprietary materials (e.g., SemiWiki) were used only for pretraining and will not be redistributed. Table 7 summarizes licensing and release policies for all CP corpus components. No personal or user-generated data were included. Upon publication, we will release all legally shareable datasets, prompts, and scripts to ensure reproducibility, consistent with applicable copyright and license terms.

Source Type	Example Sources	License / Access	Usage & Release Policy
Academic papers	IEEE, ACM, arXiv (1995–2024)	Publisher copyright (research use)	Used under institutional subscriptions and fair use for non-commercial research. Scripts and metadata only; no full-text redistribution.
Textbooks & references	University / industry textbooks	Copyrighted (educational fair use)	Used as short excerpts for continued pretraining; no redistribution of full texts. We release token counts and preprocessing code only.
Industry / SemiWiki	SemiWiki posts, vendor application notes	Mixed public / proprietary	Used solely for pretraining to improve terminology; raw text will **not** be shared. We will release only legally shareable artifacts.
Open-access / public docs	arXiv, NIST, open technical reports	CC-BY or public domain	Legally shareable; these portions and preprocessing scripts will be made available to the community or upon request to support reproducibility.

Table 7: Licensing and release policy for corpora used in continued pretraining.

The system is intended strictly for research use. In deployment, human-in-the-loop verification remains essential to avoid overreliance on automated outputs.

Future Work

Future extensions will focus on strengthening factual evaluation and grounding. One direction is to incorporate retrieval-augmented generation (RAG) using semiconductor papers and textbooks, providing explicit evidence that may further reduce hallucinations and classifier load. Expanding the dataset to additional process domains (e.g., packaging, metrology, CAD/EDA) and multilingual corpora would enhance coverage and cross-domain generality.

On the modeling side, adaptive thresholding based on question type (numeric vs. conceptual), uncertainty calibration, and self-reflection loops—where the generator adapts using classifier feedback—represent promising directions for improving robustness. Finally, human-in-the-loop verification and lightweight deployment optimizations (e.g., quantization or pruning) remain key to safe industrial integration.

Overall, this work demonstrates that domain-adaptive pretraining, reasoning transfer, and classifier gating are complementary approaches for improving factual reliability in semiconductor LLMs, establishing a foundation for future domain-safe AI assistants.

Acknowledgments

We acknowledge the use of OpenAI's ChatGPT for assistance in refining the writing of this paper. ChatGPT was used for editing, restructuring, and improving clarity in several sections. All technical content, experimental design, data collection, and analysis were carried out entirely by the authors.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *Preprint*, arXiv:1908.10063.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Michael Bommarito and Daniel Katz. 2022. Gpt-3 and the future of legal services. *Law, Technology and Humans*, 4(1).

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. Tldr: Extreme summarization of scientific documents. In *Findings of ACL: EMNLP 2020*, pages 4766–4777.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904.

DeepSeek-AI Team. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. Rarr: Researching and revising what language models say, using language models. *Preprint*, arXiv:2210.08726.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Tanmay Gupta and 1 others. 2022. Chemberta-2: Towards chemical foundation models. In *NeurIPS Workshop on AI for Science*.

Nils Holzenberger and 1 others. 2023. Legal reasoning in large language models. In *NeurIPS 2023 Workshop on AI for Law*.

Edward J. Hu and 1 others. 2021. Lora: Low-rank adaptation of large language models. In *ICLR 2021*.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tzong-Han Tsai, and Hung-yi Lee. 2024. Chat vector: A simple approach to equip llms with instruction following and model alignment in new languages. *Preprint*, arXiv:2310.04799.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Luis Jimenez-Gutierrez and 1 others. 2022. Biomedical question answering: A survey of datasets and methods. In *ACL 2022*.

Saurav Kadavath and 1 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Pavel Kuksa, Pasquale Minervini, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Renqian Luo, Liner Sun, Yixuan Xia, Tao Qin, Sheng Zhang, and Tie-Yan Liu. 2022. Biogpt: Generative pre-trained transformer for biomedical text generation and mining. *Preprint*, arXiv:2210.10341.

Lucia Magister and 1 others. 2022. Teaching arithmetic to small transformers. In *NeurIPS 2022*.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *Preprint*, arXiv:2303.08896.

Riccardo Miotto, Marco Danieletto, and 1 others. 2023. Large language models in medicine. *Nature Medicine*.

Niklas Muennighoff and 1 others. 2022. Crosslingual generalization through multilingual fine-tuning. In *ACL 2022*.

Christopher Nguyen, William Nguyen, Atsushi Suzuki, Daisuke Oku, Hong An Phan, Sang Dinh, Zooey Nguyen, Anh Ha, Shruti Raghavan, Huy Vo, Thang Nguyen, Lan Nguyen, and Yoshikuni Hirayama. 2024. Semikong: Curating, training, and evaluating a semiconductor industry-specific large language model. *Preprint*, arXiv:2411.13802.

P. Rao and 1 others. 2023. Industrial ai: Applications of large language models in engineering. *IEEE Transactions on Industrial Informatics*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Vahe Tshitoyan and 1 others. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, 571(7763):95–98.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Component	Hyperparameters
CP	AdamW, LR 1×10^{-6} (cosine w/ warm-up), seq len 8,192, batch 1, grad accum 8, 5 epochs, `bf16`.
CV	Merge weight $\lambda = 0.7$, no post-merge fine-tuning.
Classifier	AdamW, LR 9×10^{-7}, 5 epochs, batch 1, binary cross-entropy loss.

Table 8: Key hyperparameter settings for CP, CV, and CLF (aligned with thesis).

A Experimental Setup

Experiments used vLLM v0.9.2, seeds 42 (pretraining) and 123 (classifier), temperature 0.0, max tokens 32, and `bf16` precision. The environment used 8×NVIDIA H100 (80 GB) GPUs with PyTorch 2.2.1 and CUDA 12.1.

Continued pretraining (CP). Continued pretraining consumed approximately 200 M tokens on 8×H100 (80 GB) over ∼20 wall-clock hours (≈160 GPU-hours). Training used AdamW (learning rate 1×10^{-6} with cosine decay and warm-up), batch size 1 with gradient accumulation 8, and maximum sequence length 8192. Sequence packing was applied, and documents exceeding this limit were truncated.

Chat Vector merging (CV). Merged the reasoning vector from DeepSeek-R1-Distill-Qwen-14B into the domain-pretrained model with merge weight $\lambda = 0.7$. No post-merge fine-tuning was performed.

Classifier fine-tuning (CLF). Fine-tuned the classifier using AdamW (learning rate 9×10^{-7}), batch size 1, and 5 epochs with binary cross-entropy loss. Training consumed approximately 20 GPU-hours in total (∼2.5 h wall-clock).

B QA Dataset Examples

Each sample in the classifier dataset consists of one factual and one subtly hallucinated answer. Hallucinations typically modify numeric values, material types, or process parameters. Examples are in Table 9

Question	Factual Answer	Hallucinated Answer
What method models SerDes transceiver analog behavior?	IBIS-AMI behavioral modeling captures equalization, gain, and clock recovery.	IBIS-AMI modeling adds symbol-based recovery not present in the reference.
How to reduce LER in EUV resists?	Use resists with higher absorption and lower acid diffusion.	Employ a double-layer resist system with different top/bottom sensitivities.
How is chip reliability monitored in AI/data-center chips?	Use on-chip PVT monitors for real-time voltage and thermal adjustment.	Use DTVT sensors to dynamically tune voltage–frequency offsets.

Table 9: Examples of factual vs. subtly hallucinated QA pairs.

C Prompt Templates

LLM-as-a-Judge prompt

```
You are an expert evaluator. Given a key fact (ground truth) and a model-generated answer, your task is to judge whether the answer reasonably includes, implies, or supports the key fact.
Respond ONLY with "Yes" or "No": - "Yes" if the model's answer clearly includes, paraphrases, implies, or provides sufficient context for the key fact. - "No" if the key fact is clearly missing or contradicted.
Be fair and realistic — reward meaningful understanding, not exact matches.
```

Inference prompt

```
You are a senior semiconductor process engineer specializing in plasma etching, wet etching, and dry etching at sub-10nm nodes.
Your role is to assist fab process engineers and researchers in root cause analysis, recipe tuning, and etch mechanism understanding.
Always provide highly technical, mechanism-driven, and process-specific explanations. Prioritize clarity, precision, and relevance to advanced node fabrication.
Where appropriate, include concrete parameters (e.g., 10 mTorr pressure, 800 W ICP, 5% O2 in C4F8 feed gas) and explain their effects.
Address plasma control issues such as magnetic field configurations (e.g., static, rotating, pulsed) and their influence on ion trajectories, radical transport, and etch profiles.
Use insights from ALE, pulsed plasma, and emerging etch chemistries. Frame answers as if troubleshooting process integration problems or mentoring junior engineers in a fab environment.
```

Hallucination-generation prompt

```
You are an academic assistant whose task is to generate question-answer pairs from academic papers, including reasoning traces. Instructions: 1. Read the provided paper content. 2. Convert any LaTeX expressions into natural, readable scientific language. 3. Select a concise, meaningful phrase containing key technical information. 4. Write a clear, specific question directly answered by this phrase. 5. Generate: - Question - Answer (direct quote, LaTeX converted) - CoT (correct reasoning) - HallucinatedAnswer (factually incorrect but plausible) - HallucinatedCoT (reasoning leading to the hallucinated answer) Hallucinations must be subtle and realistic (e.g., incorrect parameter values, altered gas ratios, or mis-specified process steps).
```

Classifier prompt

```
You are a hallucination detector for semiconductor Q&A. Only flag clear contradictions or unjustified, highly specific details. If unsure or the answer is generic and plausible, do not flag.
Return ONLY 0 or 1.
Mark 1 ONLY if:

  • The answer clearly contradicts the question, OR

  • It gives a highly specific technical detail (exact numbers, gas/tool names, or step order) that is not justified by the question.

Otherwise return 0.

Question: {question}
Answer: {model_answer}
```

D Inference setup.

All inference experiments were conducted with vLLM v0.9.2 using `bf16` precision and maximum sequence length 8,192. For generation tasks, temperature was set to 0.0 and top-p to 1.0 (greedy decoding). Each query was processed in a single forward pass without retrieval augmentation. Latency was measured on a single NVIDIA H100 (80 GB) GPU, averaged over the 1,000-QA benchmark with batch size 1. Classifier inference shared the same setup and added an average of 200–350 ms per QA pair.

E Latency Breakdown

Classifier and end-to-end latencies were measured across the full 1,000-QA benchmark. Table 10 reports the median (p50) and 90th-percentile (p90) latencies for both the classifier alone and the complete generator–classifier pipeline.

Metric	p50 (ms)	p90 (ms)
Classifier only	201.2	321.4
Full pipeline (gen+clf)	501.2	614.2

Table 10: Latency distribution for the generator–classifier pipeline.

Question	Without Retrieval	With Retrieval (RAG)
RF power adjustment for Si/SiO_2 selectivity	*"Lower RF power improves selectivity."* (incorrect)	*"Maintain 600 W RF power as specified in experiment."* (correct)
O_2 effect on polymer formation	*"Increases polymer."* (incorrect)	*"Suppresses polymerization in HBr+O_2 chemistry."* (correct)
Etch gas ratio optimization	*"Add more CHF_3 to improve selectivity."* (incorrect)	*"Increase HBr concentration to enhance selectivity; O_2 adjusts sidewall control."* (correct)

Table 11: Effect of retrieval context (RAG) on factual accuracy for three representative seed questions.

F Preliminary RAG Diagnostic

To examine whether retrieval-augmented generation (RAG) can further complement the dual-model pipeline, we ran a small diagnostic on the 12 expert seed questions. Each question was paired with a short, lexically relevant passage from the CP corpus (one paper paragraph or textbook excerpt) appended as context.

Rather than building a full RAG system, this experiment tested whether explicit evidence reduces hallucination frequency. As shown in Table 11, retrieved context corrected several mechanism and numeric misinterpretations, suggesting that retrieval grounding can synergize with classifier gating.

Overall, retrieval and classification appear complementary: retrieval enhances factual grounding *before* generation, while the classifier enforces reliability *after* generation. This tri-stage structure (RAG → Generator → Classifier) merits further study for balancing accuracy and latency.