LATTICESPIKE: REAL-TIME TOKEN-LEVEL HALLUCINATION DETECTION VIA BOX EMBEDDINGS AND SPIKING DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models hallucinate mid-generation, yet most detectors only judge text after generation is complete. To address this gap, we propose LatticeSpike, a biologically inspired lightweight detector that runs alongside a frozen language model and scores every token as it is produced, using only the tokens generated so far. At each step, signals read from the language model are embedded as a box whose volume reflects the model's semantic spread. A recurrent spiking neural network, with dynamics modulated by that volume, continuously predicts its own next state. Hallucinated spans break this rhythm and the resulting prediction error, trained against token labels with a ranking objective, is the detection score. Furthermore, we release two human-validated and statistically tested benchmark datasets, spanning two distinct language model families, comprising 10,915 and 10,919 examples with token-level labels from a six-judge LM panel; we call these datasets THRIVE-Llama and THRIVE-Gemma, respectively. We evaluate LatticeSpike against twelve baselines, on four corpora: THRIVE-Llama, THRIVE-Gemma, RAGTruth, and TriviaQA, under a strictly causal protocol. Notably, across the four corpora and multiple generator families, LatticeSpike outperforms all tested baselines in 96/101 individual evaluation cells ($\sim$95% win rate) spanning average precision (AP) and AUROC, as well as recall, precision, and F1 score at variable false-positive rates (0.1%–5%), under full-corpus and cross-task evaluation protocols. We also provide extensive ablation experiments showing which components are most important to LatticeSpike's success, and statistical testing to show where LatticeSpike's advantage is significant and where it should not be overinterpreted. The code for the project and the released benchmarks can be found at: `https://anonymous.4open.science/r/LatticeSpike-8EB2`.

1 INTRODUCTION

Language models (LM) exhibit a well-documented failure mode: they generate fluent, confident text that is factually incorrect or semantically inconsistent with their own context (Maynez et al., 2020; Ji et al., 2023). This flaw, known as hallucination, can be problematic in any deployed setting — eroding user trust and propagating errors into any decision or downstream system that consumes the model's output unchecked.

A rich line of work detects hallucination after generation, by probing hidden states (Azaria & Mitchell, 2023), reading attention patterns (Chuang et al., 2024), or scoring model uncertainty (Sriramanan et al., 2024). However, an open problem remains: detecting hallucination in real-time, during generation.

To this end, we propose LatticeSpike, a biologically inspired, real-time, lightweight hallucination detector that runs alongside a frozen LM and scores every token as it is produced using only the tokens generated so far. At each step, signals read from the generator (language model) are embedded as a box whose volume reflects the model's semantic spread. A recurrent spiking neural network, whose dynamics are modulated by the semantic spread volume, continuously predicts its own next state, and the prediction error — the degree to which generation has broken the network's rhythm — is the detection score.

We additionally release THRIVE-Llama (THRIVE-L) and THRIVE-Gemma (THRIVE-G), two token-level hallucination benchmarks, identical in construction, which differ only in the generator producing their respective responses, Llama-3.2-1B and Gemma-2-2B, and their response count, 10,915 for THRIVE-L and 10,919 for THRIVE-G. Across both variants, responses are labeled by a six-judge language model panel, and are human-validated and statistically tested.

Against twelve baselines (a hidden-state probe, SAPLMA (Azaria & Mitchell, 2023), SAPLMA-tok, HaMI (Niu et al., 2025), Lookback Lens (Chuang et al., 2024), the ICR Probe (Zhang et al., 2025), ICR-tok, and the five LLM-Check scores (Sriramanan et al., 2024)), spanning four corpora (THRIVE-L, THRIVE-G, RAGTruth (Niu et al., 2024),

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

TriviaQA (Joshi et al., 2017)) LatticeSpike leads every token-level metric — average precision (AP), AUROC, and recall, precision, and F1 at 1% false-positive rate — and transfers best across RAGTruth's generation tasks among the tested methods, winning 42/45 unique evaluation cells. It adds 3.6–6.3 ms per token and at most 103 MiB of memory to streaming generation, with its recurrent core running faster on CPU than on GPU, therefore making it feasible for deployment on resource-constrained hardware alongside the very model it monitors.

Contributions. We make the following contributions. (1) We show that a biologically inspired architecture comprising box embeddings and spiking dynamics represents a viable and effective way to detect language model hallucinations in real-time. (2) We introduce LatticeSpike, a lightweight and edge-deployable detector of the aforementioned type that outperforms tested baselines across $96/101$ ($\sim$95%) unique evaluation cells, spanning multiple corpora, evaluation protocols, and generator families. (3) We conduct several ablation studies to show what drives LatticeSpike's performance — helping inform future work in this domain. (4) We conduct statistical testing on LatticeSpike's results to show exactly where its performance is significant and where it should not be overinterpreted. (5) We release THRIVE-Llama and THRIVE-Gemma, two human-validated and statistically tested token-level hallucination benchmarks.

2 Related Work

Embeddings and Lattice-Structured Representations. Geometric representation learning provides a natural mechanism for encoding asymmetric and transitive relations such as entailment, subsumption, and hierarchical concept structure. Box embeddings represent concepts as axis-aligned hyperrectangles in $\mathbb{R}^d$, where geometric containment can encode partial-order relations and intersection can express conjunction between concepts. Vilnis et al. (Vilnis et al., 2018) introduced probabilistic box embeddings for knowledge graphs, combining a box lattice with a probability measure that supports joint and conditional queries while allowing overlapping, disjoint, and negatively correlated concepts. Dasgupta et al. (Dasgupta et al., 2020) subsequently addressed optimization and local-identifiability issues in probabilistic box embeddings by modeling box boundaries with min- and max-Gumbel distributions. This formulation preserves closure under intersection while providing a differentiable mechanism for learning structured regions. Collectively, these works establish box geometry as a representation for probabilistic set inclusion, intersection, and lattice like semantic organization.

Galois Connections and Concept Lattices. Galois connections provide an order theoretic mechanism for relating two partially ordered domains through paired adjoint mappings. They form a central mathematical component of Formal Concept Analysis (FCA), where derivation operators between sets of objects and attributes induce closure operators whose fixed points form a concept lattice (Ganter & Wille, 1999). The resulting lattice organizes concepts according to generalization and specialization and provides explicit meet and join operations over conceptual structure. FCA and concept lattices have subsequently been applied to knowledge representation, linguistic analysis, and text retrieval. Related algebraic approaches to language have considered other forms of structured semantics; for example, Coecke et al. (Clark et al., 2011) combine distributional vector representations with pregroup grammar through a categorical framework for compositional meaning. Although mathematically distinct from FCA-style Galois connections, such work illustrates the broader use of algebraic structure to constrain and organize distributed semantic representations.

Internal Representations for Hallucination Detection. A growing line of work investigates whether hallucinations can be detected directly from the internal representations of language models rather than solely from generated text. More recent methods examine structured aspects of these internal computations. Lookback Lens (Chuang et al., 2024) focuses on contextual hallucination and constructs *lookback-ratio* features that measure, for individual attention heads, how strongly generated tokens attend to the supplied context relative to previously generated tokens. ICR Probe (Zhang et al., 2025) instead models the evolution of representations across transformer layers. It introduces the Information Contribution to Residual Stream (ICR) score to characterize how individual modules contribute to hidden-state updates and uses these signals to represent cross-layer hidden-state dynamics. HaMI (Niu et al., 2025) addresses the dependence of representation based detectors on predetermined token positions by formulating hallucination detection as multiple instance learning over token representations. This formulation jointly identifies informative tokens and aggregates them into a sequence level prediction.

Spiking Neural Networks and Temporal Anomaly Detection. Spiking neural networks (SNNs) represent information through discrete neuronal events while maintaining continuous internal state through membrane dynamics, giving them an explicit temporal computational structure (Maass, 1997; Pfeiffer & Pfeil, 2018). Recurrent spiking neural networks extend this formulation with recurrent connections that allow past events to influence subsequent neuronal

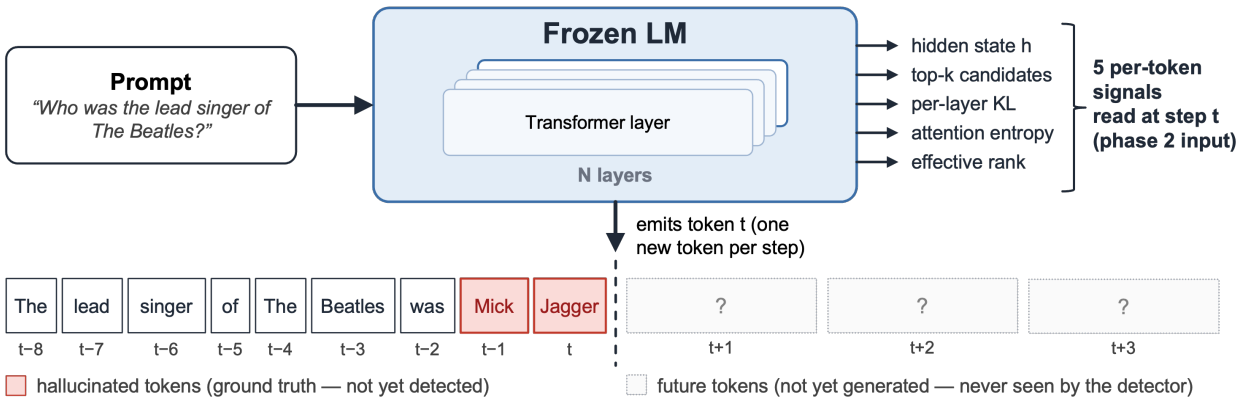


Figure 1: Phase one of the LatticeSpike pipeline.

states. A central difficulty in training SNNs is the non-differentiability of the spike generation function. Surrogate gradient methods address this issue by replacing its derivative during optimization with a smooth approximation, enabling gradient based learning and backpropagation through time (Neftci et al., 2019). These temporal properties have also motivated the application of spiking networks to anomaly detection. Maciąg et al. (Maciąg et al., 2021) formulate online unsupervised anomaly detection using evolving spiking neural networks, in which streaming observations are processed incrementally without requiring labeled anomalies. Cherdo et al. (Cherdo et al., 2023) investigate recurrent spiking architectures for time-series prediction and unsupervised anomaly detection, using recurrent neuronal dynamics to model temporal regularities and identify deviations from expected sequence behavior. Such work provides a foundation for treating anomaly detection as a problem of recognizing atypical temporal evolution rather than classifying isolated observations. More related work which detects hallucination in Language Models can be found in Section A of the Appendix.

3 Method

3.1 Problem Formulation

Given a prompt x, a response $y = (y_1, \ldots, y_T)$ generated one token at a time by a language model, and internal signals Σ_t read from the model at each step t, the goal is to produce per-token hallucination scores $s_t = f(x, y_{1:t}, \Sigma_{1:t})$, where f may use only information available at step t and may not modify the generating model. Thresholded, these scores localize hallucinated spans as they emerge, allowing generation to be flagged or interrupted before an unsupported response is completed and acted upon.

3.2 Architectural Pipeline

Phase 1 — causal signal readout. At each step t, five signals are read from the frozen language model, which together constitute Σ_t (from Section 3.1). (1) The hidden state h_t, taken halfway up the model's L-layer stack at $\ell = \lfloor L/2 \rfloor$: the depth at which semantic content is most accessible (Azaria & Mitchell, 2023). (2) The top-k next-token candidates with their probabilities (k=15): the model's shortlist of what to say next. (3) A per-layer prediction-shift trace: each layer's hidden state is projected through the LM head (logit lens (Nostalgebraist, 2020)) into its own next-token distribution p_t^ℓ, and $\mathrm{KL}(p_t^\ell \,\|\, p_{t-1}^\ell)$ measures how much layer ℓ revised its belief between consecutive steps. (4) The mean attention entropy: how focused the model's attention is on its context. (5) The effective rank (Roy & Vetterli, 2007) of the stacked layer representations: the L hidden states at step t (one per layer) are stacked into a matrix, and the entropy of its normalized singular values gives a continuous count of how many distinct directions they span — high when successive layers contribute genuinely different representations, low when the computation collapses into a few redundant ones. No gradients flow into the model, and no signal involves any position beyond t (the detector remains blind to future tokens); the signal set is passed once per generated token to Phase 2. An overview of Phase 1 is shown in Figure 1.

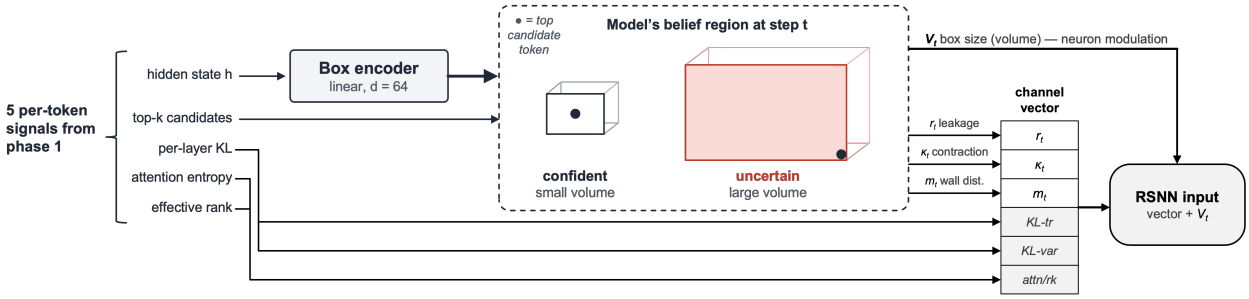


Figure 2: Phase two of the LatticeSpike pipeline.

Phase 2 — box embedding and channel construction. A learned linear encoder maps h_t (signal (1)) to a box (Vilnis et al., 2018) $B_t = [c_t^{\min}, c_t^{\max}] \subset \mathbb{R}^d$ (d=64) — an axis-aligned region representing the model's current belief. Each dimension of B_t is specified by its minimum corner and a strictly positive width $w_{t,j}$, the latter enforced through a softplus, so the maximum corner always lies above the minimum and B_t is a valid box at every step. Furthermore, the box's normalized volume $V_t = \left(\prod_{j=1}^{d} w_{t,j}\right)^{1/d}$ measures the model's semantic spread at step t: a confident step yields a tight box, an uncertain one yields an inflated box.

Next, three readouts summarize the box's geometry: the leakage residual r_t, how far B_t escapes the previous step's box; the contraction rate κ_t, how quickly the belief is narrowing; and the concretization residual m_t, the probability-weighted distance from B_t to the embeddings of the candidate tokens of signal (2), projected into box space through the model's own embedding table.

These readouts join the three remaining, pass-through signals in a six-channel vector $\mathbf{x}_t$: the prediction-shift trace enters as a compressed summary with its across-layer variance; and attention entropy pairs with effective rank in a shared channel. Concretely, Phase 2's output is $\mathbf{x}_t$ and the scalar V_t, each routed to Phase 3 independently. Phase 2 is illustrated in Figure 2.

Phase 3 — volume-modulated spiking dynamics and scoring. A recurrent spiking neural network receives two independent streams, both originating from Phase 2's output: the channel vector $\mathbf{x}_t$, responsible for driving the network's activity, and the box size (volume) V_t, responsible for modulating the network's dynamics. The network comprises 64 leaky-integrate-and-fire neurons (Maass, 1997) with recurrent connections, so its state u_t accumulates the history of the channels rather than a snapshot of them. At every step, V_t modulates the neurons' firing thresholds and membrane time constants through learned coefficients. Since V_t rises and falls with the model's semantic spread, this coupling makes the network's excitability track the model's uncertainty.

In parallel, the network continuously predicts its own next state $\hat{u}_{t+1}$. The quality of this self-prediction is the detection signal: under grounded generation the channels drive regular dynamics and the prediction lands close to the true state, while hallucinated spans disrupt that rhythm and the prediction begins to miss. The windowed prediction error $s_t = \|u_t - \hat{u}_t\|$ is therefore the per-token anomaly score — the pipeline's output and the basis on which hallucination is detected — produced for each current token while the remaining text is still being written. Phase 3 is visualized in Figure 3.

3.3 Training Objective

LatticeSpike is trained end-to-end by backpropagating a ranking loss across its entire pipeline, with the discontinuity of spiking handled by surrogate gradients (Neftci et al., 2019). The ranking loss, computed on the per-token anomaly scores s_t, enforces a single property — score hallucinated tokens above grounded ones. Specifically, the loss blends two objectives: a pairwise margin loss (Burges et al., 2005), $\mathcal{L}_{\text{pw}}$, responsible for separating the scores of hallucinated tokens from the scores of grounded tokens, and a Smooth-AP loss (Brown et al., 2020), $\mathcal{L}_{\text{AP}}$, to ensure that the highest scores belong to hallucinated tokens.

For the set of grounded tokens G and the set of hallucinated tokens H, the two objectives are

$$\mathcal{L}_{\text{pw}} = \underset{i\in H,\, j\in G}{\text{mean}} \operatorname{softplus}\big(s_j - s_i + m\big), \qquad \mathcal{L}_{\text{AP}} = 1 - \frac{1}{|H|}\sum_{i\in H} \frac{1 + \sum_{k\in H} \sigma_\tau(s_k - s_i)}{1 + \sum_{k\in H} \sigma_\tau(s_k - s_i) + \sum_{j\in G} \sigma_\tau(s_j - s_i)},$$

where s_i and s_k are the anomaly scores of hallucinated tokens $i, k \in H$, s_j is the anomaly score of grounded token $j \in G$, m is the ranking margin, and σ_τ is a sigmoid relaxation of the ranking indicator with temperature τ. The full training loss, therefore, is the weighted blend

$$\mathcal{L} = \lambda \left[\alpha \, \mathcal{L}_{\mathrm{AP}} + (1 - \alpha) \, \mathcal{L}_{\mathrm{pw}} \right],$$

where λ scales the ranking loss, and the mixing weight α is a tuned hyperparameter selected on validation-split data, separate from training- and test-split data. Moreover, uncertain tokens enter no training pair, and the frozen language model receives no gradients. Additional details regarding LatticeSpike's training objective are provided in Section C.1 of the Appendix.

4 THRIVE BENCHMARK

4.1 DATASET CONSTRUCTION

Overview. We construct two matched versions of THRIVE (**T**oken-level **H**allucination **R**ecognition via **I**nter-model **V**oting and **E**valuation): **THRIVE-L**, generated with `Llama-3.2-1B-Instruct` (Meta AI, 2024), and **THRIVE-G**, generated with `Gemma-2-2B-it` (Gemma Team et al., 2024) (Figure 10). Each record preserves the generator's native tokenization and assigns every generated token one of three labels: G (grounded), H (hallucinated), or U (explicit uncertainty/abstention). We additionally define $t^\star$ as the first H token, allowing hallucination to be studied as a generation-time event rather than only as a property of a completed answer.

Prompt regimes and matched generation. Both THRIVE-L and THRIVE-G are generated from the same prompt matrix: 9,970 open-prompt questions drawn from eight QA benchmarks and a separate 981-example RAGTruth (Niu et al., 2024) arm. The latter uses RAGTruth questions and retrieved passages but regenerates the answer with the THRIVE generator; RAGTruth's original responses and span annotations are not transferred because they refer to different text and a passage-support rubric. The answer-forced RAGTruth arm is therefore analyzed separately from open-domain QA. Holding prompts fixed across Llama and Gemma enables cross-generator evaluation without confounding model identity with source composition. Generation is greedy with a 512-token ceiling, and the emitted token IDs define the annotation coordinate system; all labels, offsets, and $t^\star$ are tied to this native token spine.

Inter-model judging. Each response is evaluated by the same heterogeneous six-model panel: Aya-Expanse-32B (Dang et al., 2024), Gemma-2-27B-IT (Gemma Team et al., 2024), Llama-3.3-70B-Instruct (Grattafiori et al., 2024), Mistral-Small-24B-Instruct-2501 (Mistral AI, 2025), OLMo-2-32B-Instruct (Team OLMo et al., 2025), and Qwen2.5-32B-Instruct (Qwen Team et al., 2024). Judges return minimal *verbatim* false or uncertain spans rather than token indices. This separates semantic judgment from span bookkeeping: models decide *what* is wrong, while deterministic alignment maps the returned text onto the generator's token offsets. Repeated quotes are disambiguated by occurrence; paraphrases are not fuzzily mapped, because doing so would introduce token supervision not explicitly specified by the judge.

Aggregation and filtering. Final labels use `sentence:2:union`: a sentence is activated when at least two resolved judges identify hallucinated content within it, after which the supported spans are unioned inside that sentence. This requires agreement at the semantically stable sentence level without forcing several judges to choose identical

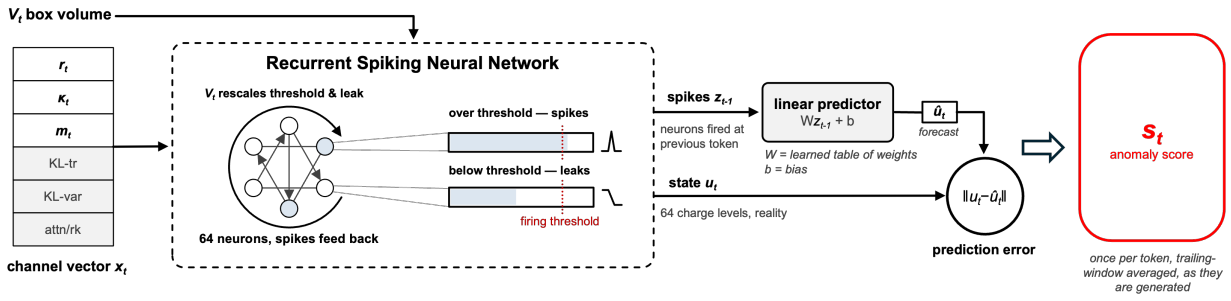


Figure 3: Phase three of the LatticeSpike pipeline.

subword boundaries. Failed judge outputs are treated as missing votes rather than implicit G labels. We exclude only reproducible failure cases, namely pathological generation loops and records with fewer than two resolving judges. Full source filtering, quote alignment, recovery, exclusion, and reproducibility details are provided in Appendix G.

4.2 Benchmark Details

Scale and evaluation. The common pre-filter matrix contains 10,951 instances per generator (9,970 open-prompt and 981 answer-forced). After deterministic exclusions, THRIVE-L contains 10,915 records (9,938 open-prompt and 977 answer-forced) and 898,586 generated tokens, while THRIVE-G contains 10,919 records (9,939 open-prompt and 980 answer-forced) and 242,160 generated tokens. Stable base prompt IDs are shared across generators, while generator-qualified record IDs preserve matched examples. The default automatic labels are evaluated against both a model-produced reference audit and independent blinded human annotations; because exact span placement is substantially less reproducible than record-level factuality even between humans, we report record-, token-, and onset-level agreement separately. Full construction statistics and validation details are provided in Appendix G.

Human labeling. We independently validate THRIVE-L and THRIVE-G with two human annotators per generator on length-stratified samples of 334 and 339 records, respectively. Samples are stratified by prompt regime and answer length and post-stratified back to the corpus distribution. Annotators use the same guide but label independently and without access to the automatic panel labels. Human–human agreement is high at the record level (Cohen's $\kappa = 0.886$ for THRIVE-L and 0.803 for THRIVE-G) but lower at the token level ($\kappa = 0.574$ and 0.634), providing an empirical inter-annotator reference for interpreting automatic localization performance.

Statistical validation. We evaluate the automatic panel against each annotator separately rather than collapsing the two annotations into a single synthetic gold label. On THRIVE-L, panel record-level F1 is 85.3% against annotator A and 81.7% against B, with token-level F1 of 49.6% and 42.9%. On THRIVE-G, record-level F1 is 79.7–82.4% and token-level F1 is 36.4–43.0%. Post-stratified human estimates place record-level hallucination prevalence at approximately 39% for both generators. Token metrics exclude formatting-only positions on both human and automatic sides; full sampling, agreement, onset, prevalence, and scoring details are reported in Appendix G.

Licensing and redistribution. Licensing and redistribution details for THRIVE-L and THRIVE-G are provided in Appendix G.5.8.

5 Experiments

5.1 Setup

Datasets. We evaluate LatticeSpike on four corpora, each pairing a different frozen generator with per-token labels. Each corpus provides its individual set of strengths that combined allow for a complete and informative assessment of our method. THRIVE (two versions, Llama-3.2-1B-Instruct (Meta AI, 2024) and Gemma-2-2B-it (Gemma Team et al., 2024) responses), explained thoroughly in Section 4, supplies questions drawn from nine QA benchmarks with long-form responses token-labeled by a human-validated six-judge panel; its two generators test whether detection survives a change of model family. RAGTruth (Niu et al., 2024) (Llama-2-7B-chat (Touvron et al., 2023) responses) supplies human span annotations covering question answering, data-to-text writing, and summarization; its gold-standard labels and cross-task evaluation protocol provide a judge-free measure of accuracy and a direct test of task transfer performance. Lastly, TriviaQA (Joshi et al., 2017) (Qwen3-8B (Yang et al., 2025) responses) supplies closed-book trivia questions with verified gold answers, isolating knowledge-gap hallucination against ground truth; token labels come from a three-judge (Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Gemma-2-9B-it (Gemma Team et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)) majority vote against the gold answer.

Baselines. We compare our method against 12 per-token detectors, spanning the main detection families. Trained heads over hidden states: a logistic-regression probe, SAPLMA (Azaria & Mitchell, 2023), a token-supervised SAPLMA variant, and HaMI (Niu et al., 2025). Training-free uncertainty scores: the five LLM-Check scores (Sriramanan et al., 2024) (token log-probability, logit entropy, window entropy, Hidden Score, Attention Score). Attention-based: Lookback Lens (Chuang et al., 2024). Cross-layer hidden-state dynamics: ICR Probe (Zhang et al., 2025) and a token-supervised variant. Every baseline runs under the same causal protocol as LatticeSpike; the adaptations causality requires and further details are documented in Appendix D.

Evaluation metrics. We report five token-level metrics: average precision (AP, equivalently AUPRC), area under the receiver operating characteristic curve (AUROC), recall, precision, and F1 score. Additionally, we report per-token detection latency and peak inference memory. Formal definitions and additional details are provided in Section E of the appendix.

Implementation. All methods run in a single evaluation harness: features are extracted from the frozen generator (language model) once per corpus and cached, and every detector consumes the same splits, prompts, and causal alignment. Trained detectors are fit per corpus on the training split. LatticeSpike's hyperparameters are tuned per corpus and selected on validation-split data (specific configurations are provided in Appendix C.2); the architecture stays fixed across datasets and the overall training loss differs only in how much weight α is assigned to each of the two loss objectives. Baseline hyperparameters are set to their respective published values, and are held fixed across corpora with necessary departures documented in Appendix D; where a source paper reports several configurations, as is the case with SAPLMA (Azaria & Mitchell, 2023), the choice follows the paper's own recommendation for the closest model family, established in advance rather than searched. Baseline architectures and loss functions follow their published forms and remain identical across corpora. Each detector's weight-matrix dimensions are set by the generator's hidden width, layer count, and vocabulary, identically for every method; no method is redesigned for a particular generator. Stochastically trained methods are run over five seeds and reported as mean $\pm$ standard deviation; deterministic fits and training-free scores admit no seed variance and are reported as single exact values.

With the exception of the architectural ablation (Section 5.3) and some hyperparameter screens (Section 5.3), which were conducted on an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM) and NVIDIA A10G GPUs (24 GB VRAM), respectively, all training and inference reported were conducted on an NVIDIA GeForce RTX 5090 GPU (32 GB VRAM). Both variants of THRIVE were constructed using NVIDIA H100 and H200 GPUs (80 GB, 141 GB VRAM).

5.2 Results

Full-corpus results. Under the full-corpus evaluation protocol, every detector is trained and evaluated on the complete training and test splits for each of the four corpora reported.

Method	AP	AUROC	R@1%FPR	P@1%FPR	F1@1%FPR
LatticeSpike (ours)	**0.447** $\pm$**0.007**	**0.866** $\pm$**0.003**	**0.272** $\pm$**0.006**	**0.645** $\pm$**0.005**	**0.382** $\pm$**0.007**
Lookback Lens	0.404	0.831	<u>0.251</u>	<u>0.626</u>	<u>0.358</u>
Hidden probe	<u>0.407</u>	<u>0.844</u>	0.240	0.616	0.345
SAPLMA	0.089 $\pm$0.007	0.594 $\pm$0.020	0.020 $\pm$0.005	0.118 $\pm$0.026	0.034 $\pm$0.009
SAPLMA (tok.)	0.360 $\pm$0.016	0.797 $\pm$0.006	0.225 $\pm$0.019	0.599 $\pm$0.020	0.327 $\pm$0.023
ICR Probe	0.084 $\pm$0.002	0.584 $\pm$0.005	0.019 $\pm$0.003	0.113 $\pm$0.013	0.033 $\pm$0.004
ICR (tok.)	0.146 $\pm$0.003	0.673 $\pm$0.002	0.067 $\pm$0.003	0.309 $\pm$0.009	0.110 $\pm$0.005
HaMI	0.146 $\pm$0.008	0.694 $\pm$0.005	0.063 $\pm$0.012	0.294 $\pm$0.039	0.104 $\pm$0.018
LLM-Chk (logit)	0.080	0.563	0.025	0.141	0.042
LLM-Chk (win.)	0.081	0.564	0.026	0.147	0.044
LLM-Chk (hid.)	0.094	0.641	0.022	0.126	0.037
LLM-Chk (attn.)	0.074	0.526	0.021	0.121	0.035
Token logprob	0.074	0.559	0.012	0.075	0.021

Table 1: RAGTruth (Llama-2-7b-chat, layer 16): 87,678 metric tokens, prevalence 6.25%, random-AP floor 0.063.

Across all full-corpus evaluations, spanning 12 baselines on THRIVE-L, THRIVE-G, RAGTruth, and TriviaQA, LatticeSpike leads every token-level metric — AP, AUROC, and recall, precision, and F1 score at a 1% false-positive rate (FPR). Additionally, to rule out sensitivity to the 1% FPR budget, we sweep the budget from 0.1% to 5% and report recall, precision, and F1 score at each respective budget (AP and AUROC are threshold-free): LatticeSpike leads every token-level metric, across all datasets, spanning 5%, 1%, and 0.5% false-positive budgets. At the 0.1% budget, LatticeSpike wins all three metrics on THRIVE-L, THRIVE-G, and TriviaQA; on RAGTruth, LatticeSpike leads on one of the three metrics (recall) and ties with Lookback Lens (Chuang et al., 2024) on the remaining two (precision and F1). Concretely, across 56 unique evaluation cells, LatticeSpike wins 54/56 and ties on the last two. Lastly, we show that thresholds fixed on validation transfer to test within 0.003 of the 1% FPR target (Figure 6, left), with LatticeSpike again attaining the best precision, recall, and F1 score on every corpus (Figure 6, right).

Lookback Lens (Chuang et al., 2024) and the Hidden Probe are consistently the two strongest performers after LatticeSpike: the probe on THRIVE-L, THRIVE-G, and RAGTruth, Lookback Lens on TriviaQA.

Results are grouped by corpus and ordered as full results, FPR-budget sweep, and validation-selected operating point: THRIVE-Llama (Tables 3, 4, 5), THRIVE-Gemma (Tables 6, 7, 8), RAGTruth (Tables 1, 2, 9), and TriviaQA (Tables 10, 11, 12). Most tables, along with further full-corpus experimental details, sit in Appendix B.1, which also plots these results across all four corpora (Figure 4) and across the full FPR-budget sweep (Figure 5). Across all results tables, bolded cells denote the best performer, underlined cells denote the runner-up.

	FPR $\leq 5\%$			FPR $\leq 1\%$			FPR $\leq 0.5\%$			FPR $\leq 0.1\%$		
Method	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LatticeSpike (ours)	**0.535** $\pm$**0.007**	**0.417** $\pm$**0.003**	**0.469** $\pm$**0.005**	**0.272** $\pm$**0.006**	**0.645** $\pm$**0.005**	**0.382** $\pm$**0.007**	**0.186** $\pm$**0.008**	**0.713** $\pm$**0.008**	**0.295** $\pm$**0.010**	**0.064** $\pm$**0.006**	**0.809** $\pm$**0.013**	**0.118** $\pm$**0.010**
Lookback Lens	0.475	0.388	0.427	0.251	0.626	0.358	0.172	0.697	0.275	0.063	**0.809**	**0.118**
Hidden probe	0.481	0.391	0.431	0.240	0.616	0.345	0.163	0.685	0.263	0.060	0.800	0.112
SAPLMA	0.096 $\pm$0.015	0.113 $\pm$0.016	0.104 $\pm$0.016	0.020 $\pm$0.005	0.118 $\pm$0.026	0.034 $\pm$0.009	0.011 $\pm$0.003	0.127 $\pm$0.033	0.020 $\pm$0.006	0.003 $\pm$0.001	0.146 $\pm$0.059	0.005 $\pm$0.002
SAPLMA (tok.)	0.448 $\pm$0.017	0.374 $\pm$0.009	0.408 $\pm$0.013	0.225 $\pm$0.019	0.599 $\pm$0.020	0.327 $\pm$0.023	0.145 $\pm$0.015	0.658 $\pm$0.024	0.237 $\pm$0.022	0.046 $\pm$0.005	0.755 $\pm$0.019	0.088 $\pm$0.009
ICR Probe	0.088 $\pm$0.005	0.105 $\pm$0.006	0.095 $\pm$0.006	0.019 $\pm$0.003	0.113 $\pm$0.013	0.033 $\pm$0.004	0.010 $\pm$0.002	0.118 $\pm$0.019	0.018 $\pm$0.003	0.002 $\pm$0.001	0.111 $\pm$0.044	0.004 $\pm$0.002
ICR (tok.)	0.180 $\pm$0.004	0.194 $\pm$0.003	0.187 $\pm$0.004	0.067 $\pm$0.003	0.309 $\pm$0.009	0.110 $\pm$0.005	0.044 $\pm$0.004	0.368 $\pm$0.022	0.078 $\pm$0.007	0.018 $\pm$0.002	0.541 $\pm$0.032	0.034 $\pm$0.004
HaMI	0.183 $\pm$0.012	0.196 $\pm$0.011	0.189 $\pm$0.012	0.063 $\pm$0.012	0.294 $\pm$0.039	0.104 $\pm$0.018	0.039 $\pm$0.009	0.341 $\pm$0.056	0.070 $\pm$0.016	0.012 $\pm$0.004	0.428 $\pm$0.093	0.023 $\pm$0.007
LLM-Chk (logit)	0.087	0.104	0.095	0.025	0.141	0.042	0.014	0.155	0.025	0.003	0.180	0.006
LLM-Chk (win.)	0.088	0.105	0.096	0.026	0.147	0.044	0.016	0.172	0.028	0.005	0.248	0.010
LLM-Chk (hid.)	0.087	0.104	0.094	0.022	0.126	0.037	0.012	0.135	0.021	0.000	0.000	0.000
LLM-Chk (attn.)	0.081	0.098	0.089	0.021	0.121	0.035	0.011	0.126	0.020	0.003	0.155	0.005
Token logprob	0.066	0.081	0.073	0.012	0.075	0.021	0.007	0.087	0.013	0.001	0.079	0.003

Table 2: RAGTruth (Llama-2-7b-chat): recall, precision, and F1 across false-positive budgets (test-set thresholds).

Cross-task evaluation. Following the transfer design of Lookback Lens (Chuang et al., 2024), introduced in their work to show that attention features transfer where hidden-state probes overfit, we train LatticeSpike, Lookback Lens, and the hidden-state probe (the strongest performers as per the results in Section 5.2) on a single RAGTruth generation task — question answering, summarization, or data-to-text — and evaluate each on all three. This evaluation yields nine source–target pairs per metric: AP (Table 13, Figure 7) and AUROC (Table 14), and recall (Table 15), precision (Table 16), and F1 score (Table 17) at 1% FPR. LatticeSpike wins all 30 out-of-domain cells (train on one task, evaluate on another), and 12/15 in-domain cells (train and evaluate on the same task), for a total of 42/45 cells won. The three in-domain cells lost belong to the data-to-text task, and are recall, precision, and F1; the hidden-state probe leads these three cells while LatticeSpike retains the task-best AP and AUROC.

Computational load and edge-deployment. We measure what LatticeSpike adds on top of streaming generation to assess whether real-time hallucination detection is feasible on resource-constrained hardware. We find that the added cost is small in absolute terms, but high relative to most baselines: 3.6–6.3 ms per token (37% of a generation step on the Llama-3.2-1B host, shrinking to 17% on the Llama-2-7B host), while adding at most 103 MiB of peak memory. The recurrent core runs faster on CPU (0.15–0.19 ms per token) than on GPU, so detection needs no accelerator of its own (Table 19). Importantly, the streaming path is verified against the reported predictions: for every method on every corpus, its AP, AUROC, and 1%-FPR metrics match those of the recorded scores to within 10^{-2}, and typically within 2×10^{-3}, over the full test pools; exact equality is precluded by non-deterministic bf16 execution. Additional details and full results tables are provided in Appendix B.3.

Additionally, we host a live demo of LatticeSpike to further emphasize and show the model's deployability, and provide readers with direct access to an interactive system: `https://huggingface.co/spaces/anonymous-submission-hugging-face/Lattice_spike_test`. More details on this demo can be found in Appendix F.

Statistical significance testing. Using a paired cluster bootstrap over test responses (B=10,000, one-sided, Benjamini–Hochberg correction at q=0.05), we compare LatticeSpike's five-seed mean against the three strongest baselines (Lookback Lens (Chuang et al., 2024), hidden-state probe, and SAPLMA-tok (Azaria & Mitchell, 2023)), on the five-metric full-corpus evaluations along with the FPR sweeps. Additionally, we compare our five-seed means from the cross-task protocol against the two cross-task baselines (Lookback Lens and hidden-state probe) under the same statistical significance testing procedure. Lastly, we provide statistical significance testing for our architectural and component ablations under the same testing protocol. A detailed discussion, including tables and findings, is provided in Section B.5 of the Appendix.

5.3 Ablation Studies

Architectural ablation. We isolate the contribution of the proposed spiking dynamics by replacing only the RSNN with a parameter-matched LSTM (Hochreiter & Schmidhuber, 1997; Gers et al., 2000) or causal Transformer (Vaswani

et al., 2017) and retraining on THRIVE-L (five random seeds per variant, 42–46). The box encoder, six input channels, training objective, data split, checkpoint-selection rule, and evaluation pipeline are held fixed. Appendix B.4.1 details the full results, controls, parameter matching, and recall sweep.

Component ablation. We remove one major mechanism at a time from the reported THRIVE-L configuration and retrain: the volume modulation of the neurons' dynamics, the concretization channel m_t, and the three box-geometry channels (r_t, κ_t, m_t zeroed jointly). Every removal degrades all five metrics relative to the full model, with the concretization channel being the most impactful of the three, followed by the box-geometry channels, and lastly the volume modulation. These results, and further discussion, are provided in Table 24 and Section B.4.2.

Hyperparameter ablation. Across the hyperparameter study we ran over 120 individual experiments screening learning rate, batch size and batch balancing, loss-term weights, the relaxation temperature τ, the objective-mix weight α, box dimension and neuron count, and encoder variants to determine how each of these hyperparameters affected LatticeSpike's performance. An overview of the most impactful experiments, along with additional details on hyperparameters, is provided in Appendix B.4.3.

6 Limitations and Future Work

Limitations. Although LatticeSpike outperforms both the hidden-state probe and Lookback Lens in 42 of 45 cross-task cells, only $\sim 58\%$ of the corresponding cross-task comparisons reach significance. Comparatively, $\sim 82\%$ of the five-metric full-corpus comparisons reach significance. For non-significant comparisons, results should not be overinterpreted. Moreover, the architectural ablation shows that, when all else is held fixed, on THRIVE-L an RSNN backbone is uniformly better on ranking metrics (AP and AUROC, mostly significant on AP and fully significant on AUROC), but not necessarily thresholded metrics (recall, precision, F1 score): the RSNN only attains the highest recall at 5% FPR, while a parameter- and loss-matched Transformer leads on all three thresholded metrics at 1% FPR, and on recall at the 0.5% and 0.1% budgets; however, no thresholded comparison reaches significance in either direction, and thus thresholded results should not be overinterpreted.

Future work. Based on the findings of this work, and the limitations discussed, we identify three opportunities for future work. First, the architectural ablation reveals that an opportunity for performance improvement could exist under an architecture that combines the strengths of both an RSNN and a Transformer, with the RSNN supplying the ranking quality (AP and AUROC) and the Transformer supplying the detection quality at tight false-positive budgets (recall, precision, and F1 score at low FPR). Second, most of LatticeSpike's added latency comes from constructing its input signals rather than from the network itself; cheaper signal approximations would remove most of the detector's overhead. Third, LatticeSpike requires token labels for each new corpus or generator, which is the main barrier to applying it more widely; training from response-level labels alone or testing transfer from a labeled corpus to an unlabeled one remain open research directions.

7 Conclusion

In this paper we have introduced LatticeSpike, an empirically compelling and biologically inspired hallucination detector capable of operating in deployed settings, in real-time. Additionally, we have released two human-validated and statistically tested token-level benchmarks, spanning two generator families, THRIVE-Llama and THRIVE-Gemma.

Notably, across four corpora and multiple generator families, LatticeSpike outperforms the tested baselines in 96 of 101 evaluation cells ($\sim 95\%$ win rate), including strong cross-task transfer on RAGTruth. Its causal, token-level design also remains suitable for real-time detection: although 3.6–6.3 ms per token of added overhead is high relative to most tested baselines, in absolute terms this is still a small computational cost for deployment, especially when paired with the detector's memory usage (at most 103 MiB of peak memory). Furthermore, through a suite of architectural, component, and hyperparameter ablations we show where LatticeSpike's success stems from — providing insight into possible directions for future work. Lastly, through statistical testing we show exactly where LatticeSpike's performance is significant and where it should not be overinterpreted.

Together, our results and released materials aim to advance the development and rigorous evaluation of practical, real-time hallucination detection systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript editing, coding, and certain aspects of the ideation and brainstorming process. We additionally used AI to assist with related work discovery. Lastly, LMs were used in our THRIVE-L and THRIVE-G dataset construction processes, as explained in detail in Section 4. Importantly, all AI-generated content has been screened and validated by the authors prior to its inclusion in this work. The authors assume full responsibility for the final content of this work, including that which was produced with the aid of generative AI.

ETHICS STATEMENT

For the LatticeSpike portion of the work, we use publicly available benchmark datasets and standard hallucination detection protocols. It does not involve human-subjects experiments, collection of sensitive personal data, or have applications with foreseeable direct societal harm.

Regarding THRIVE-L and THRIVE-G, these benchmarks are built from nine publicly released question-answering benchmarks, in accordance with their licenses and permitted redistribution criteria, reproducing their questions and grounding passages verbatim so that token labels remain aligned to the text they describe. Because the passages are reproduced unchanged, a very small amount of contact information present in the upstream sources is carried over. We are aware of 22 such occurrences across 16 records in THRIVE-L and 19 occurrences across 16 records in THRIVE-G, all of which originate from RAGTruth (Niu et al., 2024) (15 occurrences across 6 records) and HaluBench (Ravi et al., 2024) (26 occurrences across 26 records). **Importantly, we do not introduce nor gather any new personal data and perform no identification of individuals.**

Therefore, there are no ethical concerns associated with this work.

REPRODUCIBILITY STATEMENT

We provide substantive detail in this manuscript regarding LatticeSpike and both THRIVE datasets (THRIVE-L and THRIVE-G) to ensure reproducibility of our work. Furthermore, we release an anonymized version of the code and the datasets for the same purpose: `https://anonymous.4open.science/r/LatticeSpike-8EB2`.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-AP: Smoothing the path towards large-scale image retrieval. In *Computer Vision – ECCV 2020*, volume 12354 of *Lecture Notes in Computer Science*, pp. 677–694. Springer, 2020. doi: 10.1007/978-3-030-58545-7_39.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96. Association for Computing Machinery, 2005. doi: 10.1145/1102351.1102363.

Yann Cherdo, Benoît Miramond, and Alain Pegatoquet. Time series prediction and anomaly detection with recurrent spiking neural networks. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2023. doi: 10.1109/IJCNN54540.2023.10191614.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.84. URL https://aclanthology.org/2024.emnlp-main.84/.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36(1–4):345–384, 2011.

John Dang et al. Aya expanse: Combining research breakthroughs for a new multilingual frontier, 2024. URL https://arxiv.org/abs/2412.04261.

Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving local identifiability in probabilistic box embeddings. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/01c9d2c5b3ff5cbba349ec39a570b5e3-Abstract.html.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin, Heidelberg, 1999.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. doi: 10.48550/arXiv.2408.00118. URL https://arxiv.org/abs/2408.00118.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. SimpleQA verified: A reliable factuality benchmark to measure parametric knowledge, 2025. URL https://arxiv.org/abs/2509.07968.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

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, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. doi: 10.48550/arXiv.2310.06825. URL https://arxiv.org/abs/2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Matic Korun. Detecting LLM hallucinations via embedding cluster geometry: A three-type taxonomy with measurable signatures, 2026a. URL https://arxiv.org/abs/2602.14259.

Matic Korun. Whitening reveals cluster commitment as the geometric separator of hallucination types, 2026b. URL https://arxiv.org/abs/2603.07755.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2192. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c32319f4868da7613d78af9993100e42-Abstract-Conference.html.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612. URL https://aclanthology.org/P19-1612/.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6449–6464, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.397. URL https://aclanthology.org/2023.emnlp-main.397/.

Miaoran Li, Rogger Luo, and Ofer Mendelevitch. HHEM-2.1-Open, 2024. URL https://huggingface.co/vectara/hallucination_evaluation_model.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10 (9):1659–1671, 1997. doi: 10.1016/S0893-6080(97)00011-7.

Piotr S. Maciąg, Marzena Kryszkiewicz, Robert Bembenik, Jesus L. Lobo, and Javier Del Ser. Unsupervised anomaly detection in stream data with online evolving spiking neural networks. *Neural Networks*, 139:118–139, 2021. doi: 10.1016/j.neunet.2021.02.017.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL https://aclanthology.org/2020.acl-main.173/.

Meta AI. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. Meta AI Blog, September 2024. URL https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Mistral AI. Mistral small 3. https://mistral.ai/news/mistral-small-3/, January 2025. Blog post. Model card: https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.585. URL https://aclanthology.org/2024.acl-long.585/.

Mengjia Niu, Hamed Haddadi, and Guansong Pang. Robust hallucination detection in LLMs via adaptive token selection. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4213. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/b7c43d4a79dede363a2d061c6158e5a5-Abstract-Conference.html.

Nostalgebraist. Interpreting GPT: The logit lens. LessWrong, August 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2026-09-21.

Yuya Ogasa and Yuki Arase. Hallucinated span detection with multi-view attention features. In *Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pp. 381–394. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.starsem-1.31. URL https://aclanthology.org/2025.starsem-1.31/.

Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in Neuroscience*, 12:774, 2018. doi: 10.3389/fnins.2018.00774.

Edward Phillips, Sean Wu, Soheila Molaei, Danielle Belgrave, Anshul Thakur, and David Clifton. Geometric uncertainty for detecting and correcting hallucinations in LLMs, 2025. URL https://arxiv.org/abs/2509.13813.

Qwen Team, An Yang, et al. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. Lynx: An open source hallucination evaluation model, 2024. URL https://arxiv.org/abs/2407.08488.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference (EUSIPCO 2007)*, pp. 606–610, Poznań, Poland, 2007. URL https://infoscience.epfl.ch/handle/20.500.14299/10320.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. LLM-Check: Investigating detection of hallucinations in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 34188–34216. Curran Associates, Inc., 2024. doi: 10.52202/079017-1077. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3c1e1fdf305195cd620c118aaa9717ad-Abstract-Conference.html.

Team OLMo, Pete Walsh, et al. 2 OLMo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656. Shorter version accepted to COLM 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. doi: 10.48550/arXiv.2307.09288. URL https://arxiv.org/abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 263–272, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1025. URL https://aclanthology.org/P18-1025/.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.813. URL https://aclanthology.org/2024.findings-acl.813/.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models, 2024. URL https://arxiv.org/abs/2411.04368.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Zhenliang Zhang, Xinyu Hu, Huixuan Zhang, Junzhe Zhang, and Xiaojun Wan. ICR probe: Tracking hidden state dynamics for reliable hallucination detection in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17986–18002, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.880. URL https://aclanthology.org/2025.acl-long.880/.

APPENDIX

A MORE RELATED WORK

Attention structure has also been used for fine-grained localization. Ogasa and Arase (Ogasa & Arase, 2025) derive multiple complementary features from transformer attention matrices, capturing properties such as whether tokens are influential or ignored, whether attention concentrates on particular subsets of the context, and whether generation depends on narrow or broad contextual regions. These signals are processed by a Transformer based sequential classifier for hallucinated span detection.

Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) constructs a single representation whose nested coordinate prefixes are simultaneously trained to be useful at multiple dimensionalities. Instead of learning a separate embedding for each desired capacity, MRL organizes information from coarse to fine so that progressively larger prefixes provide increasingly expressive representations. This allows the dimensionality of a learned representation to be selected according to downstream computational or storage constraints while retaining a shared underlying embedding. More broadly, MRL provides a mechanism for organizing information at multiple granularities within a common representation space.

In contrast to white-box approaches that inspect the generator's internal computation, HHEM-2.1-Open (Li et al., 2024) formulates hallucination evaluation as reference-conditioned factual consistency: a premise and generated hypothesis are jointly scored according to whether the hypothesis is supported by the premise. These directions illustrate complementary approaches based on hidden activations, attention behavior, token selection, sequential labeling, and external support relationships.

Another family of methods approaches hallucination through uncertainty and model self-knowledge. Kadavath et al. (Kadavath et al., 2022) investigate whether language models can estimate the correctness of their own answers through quantities such as $P(\mathrm{True})$, which evaluates a proposed answer, and $P(\mathrm{IK})$, which estimates whether the model "knows" the answer independently of a particular generation. This perspective is closely related to selective prediction, where a predictor may abstain on uncertain inputs and is evaluated through the trade-off between predictive risk and coverage (El-Yaniv & Wiener, 2010).

Semantic uncertainty methods instead reason over multiple possible generations. Kuhn et al. (Kuhn et al., 2023) introduce *semantic entropy*, which accounts for the fact that lexically different responses can express the same underlying meaning. Rather than treating every sampled sequence as a distinct outcome, semantically equivalent generations are grouped before uncertainty is computed. This provides an uncertainty measure over meanings rather than surface forms. Complementary work has examined whether factuality itself is geometrically represented inside language models. Marks and Tegmark (Marks & Tegmark, 2024) study hidden representations of true and false statements and identify approximately linear directions associated with truth values, linking probing-based detection to the geometry of internal representation spaces.

Recent concurrent work has investigated hallucination through the geometry of embedding spaces. Korun (Korun, 2026a) proposes a taxonomy based on token-embedding cluster structure, distinguishing *center-drift*, *wrong-well convergence*, and *coverage-gap* behaviors and characterizing them using statistics related to cluster polarity, cohesion, and radial organization. Follow-up work (Korun, 2026b) studies this taxonomy after whitening the embedding space and examines *cluster commitment* as a geometric property for distinguishing different failure regimes. These approaches view hallucination as a change in the organization or trajectory of representations relative to the local geometry of the embedding space.

Phillips et al. (Phillips et al., 2025) develop a complementary black-box geometric approach based on multiple sampled responses rather than internal token representations. Their framework applies archetypal analysis to response embeddings and defines a global uncertainty quantity using the convex-hull volume of the resulting archetypes. It additionally assigns response-level geometric suspicion scores to characterize individual generations within the sampled response set. Together with work on linear truth representations, these studies motivate geometric descriptions of factuality and uncertainty at several scales, ranging from hidden-state directions and token clusters to sets of complete response embeddings.

B Additional Experiments

B.1 Full-Corpus Results

Figure 4 summarizes the full-corpus results of Tables 3, 6, 1, and 10 across all five metrics, and Figure 5 shows the recall-vs-FPR-budget sweep of Tables 4, 7, 2, and 11.

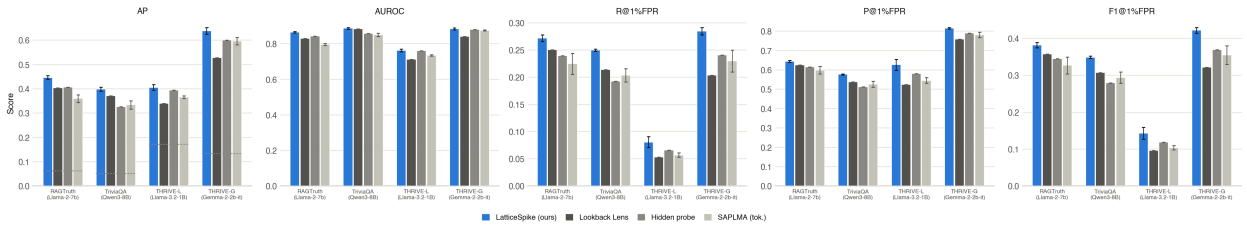


Figure 4: Main results across the four evaluation corpora, restricted to the four token-level-comparable methods (LatticeSpike, Lookback Lens, the hidden-state probe, and SAPLMA-tok.). Dashed line marks the random-AP floor; error bars are sd over five seeds.

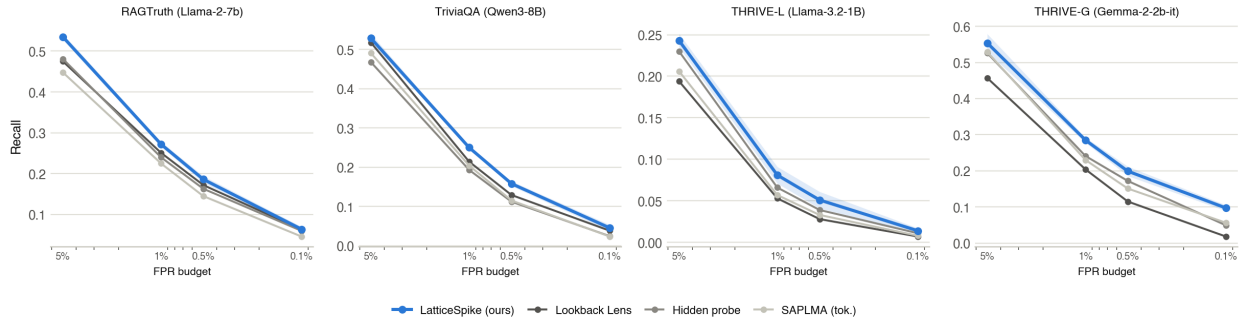


Figure 5: Recall across false-positive budgets for the four token-level-comparable methods. The gap over the best baseline is generally largest around the 1–0.5% FPR budgets and narrows again toward 0.1%, where the small number of tokens defining that threshold makes all methods noisier. Shaded band is sd over five seeds for LatticeSpike.

THRIVE-Llama. Table 3 reports the full set of results for THRIVE-Llama; recall, precision, and F1 score are under 1% FPR. Table 4 reports recall, precision, and F1 across false-positive-rate budgets from 0.1% to 5%, and Table 5 reports the deployed operating point. Under the same strict 1% false-alarm budget, LatticeSpike detects hallucinations more efficiently than the hidden-state probe, improving recall from 0.066 to 0.081 and F1 from 0.119 to 0.143.

THRIVE-Gemma. Table 6 shows the full set of results for THRIVE-Gemma; recall, precision, and F1 score are under 1% FPR. Table 7 reports recall, precision, and F1 across false-positive-rate budgets from 0.1% to 5%, and Table 8 reports the deployed operating point. Most methods achieve higher absolute performance on THRIVE-Gemma than on THRIVE-Llama, suggesting that token-level hallucination detectability may vary with the generator. Despite this variation, LatticeSpike remains the top-performing method on both corpora.

RAGTruth. Table 1, shown in the main text, displays the full set of results for RAGTruth; recall, precision, and F1 score are under 1% FPR. Table 2, also shown in the main text, reports recall, precision, and F1 across false-positive-rate budgets from 0.1% to 5%, and Table 9 reports the deployed operating point. LatticeSpike leads across the 5%, 1%, and 0.5% budgets; at the most restrictive 0.1% budget, it matches Lookback Lens in precision and F1 while retaining slightly higher recall. Its advantage also survives threshold transfer, achieving a deployed F1 score of 0.342 compared with 0.324 for the strongest baseline.

Method	AP	AUROC	R@1%FPR	P@1%FPR	F1@1%FPR
LatticeSpike (ours)	**0.406** ±0.012	**0.763** ±0.007	**0.081** ±0.010	**0.627** ±0.028	**0.143** ±0.016
Lookback Lens	0.340	0.713	0.053	0.525	0.096
Hidden probe	0.394	0.761	0.066	0.581	0.119
SAPLMA	0.233 ±0.007	0.602 ±0.008	0.023 ±0.002	0.325 ±0.023	0.043 ±0.004
SAPLMA (tok.)	0.366 ±0.006	0.735 ±0.004	0.057 ±0.004	0.546 ±0.015	0.104 ±0.006
ICR Probe	0.172 ±0.001	0.508 ±0.003	0.007 ±0.000	0.126 ±0.008	0.013 ±0.001
ICR (tok.)	0.232 ±0.001	0.581 ±0.001	0.029 ±0.001	0.379 ±0.005	0.054 ±0.001
HaMI	0.228 ±0.018	0.587 ±0.012	0.022 ±0.007	0.307 ±0.072	0.041 ±0.013
LLM-Chk (logit)	0.220	0.564	0.029	0.374	0.053
LLM-Chk (win.)	0.224	0.566	0.033	0.408	0.061
LLM-Chk (hid.)	0.164	0.482	0.009	0.154	0.016
LLM-Chk (attn.)	0.195	0.540	0.015	0.237	0.028
Token logprob	0.217	0.561	0.030	0.385	0.055

Table 3: THRIVE-L (Llama-3.2-1B); 83,582 metric tokens, prevalence 17.30%, random-AP floor 0.173.

	FPR ≤ 5%			FPR ≤ 1%			FPR ≤ 0.5%			FPR ≤ 0.1%		
Method	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LatticeSpike (ours)	**0.243** ±0.008	**0.504** ±0.009	**0.328** ±0.009	**0.081** ±0.010	**0.627** ±0.028	**0.143** ±0.016	**0.051** ±0.010	**0.675** ±0.041	**0.094** ±0.017	**0.014** ±0.003	**0.741** ±0.046	**0.028** ±0.006
Lookback Lens	0.194	0.448	0.270	0.053	0.525	0.096	0.028	0.539	0.053	0.007	0.599	0.014
Hidden probe	0.230	0.490	0.313	0.066	0.581	0.119	0.039	0.621	0.074	0.011	0.688	0.021
SAPLMA	0.094 ±0.005	0.283 ±0.011	0.142 ±0.007	0.023 ±0.002	0.325 ±0.023	0.043 ±0.004	0.013 ±0.002	0.351 ±0.027	0.025 ±0.003	0.003 ±0.000	0.362 ±0.035	0.005 ±0.001
SAPLMA (tok.)	0.206 ±0.007	0.463 ±0.008	0.285 ±0.008	0.057 ±0.004	0.546 ±0.015	0.104 ±0.006	0.033 ±0.002	0.576 ±0.014	0.062 ±0.003	0.008 ±0.001	0.621 ±0.017	0.015 ±0.001
ICR Probe	0.042 ±0.001	0.151 ±0.003	0.066 ±0.002	0.007 ±0.000	0.126 ±0.008	0.013 ±0.001	0.003 ±0.000	0.124 ±0.009	0.007 ±0.001	0.001 ±0.000	0.124 ±0.028	0.001 ±0.000
ICR (tok.)	0.110 ±0.001	0.314 ±0.002	0.162 ±0.001	0.029 ±0.001	0.379 ±0.005	0.054 ±0.001	0.016 ±0.000	0.401 ±0.005	0.031 ±0.001	0.004 ±0.000	0.440 ±0.015	0.007 ±0.000
HaMI	0.093 ±0.020	0.279 ±0.043	0.140 ±0.028	0.022 ±0.007	0.307 ±0.072	0.041 ±0.013	0.012 ±0.005	0.325 ±0.087	0.023 ±0.009	0.004 ±0.002	0.394 ±0.163	0.007 ±0.004
LLM-Chk (logit)	0.092	0.277	0.138	0.029	0.374	0.053	0.020	0.452	0.038	0.007	0.608	0.015
LLM-Chk (win.)	0.097	0.289	0.145	0.033	0.408	0.061	0.022	0.476	0.042	0.008	0.621	0.015
LLM-Chk (hid.)	0.043	0.154	0.068	0.009	0.154	0.016	0.003	0.099	0.005	0.001	0.138	0.002
LLM-Chk (attn.)	0.070	0.225	0.106	0.015	0.237	0.028	0.006	0.212	0.012	0.002	0.250	0.003
Token logprob	0.086	0.265	0.130	0.030	0.385	0.055	0.019	0.443	0.036	0.008	0.617	0.015

Table 4: THRIVE-L (Llama-3.2-1B): recall, precision, and F1 across false-positive budgets (test-set thresholds).

TriviaQA. Table 10 shows the full set of results for TriviaQA; recall, precision, and F1 score are under 1% FPR. Table 11 reports recall, precision, and F1 across false-positive-rate budgets from 0.1% to 5%, and Table 12 reports the deployed operating point. LatticeSpike consistently outperforms Lookback Lens, the strongest baseline on this corpus, across every evaluated FPR budget; at 1% FPR, its F1 score increases from 0.307 to 0.349. The validation-selected operating point preserves this ordering, with F1 scores of 0.336 and 0.292, respectively.

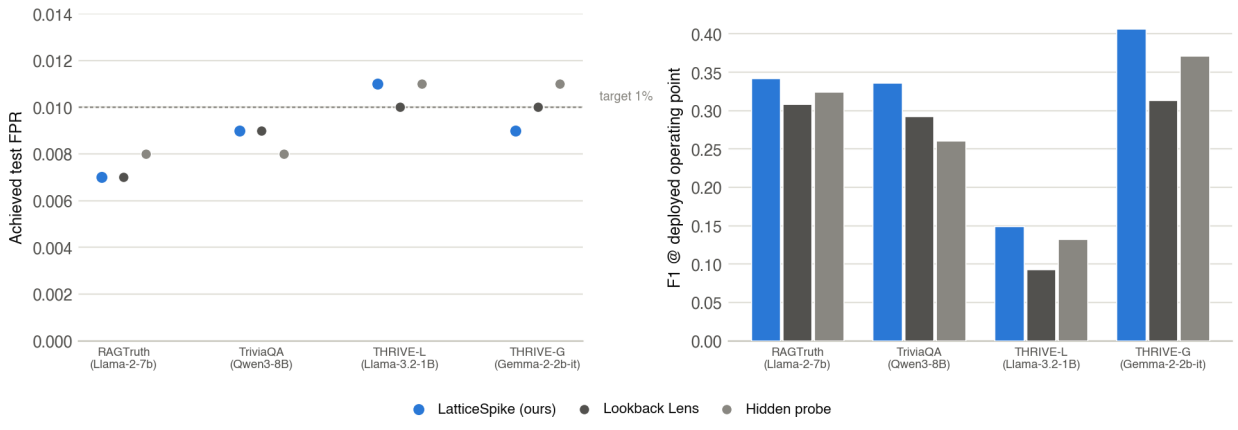


Figure 6: Deployed operating point across the four corpora using Tables 5, 8, 9, and 12. The left panel shows achieved test FPR against the 1% validation target; the right panel shows F1 at that fixed threshold.

Method	Test FPR (target ≤0.010)	Precision	Recall	F1
LatticeSpike (ours)	0.011	**0.623**	**0.084**	**0.149**
Lookback Lens	0.010	0.529	0.051	0.093
Hidden probe	0.011	0.579	0.074	0.132
SAPLMA	0.009	0.333	0.021	0.039
SAPLMA (tok.)	0.009	0.548	0.055	0.100
ICR Probe	0.012	0.126	0.008	0.015
ICR (tok.)	0.009	0.381	0.027	0.050
HaMI	0.013	0.306	0.027	0.049
LLM-Chk (logit)	0.011	0.367	0.030	0.056
LLM-Chk (win.)	0.011	0.399	0.035	0.064
LLM-Chk (hid.)	0.016	0.157	0.014	0.026
LLM-Chk (attn.)	0.009	0.237	0.014	0.026
Token logprob	0.011	0.381	0.032	0.058

Table 5: THRIVE-L (Llama-3.2-1B): deployed operating point – threshold selected on validation at ≤1% FPR, applied unchanged to test.

Method	AP	AUROC	R@1%FPR	P@1%FPR	F1@1%FPR
LatticeSpike (ours)	**0.639** ±0.014	**0.885** ±0.006	**0.285** ±0.007	**0.816** ±0.004	**0.422** ±0.008
Lookback Lens	0.528	0.842	0.204	0.760	0.322
Hidden probe	0.601	0.880	0.241	0.790	0.369
SAPLMA	0.245 ±0.021	0.692 ±0.018	0.031 ±0.008	0.325 ±0.049	0.057 ±0.013
SAPLMA (tok.)	0.597 ±0.015	0.876 ±0.003	0.230 ±0.020	0.781 ±0.014	0.355 ±0.025
ICR Probe	0.175 ±0.006	0.595 ±0.011	0.019 ±0.003	0.230 ±0.030	0.036 ±0.006
ICR (tok.)	0.426 ±0.004	0.793 ±0.002	0.125 ±0.003	0.661 ±0.005	0.211 ±0.004
HaMI	0.371 ±0.051	0.776 ±0.013	0.064 ±0.043	0.442 ±0.209	0.111 ±0.073
LLM-Chk (logit)	0.254	0.600	0.085	0.570	0.148
LLM-Chk (win.)	0.256	0.602	0.102	0.614	0.176
LLM-Chk (hid.)	0.151	0.557	0.011	0.151	0.021
LLM-Chk (attn.)	0.189	0.611	0.022	0.251	0.040
Token logprob	0.249	0.597	0.084	0.566	0.146

Table 6: THRIVE-G (Gemma-2-2b-it, layer 13): 22,809 metric tokens, prevalence 13.43%, random-AP floor 0.134.

B.2 Cross-Task Evaluation

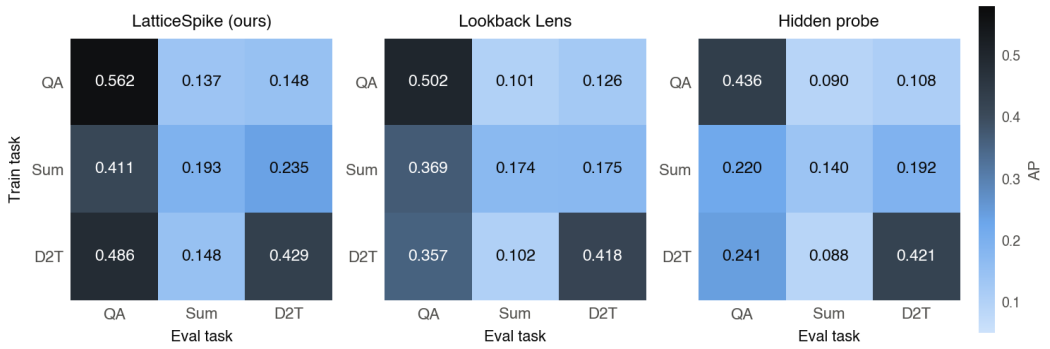


Figure 7: RAGTruth cross-task transfer AP from Table 13 as source×target heatmaps, one per method on a shared color scale. Diagonal cells denote in-domain evaluation, where training and evaluation use the same task. Compare cells within each target-task column, not across columns, because the random-AP floor differs by target task: QA 0.073, Sum. 0.045, and D2T 0.062.

Method	FPR ≤ 5%			FPR ≤ 1%			FPR ≤ 0.5%			FPR ≤ 0.1%		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LatticeSpike (ours)	**0.554** ±0.025	**0.632** ±0.011	**0.590** ±0.019	**0.285** ±0.007	**0.816** ±0.004	**0.422** ±0.008	**0.199** ±0.013	**0.861** ±0.008	**0.324** ±0.018	**0.097** ±0.008	**0.940** ±0.005	**0.176** ±0.013
Lookback Lens	0.457	0.586	0.513	0.204	0.760	0.322	0.115	0.782	0.200	0.018	0.747	0.036
Hidden probe	0.526	0.620	0.570	0.241	0.790	0.369	0.172	0.843	0.285	0.049	0.887	0.092
SAPLMA	0.148 ±0.024	0.314 ±0.034	0.201 ±0.029	0.031 ±0.008	0.325 ±0.049	0.057 ±0.013	0.016 ±0.006	0.332 ±0.077	0.031 ±0.012	0.003 ±0.002	0.325 ±0.146	0.007 ±0.005
SAPLMA (tok.)	0.529 ±0.008	0.621 ±0.004	0.571 ±0.006	0.230 ±0.020	0.781 ±0.014	0.355 ±0.025	0.151 ±0.022	0.823 ±0.020	0.254 ±0.032	0.056 ±0.020	0.894 ±0.025	0.105 ±0.035
ICR Probe	0.088 ±0.006	0.215 ±0.011	0.125 ±0.008	0.019 ±0.003	0.230 ±0.030	0.036 ±0.006	0.009 ±0.002	0.211 ±0.039	0.017 ±0.004	0.001 ±0.001	0.116 ±0.067	0.002 ±0.001
ICR (tok.)	0.311 ±0.007	0.491 ±0.006	0.381 ±0.007	0.125 ±0.003	0.661 ±0.005	0.211 ±0.004	0.085 ±0.001	0.726 ±0.003	0.152 ±0.002	0.036 ±0.003	0.853 ±0.009	0.069 ±0.005
HaMI	0.284 ±0.065	0.463 ±0.064	0.351 ±0.069	0.064 ±0.043	0.442 ±0.209	0.111 ±0.073	0.034 ±0.027	0.426 ±0.268	0.062 ±0.049	0.007 ±0.006	0.405 ±0.330	0.014 ±0.012
LLM-Chk (logit)	0.174	0.351	0.233	0.085	0.570	0.148	0.064	0.668	0.117	0.030	0.829	0.058
LLM-Chk (win.)	0.181	0.360	0.241	0.102	0.614	0.176	0.052	0.620	0.096	0.013	0.672	0.025
LLM-Chk (hid.)	0.056	0.148	0.081	0.011	0.151	0.021	0.007	0.169	0.013	0.001	0.136	0.002
LLM-Chk (attn.)	0.113	0.260	0.157	0.022	0.251	0.040	0.008	0.203	0.016	0.000	0.050	0.001
Token logprob	0.171	0.347	0.229	0.084	0.566	0.146	0.059	0.647	0.108	0.031	0.832	0.059

Table 7: THRIVE-G (Gemma-2-2b-it): recall, precision, and F1 across false-positive budgets (test-set thresholds).

Method	Test FPR (target ≤0.010)	Precision	Recall	F1
LatticeSpike (ours)	0.009	**0.825**	**0.269**	**0.406**
Lookback Lens	0.010	0.762	0.197	0.313
Hidden probe	0.011	0.782	0.243	0.371
SAPLMA	0.010	0.322	0.031	0.057
SAPLMA (tok.)	0.009	0.792	0.210	0.331
ICR Probe	0.009	0.224	0.017	0.031
ICR (tok.)	0.008	0.682	0.109	0.187
HaMI	0.009	0.433	0.060	0.105
LLM-Chk (logit)	0.009	0.577	0.083	0.145
LLM-Chk (win.)	0.009	0.624	0.099	0.170
LLM-Chk (hid.)	0.010	0.151	0.011	0.021
LLM-Chk (attn.)	0.011	0.248	0.023	0.042
Token logprob	0.010	0.567	0.083	0.144

Table 8: THRIVE-G (Gemma-2-2b-it): deployed operating point – threshold selected on validation at ≤1% FPR, applied unchanged to test.

B.3 Latency and Memory Usage Testing

Configuration. The generator streams token-by-token at batch size one with a KV cache; each detector consumes only the tensors of the forward pass that predicts the scored token, matching the causal alignment of Appendix D. Latency is the detector's added wall-clock time per step, device-synchronized, reported as the median over the full test set. Work incurred once per response—LatticeSpike's recurrent forward and ICR's attention-map reassembly—is divided by response length and included. Peak memory is measured on each corpus's four longest prompt-plus-response sequences as the increase over bare generation for each detector group: LatticeSpike and the ICR Probe are measured individually, whereas the ten lightweight heads are measured in a single joint pass, so their group value upper-bounds the cost of each member. All measurements ran on the hardware of Section 5.1 with an otherwise idle GPU.

Score equivalence verification. Timings are reported only for a streaming path verified to reproduce the reported scores. Verification has three levels. First, every adapter is replayed on a teacher-forced full pass, the extraction pipeline's exact batch shape, isolating porting error from numerics. Second, streaming per-token scores are compared against one corresponding reported run for each method–corpus pair (one seed for stochastic methods). Seeds change detector weights, not the computation path. Third, AP, AUROC, and the three 1%-FPR metrics computed from streaming scores are compared against the same metrics from the reported scores over the full test pools, with pre-set gates of 10^{-2} for AP, recall, and F1, 2×10^{-3} for AUROC, and 2×10^{-2} for precision. Every method–corpus pair passes every gate, typically agreeing within 2×10^{-3}. Exact equality is unattainable: bf16 execution is not bitwise stable across execution orders, so KV-cache streaming and full-pass extraction exhibit small numerical differences, which tie-sensitive metrics can amplify to the observed 10^{-3} scale.

Latency. Table 18 covers added per-token latency for all thirteen methods on all four corpora, with ICR-tok sharing the ICR Probe's feature-extraction timing. LatticeSpike's cost is dominated by signal construction: the per-layer distribution trace scales with the generator's depth and vocabulary, which produces the 3.6–6.3 ms range across generators.

Method	Test FPR (target $\leq$0.010)	Precision	Recall	F1
LatticeSpike (ours)	0.007	**0.679**	**0.229**	**0.342**
Lookback Lens	0.007	0.669	0.200	0.308
Hidden probe	0.008	0.633	0.218	0.324
SAPLMA	0.012	0.116	0.023	0.038
SAPLMA (tok.)	0.007	0.630	0.185	0.286
ICR Probe	0.010	0.115	0.019	0.032
ICR (tok.)	0.008	0.324	0.059	0.100
HaMI	0.010	0.294	0.062	0.103
LLM-Chk (logit)	0.010	0.141	0.025	0.042
LLM-Chk (win.)	0.010	0.145	0.026	0.044
LLM-Chk (hid.)	0.004	0.068	0.004	0.008
LLM-Chk (attn.)	0.009	0.127	0.020	0.035
Token logprob	0.010	0.076	0.013	0.022

Table 9: RAGTruth (Llama-2-7b-chat): deployed operating point – threshold selected on validation at $\leq$1% FPR, applied unchanged to test.

Method	AP	AUROC	R@1%FPR	P@1%FPR	F1@1%FPR
LatticeSpike (ours)	**0.399** $\pm$**0.008**	**0.888** $\pm$**0.005**	**0.250** $\pm$**0.002**	**0.577** $\pm$**0.002**	**0.349** $\pm$**0.003**
Lookback Lens	0.371	0.884	0.214	0.539	0.307
Hidden probe	0.326	0.859	0.193	0.513	0.280
SAPLMA	0.095 $\pm$0.009	0.696 $\pm$0.021	0.025 $\pm$0.004	0.121 $\pm$0.018	0.042 $\pm$0.007
SAPLMA (tok.)	0.334 $\pm$0.017	0.851 $\pm$0.009	0.204 $\pm$0.012	0.526 $\pm$0.015	0.294 $\pm$0.015
ICR Probe	0.062 $\pm$0.003	0.539 $\pm$0.006	0.020 $\pm$0.006	0.100 $\pm$0.028	0.034 $\pm$0.010
ICR (tok.)	0.210 $\pm$0.002	0.781 $\pm$0.001	0.117 $\pm$0.003	0.390 $\pm$0.007	0.180 $\pm$0.005
HaMI	0.133 $\pm$0.018	0.727 $\pm$0.018	0.059 $\pm$0.014	0.241 $\pm$0.042	0.094 $\pm$0.021
LLM-Chk (logit)	0.168	0.706	0.124	0.403	0.189
LLM-Chk (win.)	0.169	0.707	0.129	0.412	0.196
LLM-Chk (hid.)	0.055	0.573	0.000	0.000	0.000
LLM-Chk (attn.)	0.068	0.591	0.019	0.093	0.031
Token logprob	0.145	0.697	0.096	0.345	0.151

Table 10: TriviaQA (Qwen3-8B, layer 18); 57,762 metric tokens, prevalence 5.17%, random-AP floor 0.052; Cap-hit generations excluded at build (pre-registered).

The recurrent core contributes $\sim$0.5 ms on GPU and 0.15–0.19 ms on CPU, where kernel-launch overhead disappears. Relative overhead ranges from 17% to 37%, with the lowest value observed on Llama-2-7B. Among baselines, the lightweight heads are cheapest at approximately 0.1 ms, although their accuracy varies substantially and the hidden-state probe is among the strongest baselines; Lookback Lens costs 0.9–1.9 ms; LLM-Check's hidden score computes an expanding-prefix SVD, with median cost reaching 1.55 ms on RAGTruth.

Memory. Table 20 reports added peak memory against bare generation peaks of 6.3–27.2 GiB. LatticeSpike adds at most 103 MiB; its detector holds 0.4–0.8M parameters, scaling with the generator's hidden width and vocabulary. The lightweight heads add under 8 MiB jointly. The ICR Probe must retain every layer's attention map and pays for it in one currency or the other: up to 4.7 GiB of additional GPU memory, or, with its buffers parked in host memory on RAGTruth, 56 ms per token — more than the generation step itself.

	FPR $\leq$ 5%			FPR $\leq$ 1%			FPR $\leq$ 0.5%			FPR $\leq$ 0.1%		
Method	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LatticeSpike (ours)	**0.529** $\pm$**0.011**	**0.366** $\pm$**0.005**	**0.433** $\pm$**0.007**	**0.250** $\pm$**0.002**	**0.577** $\pm$**0.002**	**0.349** $\pm$**0.003**	**0.158** $\pm$**0.006**	**0.634** $\pm$**0.009**	**0.254** $\pm$**0.008**	**0.045** $\pm$**0.008**	**0.710** $\pm$**0.041**	**0.085** $\pm$**0.014**
Lookback Lens	0.518	0.361	0.425	0.214	0.539	0.307	0.129	0.586	0.212	0.038	0.679	0.072
Hidden probe	0.468	0.338	0.393	0.193	0.513	0.280	0.112	0.552	0.187	0.024	0.575	0.047
SAPLMA	0.105 $\pm$0.015	0.103 $\pm$0.013	0.104 $\pm$0.014	0.025 $\pm$0.004	0.121 $\pm$0.018	0.042 $\pm$0.007	0.014 $\pm$0.003	0.129 $\pm$0.028	0.025 $\pm$0.006	0.004 $\pm$0.001	0.164 $\pm$0.048	0.007 $\pm$0.002
SAPLMA (tok.)	0.491 $\pm$0.015	0.349 $\pm$0.007	0.408 $\pm$0.010	0.204 $\pm$0.012	0.526 $\pm$0.015	0.294 $\pm$0.015	0.115 $\pm$0.010	0.557 $\pm$0.023	0.191 $\pm$0.016	0.023 $\pm$0.004	0.558 $\pm$0.049	0.045 $\pm$0.008
ICR Probe	0.081 $\pm$0.014	0.081 $\pm$0.013	0.081 $\pm$0.013	0.020 $\pm$0.006	0.100 $\pm$0.028	0.034 $\pm$0.010	0.011 $\pm$0.004	0.109 $\pm$0.034	0.021 $\pm$0.007	0.003 $\pm$0.001	0.130 $\pm$0.047	0.005 $\pm$0.002
ICR (tok.)	0.325 $\pm$0.004	0.262 $\pm$0.002	0.290 $\pm$0.003	0.117 $\pm$0.003	0.390 $\pm$0.007	0.180 $\pm$0.005	0.069 $\pm$0.002	0.430 $\pm$0.008	0.119 $\pm$0.004	0.017 $\pm$0.001	0.485 $\pm$0.014	0.033 $\pm$0.002
HaMI	0.200 $\pm$0.032	0.178 $\pm$0.023	0.189 $\pm$0.027	0.059 $\pm$0.014	0.241 $\pm$0.042	0.094 $\pm$0.021	0.032 $\pm$0.008	0.257 $\pm$0.046	0.057 $\pm$0.014	0.006 $\pm$0.002	0.245 $\pm$0.055	0.012 $\pm$0.003
LLM-Chk (logit)	0.230	0.200	0.214	0.124	0.403	0.189	0.085	0.482	0.145	0.022	0.554	0.043
LLM-Chk (win.)	0.232	0.202	0.216	0.129	0.412	0.196	0.089	0.494	0.151	0.022	0.550	0.042
LLM-Chk (hid.)	0.019	0.021	0.020	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LLM-Chk (attn.)	0.078	0.079	0.079	0.019	0.093	0.031	0.010	0.102	0.019	0.003	0.156	0.007
Token logprob	0.197	0.177	0.187	0.096	0.345	0.151	0.066	0.418	0.113	0.023	0.557	0.044

Table 11: TriviaQA with Qwen3-8B: recall, precision, and F1 across false-positive budgets using test-set thresholds.

Method	Test FPR; target $\leq$0.010	Precision	Recall	F1
LatticeSpike	0.009	**0.588**	**0.235**	**0.336**
Lookback Lens	0.009	0.546	0.199	0.292
Hidden probe	0.008	0.526	0.172	0.260
SAPLMA	0.009	0.122	0.024	0.040
SAPLMA-tok	0.009	0.528	0.192	0.281
ICR Probe	0.010	0.099	0.020	0.033
ICR-tok	0.011	0.382	0.124	0.187
HaMI	0.009	0.245	0.056	0.092
LLM-Chk logit	0.008	0.425	0.109	0.173
LLM-Chk win.	0.008	0.433	0.114	0.181
LLM-Chk hid.	0.021	0.004	0.002	0.002
LLM-Chk attn.	0.011	0.089	0.020	0.033
Token logprob	0.008	0.359	0.082	0.134

Table 12: TriviaQA with Qwen3-8B: deployed operating point using a threshold selected on validation at $\leq$1% FPR and applied unchanged to test.

Table 13: RAGTruth cross-task transfer using Llama-2-7b-chat: AP. LatticeSpike cells are mean$\pm$SD over five seeds; deterministic baselines are reported as single exact values. Train on one task and evaluate on all three. Bold and underlined values rank methods within each source–target cell. Columns are not mutually comparable because their random-AP floors differ: QA 0.073, Sum. 0.045, and D2T 0.062.

		AP			
Method	**Src**	**In-d.**	**$\rightarrow$QA**	**$\rightarrow$Sum**	**$\rightarrow$D2T**
LatticeSpike	QA	**0.562** $\pm$**0.008**	–	**0.137** $\pm$**0.008**	**0.148** $\pm$**0.016**
	Sum.	**0.193** $\pm$**0.010**	**0.411** $\pm$**0.021**	–	**0.235** $\pm$**0.014**
	D2T	**0.429** $\pm$**0.003**	**0.486** $\pm$**0.008**	**0.148** $\pm$**0.009**	–
Hidden-state probe	QA	0.436	–	0.090	0.108
	Sum.	0.140	0.220	–	0.192
	D2T	0.421	0.241	0.088	–
Lookback Lens	QA	0.502	–	0.101	0.126
	Sum.	0.174	0.369	–	0.175
	D2T	0.418	0.357	0.102	–

Table 14: RAGTruth cross-task transfer using Llama-2-7b-chat: AUROC. LatticeSpike cells are mean±SD over five seeds; deterministic baselines are reported as single exact values. Train on one task and evaluate on all three. Bold and underlined values rank methods within each source–target cell. Chance level is 0.5 in every column; columns are mutually comparable.

AUROC					
Method	**Src**	**In-d.**	**→QA**	**→Sum**	**→D2T**
LatticeSpike	QA	**0.895** ±0.004	–	**0.718** ±0.013	**0.679** ±0.017
	Sum.	**0.752** ±0.013	**0.842** ±0.012	–	**0.773** ±0.015
	D2T	**0.882** ±0.002	**0.881** ±0.004	**0.731** ±0.005	–
Hidden-state probe	QA	0.848	–	0.636	0.619
	Sum.	0.705	0.724	–	0.715
	D2T	0.860	0.738	0.607	–
Lookback Lens	QA	0.861	–	0.652	0.659
	Sum.	0.714	0.811	–	0.704
	D2T	0.869	0.798	0.646	–

Table 15: RAGTruth cross-task transfer using Llama-2-7b-chat: R@1%FPR. LatticeSpike cells are mean±SD over five seeds; deterministic baselines are reported as single exact values. Train on one task and evaluate on all three. Bold and underlined values rank methods within each source–target cell. A random scorer attains recall ≈0.01 at the 1% budget regardless of prevalence; columns are mutually comparable.

R@1%FPR					
Method	**Src**	**In-d.**	**→QA**	**→Sum**	**→D2T**
LatticeSpike	QA	**0.337** ±0.016	–	**0.069** ±0.004	**0.079** ±0.014
	Sum.	**0.148** ±0.009	**0.221** ±0.021	–	**0.121** ±0.014
	D2T	0.239 ±0.005	**0.249** ±0.010	**0.082** ±0.011	–
Hidden-state probe	QA	0.231	–	0.052	0.044
	Sum.	0.092	0.095	–	0.100
	D2T	**0.258**	0.102	0.058	–
Lookback Lens	QA	0.289	–	0.060	0.058
	Sum.	0.126	0.187	–	0.088
	D2T	0.247	0.184	0.056	–

Table 16: RAGTruth cross-task transfer using Llama-2-7b-chat: P@1%FPR. LatticeSpike cells are mean±SD over five seeds; deterministic baselines are reported as single exact values. Train on one task and evaluate on all three. Bold and underlined values rank methods within each source–target cell. Columns are not mutually comparable because random precision at a fixed FPR tracks task prevalence: QA 0.073, Sum. 0.045, and D2T 0.062.

P@1%FPR					
Method	**Src**	**In-d.**	**→QA**	**→Sum**	**→D2T**
LatticeSpike	QA	**0.728** ±0.009	–	**0.245** ±0.011	**0.343** ±0.041
	Sum.	**0.410** ±0.015	**0.636** ±0.023	–	**0.445** ±0.029
	D2T	0.614 ±0.005	**0.664** ±0.009	**0.278** ±0.028	–
Hidden-state probe	QA	0.647	–	0.196	0.225
	Sum.	0.303	0.431	–	0.401
	D2T	**0.632**	0.448	0.214	–
Lookback Lens	QA	0.696	–	0.220	0.280
	Sum.	0.372	0.598	–	0.371
	D2T	0.622	0.594	0.208	–

Table 17: RAGTruth cross-task transfer using Llama-2-7b-chat: F1@1%FPR. LatticeSpike cells are mean±SD over five seeds; deterministic baselines are reported as single exact values. Train on one task and evaluate on all three. Bold and underlined values rank methods within each source–target cell. Columns are not mutually comparable because F1 inherits the prevalence dependence of precision.

		F1@1%FPR			
Method	**Src**	**In-d.**	**→QA**	**→Sum**	**→D2T**
LatticeSpike	QA	**0.461** ±0.016	–	**0.108** ±0.006	**0.129** ±0.021
	Sum.	**0.217** ±0.012	**0.328** ±0.026	–	**0.190** ±0.020
	D2T	0.344 ±0.006	**0.362** ±0.012	**0.127** ±0.017	–
Hidden-state probe	QA	0.340	–	0.082	0.073
	Sum.	0.142	0.156	–	0.161
	D2T	**0.366**	0.167	0.091	–
Lookback Lens	QA	0.408	–	0.094	0.097
	Sum.	0.188	0.285	–	0.143
	D2T	0.354	0.281	0.088	–

	Added latency (ms/token, median)			
Method	THRIVE-L	THRIVE-G	TriviaQA	RAGTruth
Generation step (reference)	9.49	20.94	24.55	28.58
LatticeSpike (ours)	3.55	5.29	6.29	4.92
Lookback Lens	0.94	1.42	1.86	1.73
Hidden probe	0.10	0.10	0.10	0.10
SAPLMA	0.09	0.09	0.09	0.09
SAPLMA (tok.)	**0.08**	**0.08**	**0.08**	0.21[a]
ICR Probe	1.36	6.58	4.29	56.11[b]
ICR (tok.)[c]	–	–	–	–
HaMI	**0.08**	**0.08**	**0.08**	**0.08**
LLM-Chk (logit)	0.29	0.31	0.29	0.29
LLM-Chk (win.)	0.14	0.16	0.14	0.13
LLM-Chk (hid.)	0.69	0.29	0.28	1.55
LLM-Chk (attn.)	0.12	0.12	0.13	0.13
Token logprob	0.12	0.15	0.12	0.10

Table 18: Deployment benchmark, latency: per-token cost added to streaming generation with batch size one and a KV cache, using CUDA-synchronized medians over the full test sets. Per-response work — LatticeSpike's recurrent forward and ICR's map reassembly — is amortized per token and included. [a]Measured in the separate verification pass; [b]with attention-row buffers parked in host memory; [c]shares the ICR Probe's features and timing, adding only its final scoring head. Among detectors, lower is better; bold marks the lowest latency and underline marks the second-lowest distinct latency in each column. Tied values receive the same marking.

	Generation	Recurrent core		Total added	
Corpus	step	GPU	CPU	GPU	CPU
THRIVE-L	9.49	0.53	**0.19**	3.55	**3.26**
THRIVE-G	20.94	0.59	**0.15**	5.29	**4.86**
RAGTruth	28.58	0.52	**0.18**	4.92	**3.89**
TriviaQA	24.55	0.56	**0.14**	6.29	**5.69**

Table 19: Detector latency in ms per token with the recurrent core on GPU versus CPU. Signal construction reads the generator's tensors and stays on GPU in both settings.

	Peak GPU memory (MiB)			
	THRIVE-L	THRIVE-G	TriviaQA	RAGTruth
Bare generation peak	6446	9286	16897	27871
+ LatticeSpike	**+0.0**	<u>+4.4</u>	<u>+102.8</u>	**+0.0**
+ ICR Probe	+4716.3	**+2.5**	+1001.9	<u>+1.4</u>[b]
+ ten lightweight heads (jointly)	<u>+0.8</u>	+7.3	**+1.1**	+1.5

Table 20: Deployment benchmark, memory: peak GPU memory added over bare streaming generation, measured on each corpus's four longest prompt-plus-response sequences; values within ± 1 MiB are allocator noise. LatticeSpike has 412K–806K detector parameters, scaling with generator width. [b]RAGTruth ICR parks attention-row buffers in host memory, using $\sim$9 GiB at the longest rows; the cost reappears as the 56 ms/token in the latency table. Lower is better; bold marks the lowest reported increment and underline marks the second-lowest in each column. Differences within ± 1 MiB remain within allocator noise.

B.4 Ablation Studies

B.4.1 Architectural ablations

Dynamics layer	AUPRC	AUROC	F1@1% FPR	P@1% FPR	R@1% FPR
RSNN	**0.406 ± 0.012**	**0.763 ± 0.007**	0.143 ± 0.016	0.627 ± 0.028	0.081 ± 0.010
Transformer without volume	0.399 ± 0.008	0.742 ± 0.005	**0.155 ± 0.015**	**0.647 ± 0.024**	**0.088 ± 0.009**
LSTM with volume	0.395 ± 0.005	0.749 ± 0.005	0.146 ± 0.006	0.633 ± 0.010	0.083 ± 0.004
LSTM without volume	0.390 ± 0.007	0.747 ± 0.004	0.138 ± 0.011	0.618 ± 0.020	0.078 ± 0.007
Transformer with volume	0.385 ± 0.007	0.734 ± 0.006	0.135 ± 0.007	0.614 ± 0.014	0.076 ± 0.004

Table 21: Dynamics-layer ablation on THRIVE-L. Results are means ± sample standard deviations over five random seeds on the test split. Precision, recall, and F1 use the most permissive test-set threshold satisfying an FPR budget of 1%, following the same protocol as the main results. "With volume" appends $\log V_t$ to the replacement's input; "without volume" omits it. Bold marks the best mean and underline marks the second-best mean in each metric.

Box volume. Each replacement is evaluated both with and without access to the box volume V_t. In the former case, $\log V_t$ is appended to the input; the RSNN instead uses V_t to modulate its firing threshold and membrane decay.

Dynamics-layer replacements. This ablation tests whether the volume-modulated RSNN contributes beyond the six input channels themselves. We replace only the dynamics layer with either a one-layer LSTM (Hochreiter & Schmidhuber, 1997; Gers et al., 2000) or a one-block causal Transformer (Vaswani et al., 2017), while holding all upstream components, training choices, and evaluation code fixed. The Transformer is a single pre-layer-normalized causal block with four attention heads, sinusoidal positional encodings, and no dropout.

The RSNN uses V_t to modulate its firing threshold and membrane decay. To separate this mechanism from access to the volume signal itself, each replacement is evaluated once with $\log V_t$ appended to its input, denoted "with volume," and once with the volume omitted, denoted "without volume." Thus, the with-volume comparison matches the available information without imposing the RSNN's mechanism on the replacement architectures.

Parameter matching. We match trainable capacity within the swapped dynamics layer, excluding the encoder and signal construction shared by all variants. The RSNN contains 18,313 dynamics-layer parameters. Selecting the closest feasible width gives an LSTM hidden size of 25 and a Transformer model width of 32; every replacement is within 1.5% of the RSNN parameter count. This control prevents differences in dynamics-layer capacity from explaining the result. Table 22 summarizes the resulting model widths and parameter counts.

Dynamics layer	Input width	State width	Parameters
RSNN	149	64	18,313
LSTM without volume	149	25	18,250
LSTM with volume	150	25	18,350
Transformer without volume	149	32	18,560
Transformer with volume	150	32	18,592

Table 22: Parameter matching for the dynamics-layer ablation. Counts include only the swapped dynamics layer; the encoder and signal-construction components are identical across variants.

Evaluation protocol. All variants use the split and training recipe described in Section 5.1 and Appendix C.2. We report the mean and sample standard deviation over seeds 42–46, without architecture-specific hyperparameter tuning. The RSNN aggregate statistics come from the corresponding five-seed evaluation; the four replacement rows come from the dynamics-layer sweep. As in the main results, fixed-FPR metrics are computed directly from each seed's test-set ranking.

Results at a glance. As shown in Table 21, the default LatticeSpike model with RSNN dynamics attains the highest mean AP and AUROC, exceeding the four replacements by 0.007–0.022 AP and 0.013–0.029 AUROC. Supplying volume as an ordinary input feature does not recover this ranking performance: it yields a small improvement for the LSTM and a decrease for the Transformer. At the test-set 1% FPR operating point, Transformer without volume attains the highest mean precision, recall, and F1. We therefore interpret the ablation as evidence that the RSNN

improves global ranking quality, not as uniform dominance at every operating point. Appendix B.4.1 details the controls, parameter matching, and recall sweep, and presents additional tables.

Detailed results. Table 21 reports the primary results. The RSNN achieves the highest mean AP of 0.406 and AUROC of 0.763. The strongest replacement by AP is Transformer without volume at 0.399, whereas the strongest by AUROC is LSTM with volume at 0.749. By comparison, the four replacement models occupy a relatively narrow range of 0.385–0.399 AP and 0.734–0.749 AUROC. Under this controlled configuration, changing between recurrence and causal self-attention therefore has a smaller effect than replacing either with the proposed spiking dynamics. Access to volume as an input feature is not consistently beneficial and does not close the gap in either ranking metric.

The fixed-FPR sweep in Table 23 provides a more qualified view. The RSNN has the highest mean recall at 5% FPR, whereas Transformer without volume has the highest mean recall at 1%, 0.5%, and 0.1% FPR. Both this sweep and Table 21 use thresholds selected directly on the test ranking; the sweep additionally reports several FPR budgets. Collectively, the results show that the RSNN improves the overall ranking measured by AP and AUROC, but does not dominate at every low-FPR operating point.

Dynamics layer	R@5% FPR	R@1% FPR	R@0.5% FPR	R@0.1% FPR
RSNN	**0.243 ± 0.008**	0.081 ± 0.010	<u>0.051 ± 0.010</u>	0.014 ± 0.003
Transformer without volume	<u>0.229 ± 0.009</u>	**0.088 ± 0.009**	**0.061 ± 0.008**	**0.024 ± 0.004**
LSTM with volume	0.228 ± 0.007	<u>0.083 ± 0.004</u>	<u>0.051 ± 0.002</u>	0.015 ± 0.003
LSTM without volume	0.228 ± 0.008	0.078 ± 0.007	0.047 ± 0.006	0.013 ± 0.003
Transformer with volume	0.223 ± 0.010	0.076 ± 0.004	0.048 ± 0.003	<u>0.017 ± 0.001</u>

Table 23: Recall across test-set FPR budgets, reported as mean ± sample standard deviation over five seeds. Each threshold is the most permissive test-set threshold satisfying the stated FPR budget, following the convention in Appendix E. Bold marks the best mean and underline marks the second-best mean in each column; tied second-best means are both underlined.

Scope. This controlled ablation isolates the dynamics layer on THRIVE-L and does not establish that the same ordering holds on THRIVE-G, RAGTruth, or TriviaQA. The absolute AP advantage over the best replacement is modest, while the AUROC separation is more consistent across seeds. We therefore interpret these results as evidence that the proposed spiking dynamics improve ranking performance under the tested configuration, rather than as a claim that they dominate conventional sequence models at every operating point or on every corpus.

B.4.2 Component Ablation

Each arm disables exactly one mechanism, all else identical to the reported configuration, using seeds 42–46 for each arm. Table 24 reports all five metrics with per-metric differences from the full model. Two details refine the main-text summary. First, removing the box-geometry channels degrades the two ranking metrics by -0.018 AP and -0.019 AUROC while leaving the 1%-FPR operating point nearly intact, with changes of -0.002 to -0.003 within seed noise. The geometric channels chiefly shape the global ranking rather than the extreme top. Second, the concretization channel dominates in every column; its removal alone costs more than both other removals combined, and it is the only removal that also drops the model below the strongest baseline at the 1%-FPR operating point.

	AP		AUROC		R@1% FPR		P@1% FPR		F1@1% FPR	
Variant		Δ		Δ		Δ		Δ		Δ
Full model	**0.406 ±0.012**	—	**0.762 ±0.007**	—	**0.081 ±0.010**	—	**0.627 ±0.028**	—	**0.143 ±0.016**	—
− volume modulation	0.391 ±0.006	−0.016	0.749 ±0.005	−0.013	0.075 ±0.007	−0.006	0.610 ±0.021	−0.017	0.134 ±0.011	−0.010
− box-geometry channels (r, κ, m zeroed)	0.388 ±0.003	−0.018	0.744 ±0.001	−0.019	<u>0.079 ±0.003</u>	−0.002	<u>0.623 ±0.009</u>	−0.003	<u>0.141 ±0.005</u>	−0.003
− concretization channel (m removed)	0.361 ±0.005	−0.046	0.731 ±0.002	−0.032	0.058 ±0.006	−0.023	0.545 ±0.028	−0.082	0.104 ±0.011	−0.039
Hidden-state probe (strongest baseline)	<u>0.394</u>	−0.012	<u>0.761</u>	−0.002	0.066	−0.014	0.582	−0.045	0.119	−0.024

Table 24: Component ablation on the THRIVE-L test split: one mechanism removed per arm from the reported configuration, five seeds (42–46) per arm, mean ± SD; Δ columns are differences of five-seed means from the full model (the probe row's Δ shows its gap for reference). Removing the concretization channel is the largest single loss, and every ablated variant falls below the strongest baseline on AP and AUROC; removing the box-geometry channels leaves the 1%-FPR operating point nearly intact while degrading both ranking metrics. Bold marks the best value and underline marks the second-best value in each metric column.

B.4.3 HYPERPARAMETER ABLATION CONTEXT AND RESULTS

Hyperparameter selection protocol. The screens ran as rounds of parallel arms, each varying one axis against the reference configuration at that stage on a shared split, with checkpoints and thresholds selected on validation and the test split consulted only after selection. Later rounds pre-registered their selection criteria before any arm executed, and the reference configuration stood whenever a criterion went unmet or arms tied. Because single-seed validation excursions repeatedly failed to replicate, every selected arm was confirmed on seeds 43–46 before it entered the reported configuration, and only confirmed arms appear in the reported results.

What had the biggest impact. Table 25 orders the adopted changes by relative effect. The objective dominates. The largest single change of the hyperparameter study was removing the auxiliary geometric losses— containment, volume, reconstruction, and nesting—worth $+54\%$ relative validation AP: an early sweep on the ranking-loss weight had appeared to be its own lever, but halving the auxiliary weights and doubling the ranking weight agree to 0.0001, which identified the two as one mechanism and the auxiliary terms as the drag. The box encoder and spiking network remain in the forward path; only the losses that regularized them are gone. Adopting the standardized smooth-AP estimator followed at $+9\%$. Batch size is the one optimization axis that mattered, and it is corpus-dependent: RAGTruth converges in roughly half the optimizer updates of the THRIVE corpora and prefers fewer, larger updates, yielding $+8\%$ validation AP at batch 10 over the default 4, while THRIVE-L, THRIVE-G, and TriviaQA settled at 2, 2, and 4.

Objective-mix screens. The per-corpus mixing weight α produced the largest gains of the final experiments, and its optimum moves monotonically across corpora: $1 \to 0.75 \to 0.25 \to 0$ for TriviaQA, THRIVE-L, RAGTruth, and THRIVE-G, respectively. Four arms per corpus were screened at the frozen configuration; TriviaQA rejected every mix arm and keeps the pure-AP endpoint, while THRIVE-G moved furthest, to pure pairwise. The pairwise term acts as a mid-ranking regularizer on top of the AP surrogate, and its useful weight grows as corpora shift from open-ended factoid ranking toward grounded, low-prevalence detection.

Change	Corpus	Metric	Before	After	Δ
Auxiliary losses: full weight $\to$ zero	THRIVE-L[a]	VAL AP	0.164	0.251	$+53.7\%$
Standardized smooth-AP estimator	THRIVE-L[a]	TEST AP	0.256	0.280	$+9.5\%$
Batch size $4 \to 10$	RAGTruth	VAL AP	0.466	0.503	$+7.9\%$
Mix α: $1 \to 0.25$	RAGTruth	TEST AP	0.415	0.442	$+6.6\%$
		AUROC	0.840	0.864	$+2.9\%$
Mix α: $0.75 \to 0$	THRIVE-G	TEST AP	0.605	0.639	$+5.5\%$
		AUROC	0.866	0.885	$+2.2\%$
Batch size $4 \to 2$	THRIVE-L	VAL AP	0.398	0.410	$+2.9\%$
Mix α: $1 \to 0.75$	THRIVE-L	TEST AP	0.410	0.419	$+2.3\%$
		AUROC	0.760	0.766	$+0.8\%$

Table 25: Adopted changes ordered by relative effect. Each row varies one axis against the reference configuration at that stage; deltas are relative, as absolute values are not comparable across corpora and metrics. THRIVE-G rows and the THRIVE-L batch row are five-seed means; the rest are seed-42 screening results whose selected arms were subsequently confirmed on seeds 43–46. [a]A preliminary corpus construction used a different judging protocol and had 1.6% token prevalence, so its absolute values are not comparable to the THRIVE-L rows.

B.5 STATISTICAL SIGNIFICANCE TESTING

Design. Each test resamples entire test responses with replacement. Because tokens within a response are correlated, the response is the exchangeable unit. The test recomputes the pooled metric for every method on the shared resample, and compares LatticeSpike's mean over its five seeds against one baseline; seeded baselines are likewise seed-averaged, and deterministic baselines enter as single values. With B=10,000 resamples, let $\Delta_b = M_{\mathrm{LS}}^{(b)} - M_{\mathrm{base}}^{(b)}$ denote the paired metric difference in bootstrap replicate b. The one-sided p-value is estimated as

$$\hat{p} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}[\Delta_b \le 0]}{B + 1}.$$

Benjamini–Hochberg correction at q=0.05 is applied separately to the 60 full-corpus tests and the 90 cross-task tests; under the stricter Holm correction, which controls family-wise error, 31 of 60 and 31 of 90 comparisons remain significant.

Results at a glance. We find that 49 of the 60 full-corpus comparisons and 52 of the 90 cross-task comparisons are significant. The full-corpus family comprises 4 corpora $\times$ 5 metrics $\times$ 3 baselines; the cross-task family comprises 9 source–target pairs $\times$ 5 metrics $\times$ 2 baselines. Among the full-corpus comparisons, AP is significant in 11 of 12 cases, while recall, precision, and F1 at 1% FPR are each significant in 10 of 12 cases. In the cross-task grid, wins are mostly significant against the hidden-state probe across all metrics; AP is significant in 8 of 9 cases and AUROC in all 9. The insignificant comparisons tend to concentrate on the few tight AUROC margins for the full-corpus evaluation, and on the cross-task comparison against Lookback Lens, where only our AP and AUROC are significant at least 50% of the time.

Detailed results. Tables 26 and 27 list the BH-adjusted p-values, and Table 28 summarizes the counts per metric. Three patterns organize the insignificant comparisons. First, the tight AUROC margins: on THRIVE-L, THRIVE-G, and TriviaQA the AUROC gap to the closest competitor is 0.002–0.005, and these cells do not reach significance. Second, THRIVE-L against the probe: the three 1%-FPR metrics fall just above threshold at $p_{\mathrm{BH}} = 0.053$–0.062. Third, the cross-task grid against Lookback Lens: RAGTruth's test split provides 150 responses per task, an order of magnitude fewer than the full-corpus pools, and twelve of these comparisons sit at p_{BH} between 0.05 and 0.09; LatticeSpike's mean nonetheless leads in every out-of-domain cell. Conversely, Table 28 shows the two ranking metrics are near-universal where each matters most: AP is significant in 11 of 12 full-corpus comparisons, and AUROC is significant in 16 of 18 transfer comparisons, including all nine comparisons against the probe. The three 1%-FPR metrics are significant in 10 of 12 full-corpus comparisons each and share the cross-task power limit against Lookback Lens.

Significance over the sweep tables follows the same protocol (Table 29): of the 144 comparisons (4 corpora $\times$ 4 budgets $\times$ 3 metrics $\times$ 3 baselines), 104 are significant. Because recall, precision, and F1 are monotone transforms of one another at a fixed false-positive rate, the three metrics agree within 0.031 on every comparison and disagree on significance only once: on THRIVE-L at the 0.5% budget against the hidden-state probe, where recall and F1 reach $p_{\mathrm{BH}} = 0.049$ and precision 0.064.

As for the architectural ablation, the RSNN's AUROC lead is significant against all four replacements ($p_{\mathrm{BH}} < 10^{-3}$) and its AP lead against three of four, the exception being Transformer $-$volume (Table 30). None of the twelve thresholded comparisons is significant in either direction; the strongest opposing comparison to the RSNN does not clear the threshold even without correction ($p = 0.15$). The spiking dynamics therefore appear to drive ranking quality, while the operating point is indistinguishable across architectures at this sample size.

Regarding the component ablation, all six AP and AUROC comparisons clear correction, five at $p_{\mathrm{BH}} < 10^{-3}$ (Table 31). For thresholded metrics, at the 1% operating point only the concretization removal stays significant (0.004); volume modulation falls just short (0.062–0.076) and the box-geometry channels do not approach it (≈ 0.39). Nine of fifteen comparisons are significant, and the six exceptions confirm rather than complicate the reading in Section 5.3: the concretization channel is what carries the biggest advantage.

C Architecture and Loss Details

C.1 Training Objective and Losses

Ranking-loss constants. The ranking margin is $m = 0.1$. The sigmoid relaxation of the ranking indicator is $\sigma_\tau(x) = 1/(1 + e^{-x/\tau})$, with temperature $\tau = 0.25$ applied to standardized scores as described below. The overall loss weight is $\lambda = 12$. In the released code, α corresponds to the flag `ap-mix`, λ to `lam-rank`, and τ to `ap-tau`.

Pair construction. Ranking pairs are drawn from two sources: pairs of tokens from the same response, and pairs of tokens from different responses in the same batch. The loss uses the union of both sets, and every eligible pair receives equal weight. A token enters a pair only if it is labeled grounded or hallucinated; uncertain tokens are excluded. The first token of each response is also excluded, since no causal prefix exists at that position.

The number of pairs per batch is capped at 4,096. When a batch exceeds the cap, pairs are selected by evenly spaced deterministic sampling of the flattened pair matrix rather than by random sampling. Determinism here matters for one reason: validation losses remain exactly reproducible across runs and cannot vary with the state of the random number generator.

Score standardization. Inside the Smooth-AP loss, scores are standardized before the sigmoid relaxation is applied: the batch mean is subtracted and the result is divided by the batch standard deviation, with gradients flowing through

Corpus	Metric	vs. probe	vs. Lookback	vs. SAPLMA-tok	Sig.
THRIVE-L	AP	0.136	$<10^{-3}$	0.008	†‡
THRIVE-L	AUROC	0.360	$<10^{-3}$	0.002	†‡
THRIVE-L	R@1%	0.053	0.002	0.020	†‡
THRIVE-L	P@1%	0.062	0.002	0.022	†‡
THRIVE-L	F1@1%	0.053	0.002	0.020	†‡
THRIVE-G	AP	0.001	$<10^{-3}$	0.003	•
THRIVE-G	AUROC	0.149	$<10^{-3}$	0.096	†
THRIVE-G	R@1%	0.008	$<10^{-3}$	0.004	•
THRIVE-G	P@1%	0.009	$<10^{-3}$	0.003	•
THRIVE-G	F1@1%	0.008	$<10^{-3}$	0.004	•
RAGTruth	AP	0.001	0.006	$<10^{-3}$	•
RAGTruth	AUROC	$<10^{-3}$	$<10^{-3}$	$<10^{-3}$	•
RAGTruth	R@1%	0.011	0.113	0.013	∗‡
RAGTruth	P@1%	0.011	0.113	0.013	∗‡
RAGTruth	F1@1%	0.011	0.113	0.013	∗‡
TriviaQA	AP	$<10^{-3}$	0.001	$<10^{-3}$	•
TriviaQA	AUROC	$<10^{-3}$	0.152	$<10^{-3}$	∗‡
TriviaQA	R@1%	$<10^{-3}$	0.002	$<10^{-3}$	•
TriviaQA	P@1%	$<10^{-3}$	0.002	$<10^{-3}$	•
TriviaQA	F1@1%	$<10^{-3}$	0.002	$<10^{-3}$	•

Table 26: Full-corpus significance: Benjamini–Hochberg-adjusted one-sided p-values from a paired cluster bootstrap at the response level B=10,000. Each test compares LatticeSpike's five-seed mean with one competitor, with correction within this 60-test family. Markers denote significance at $p_{\mathrm{BH}} < 0.05$: $*$ vs. the probe, † vs. Lookback Lens, ‡ vs. SAPLMA-tok; • replaces the three markers when significant vs. all competitors.

both statistics. The temperature τ then applies to the standardized values, and the self-comparison term $\sigma_\tau(0)$ is subtracted from each positive count.

Standardization closes two failure modes observed during development. With an absolute temperature applied to raw scores, the sigmoid saturates once the score scale grows, and the gradient vanishes. With a temperature proportional to a detached estimate of the score scale, the objective becomes fully scale-invariant, and the model can reduce its loss by inflating the score scale rather than by improving the ranking. Standardizing inside the computation graph removes both escapes. Standardization is a positive affine transformation, so the average precision being approximated is unchanged; only the sharpness of the relaxation is fixed. The pairwise margin loss operates on raw score differences and requires no standardization.

Optimization. All models train with AdamW at learning rate 2×10^{-4}, gradient clipping at norm 1.0, and no weight decay, for at most 15 epochs. Checkpoints are selected by validation average precision, with early stopping after four epochs without improvement. Batch size and the mixing weight α are tuned hyperparameters; their values, together with all corpus-specific settings, are listed in the experimental setup (Section 5.1).

C.2 Architecture and Per-Corpus Configuration

The detector's architectural constants are fixed across all four corpora: a 64-dimensional box space with a linear box encoder that shares the generator's token-embedding table; the six-channel input of Section 3; 64 leaky-integrate-and-fire neurons with an arctangent surrogate gradient, target spike density 0.15, and a spike-rate window of 8 steps; a residual score head with 64 hidden units; and $k = 15$ candidate tokens. The optimization recipe (AdamW, learning rate 2×10^{-4}, gradient clipping, early stopping) is shared as well and given in Appendix C.1. Per-corpus settings, all selected on validation, are listed in Table 32.

D Baseline Fidelity and Causal Adaptations

Every baseline scores token t from the forward pass that predicts it: the prompt prefill for $t = 0$, afterwards the pass that consumed token $t - 1$. The scored token is never visible to its own features. Methods published as response-level

Src→Tgt	vs. probe	vs. Lookback	Sig.
AP (Table 13)			
QA→QA	$< 10^{-3}$	0.032	•
QA→Sum.	0.019	0.052	∗
QA→D2T	$< 10^{-3}$	0.040	•
Sum.→QA	$< 10^{-3}$	0.073	∗
Sum.→Sum.	0.002	0.196	∗
Sum.→D2T	$< 10^{-3}$	0.006	•
D2T→QA	$< 10^{-3}$	$< 10^{-3}$	•
D2T→Sum.	$< 10^{-3}$	0.009	•
D2T→D2T	0.320	0.259	–
AUROC (Table 14)			
QA→QA	$< 10^{-3}$	$< 10^{-3}$	•
QA→Sum.	0.004	0.029	•
QA→D2T	$< 10^{-3}$	0.082	∗
Sum.→QA	$< 10^{-3}$	0.005	•
Sum.→Sum.	0.009	0.037	•
Sum.→D2T	$< 10^{-3}$	$< 10^{-3}$	•
D2T→QA	$< 10^{-3}$	$< 10^{-3}$	•
D2T→Sum.	$< 10^{-3}$	$< 10^{-3}$	•
D2T→D2T	0.002	0.062	∗
R@1%FPR (Table 15)			
QA→QA	0.005	0.071	∗
QA→Sum.	0.151	0.310	–
QA→D2T	0.001	0.046	•
Sum.→QA	$< 10^{-3}$	0.133	∗
Sum.→Sum.	$< 10^{-3}$	0.189	∗
Sum.→D2T	0.029	0.053	∗
D2T→QA	$< 10^{-3}$	0.030	•
D2T→Sum.	0.071	0.053	–
D2T→D2T	0.815	0.724	–
P@1%FPR (Table 16)			
QA→QA	0.005	0.071	∗
QA→Sum.	0.151	0.310	–
QA→D2T	0.002	0.050	∗
Sum.→QA	$< 10^{-3}$	0.139	∗
Sum.→Sum.	$< 10^{-3}$	0.189	∗
Sum.→D2T	0.035	0.053	∗
D2T→QA	$< 10^{-3}$	0.031	•
D2T→Sum.	0.071	0.054	–
D2T→D2T	0.815	0.724	–
F1@1%FPR (Table 17)			
QA→QA	0.005	0.071	∗
QA→Sum.	0.151	0.310	–
QA→D2T	0.002	0.046	•
Sum.→QA	$< 10^{-3}$	0.133	∗
Sum.→Sum.	$< 10^{-3}$	0.189	∗
Sum.→D2T	0.029	0.053	∗
D2T→QA	$< 10^{-3}$	0.030	•
D2T→Sum.	0.071	0.053	–
D2T→D2T	0.815	0.724	–

Table 27: Cross-task significance, as in Table 26 (correction within this 90-test family; two competitors, so • replaces ∗† when significant vs. both). Blocks follow the per-metric cross-task tables. RAGTruth's test split provides 150 responses per task, so confidence intervals are correspondingly wider compared to the full-corpus evaluation.

scores receive the minimal restriction that makes them causal, stated per method below; nothing is tuned on test data. SAPLMA and the ICR Probe are the only baselines published with response-level supervision whose architecture admits token labels unchanged, so they are the only two with token-supervised variants; the probe and Lookback Lens already train on token labels, the LLM-Check scores are training-free, and HaMI's multiple-instance formulation is defined by response-level supervision.

Hidden-state heads. The logistic-regression probe, SAPLMA, and HaMI read the hidden state at the model's middle layer from the predicting pass. SAPLMA keeps its published 256/128/64 feedforward architecture and training recipe; the token-supervised variant replaces response-level supervision with our token labels, all else unchanged. HaMI trains as published, by multiple-instance learning from response-level labels, and already emits per-token scores.

Metric	Full-corpus (of 12)	Cross-task vs. probe (of 9)	Cross-task vs. Lookback (of 9)	Overall (of 30)
AP	11	8	5	24
AUROC	8	9	7	24
R@1%	10	6	2	18
P@1%	10	6	1	17
F1@1%	10	6	2	18

Table 28: Significant comparisons per metric ($p_{\mathrm{BH}} < 0.05$), summarizing Tables 26 and 27. Full-corpus counts pool 4 corpora $\times$ 3 baselines; cross-task counts pool the 9 source–target pairs per baseline.

LLM-Check. The five scores are training-free and published per response. Token log-probability and both entropies are defined at prediction rows and are causal as published. The Hidden Score computes the mean log singular value of the centered token Gram matrix over the expanding causal prefix, the direct restriction of the reference's whole-response computation, introducing no new hyperparameter. The Attention Score reads the log attention diagonal at the prediction row, where the diagonal is attention to the most recent prefix token; the published own-row reading would require the scored token to exist. LLM-Check ships no score orientation, and a wrong sign turns a working detector into a below-chance one, so one sign bit per score is fitted on the training split.

Lookback Lens. The context-versus-generated attention ratio is computed at query row $p + t - 1$ for prompt length p, so the feature for token t predates the token itself; the logistic classifier over layer–head ratios follows the reference.

ICR Probe. ICR features are computed at prediction rows from the replayed forward pass; for a causal model the attention row and hidden state at position r are identical whether r was reached incrementally or in one pass, so the adaptation changes alignment only. The response-supervised probe follows the reference; the token-supervised variant replaces mean-pooled response supervision with our token labels.

D.1 Integration Provenance

All baselines run inside one evaluation harness rather than their release repositories, since every method must consume identical splits, prompts, and causal alignment. Lookback Lens, HaMI, LLM-Check, and the ICR Probe are transcribed from their official code releases and verified line by line against the reference implementations. SAPLMA has no official released implementation (as far as we are aware); thus, ours follows the released paper (Azaria & Mitchell, 2023), which fixes the architecture (256/128/64 with ReLU and a sigmoid output), the optimizer (Adam), and the epoch count (5), so its fidelity rests on paper-level rather than code-level correspondence. The logistic-regression probe is our own control, fit with scikit-learn. No released weights are used anywhere: every trained baseline is retrained on each corpus, and the LLM-Check scores are training-free.

E Metric Definitions

E.1 Token-level metrics

Test-set tokens are pooled across responses. Each retained token i carries a label $y_i \in \{0, 1\}$ (hallucinated $= 1$) and a score s_i; uncertain tokens and each response's first token are removed before pooling. Let n_+ and n_- count positive and negative tokens. Tokens are sorted by decreasing score, and TP_k, FP_k, P_k, R_k denote the true positives, false positives, precision, and recall of the top-k prefix.

Average precision.

$$\mathrm{AP} = \sum_k (R_k - R_{k-1})\, P_k.$$

Step interpolation: only achievable prefix points contribute, with no smoothing between them. The chance level equals the positive prevalence $n_+/(n_+ + n_-)$, stated in each table caption.

AUROC.

$$\mathrm{AUROC} = \frac{1}{n_+ n_-} \sum_{i:\, y_i=1} \sum_{j:\, y_j=0} \Big[\mathbf{1}(s_i > s_j) + \tfrac{1}{2}\, \mathbf{1}(s_i = s_j) \Big],$$

Metric	vs. probe	vs. Lookback	vs. SAPLMA-tok	Sig.
THRIVE-L				
R@5%	0.181	$< 10^{-3}$	0.037	†‡
P@5%	0.182	$< 10^{-3}$	0.037	†‡
F1@5%	0.181	$< 10^{-3}$	0.037	†‡
R@1%	0.061	0.003	0.026	†‡
P@1%	0.070	0.003	0.029	†‡
F1@1%	0.061	0.003	0.026	†‡
R@0.5%	0.049	0.001	0.022	•
P@0.5%	0.064	0.003	0.026	†‡
F1@0.5%	0.049	0.001	0.022	•
R@0.1%	0.132	0.009	0.031	†‡
P@0.1%	0.163	0.012	0.032	†‡
F1@0.1%	0.132	0.009	0.031	†‡
THRIVE-G				
R@5%	0.089	$< 10^{-3}$	0.128	†
P@5%	0.092	$< 10^{-3}$	0.128	†
F1@5%	0.090	$< 10^{-3}$	0.128	†
R@1%	0.012	$< 10^{-3}$	0.006	•
P@1%	0.013	$< 10^{-3}$	0.005	•
F1@1%	0.012	$< 10^{-3}$	0.005	•
R@0.5%	0.035	$< 10^{-3}$	0.007	•
P@0.5%	0.038	$< 10^{-3}$	0.007	•
F1@0.5%	0.035	$< 10^{-3}$	0.007	•
R@0.1%	0.013	0.001	0.035	•
P@0.1%	0.018	0.002	0.032	•
F1@0.1%	0.013	0.001	0.035	•
RAGTruth				
R@5%	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	•
P@5%	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	•
F1@5%	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	•
R@1%	0.014	0.128	0.018	∗‡
P@1%	0.014	0.128	0.017	∗‡
F1@1%	0.014	0.128	0.017	∗‡
R@0.5%	0.064	0.230	0.024	‡
P@0.5%	0.065	0.233	0.023	‡
F1@0.5%	0.064	0.230	0.024	‡
R@0.1%	0.413	0.471	0.132	–
P@0.1%	0.428	0.482	0.132	–
F1@0.1%	0.413	0.471	0.132	–
TriviaQA				
R@5%	$< 10^{-3}$	0.156	$< 10^{-3}$	∗‡
P@5%	$< 10^{-3}$	0.156	$< 10^{-3}$	∗‡
F1@5%	$< 10^{-3}$	0.156	$< 10^{-3}$	∗‡
R@1%	$< 10^{-3}$	0.003	$< 10^{-3}$	•
P@1%	$< 10^{-3}$	0.003	$< 10^{-3}$	•
F1@1%	$< 10^{-3}$	0.003	$< 10^{-3}$	•
R@0.5%	$< 10^{-3}$	0.014	$< 10^{-3}$	•
P@0.5%	$< 10^{-3}$	0.014	$< 10^{-3}$	•
F1@0.5%	$< 10^{-3}$	0.014	$< 10^{-3}$	•
R@0.1%	0.013	0.157	0.005	∗‡
P@0.1%	0.016	0.184	0.005	∗‡
F1@0.1%	0.013	0.157	0.005	∗‡

Table 29: FPR-sweep significance, as in Table 26 (correction within this 144-test family). Rows follow the sweep tables' budgets.

the Mann–Whitney form; tied scores count one half.

Recall. The fraction of hallucinated tokens retrieved, $\mathrm{R} = \mathrm{TP}/n_+$. A continuous score yields one recall value per decision threshold; the thresholds we use are stated below.

Precision. The fraction of flagged tokens that are hallucinated, $\mathrm{P} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$, equal to TP_k/k for the top-k prefix.

F1 score. The harmonic mean $2\mathrm{PR}/(\mathrm{P} + \mathrm{R})$, zero when either factor is zero.

RSNN vs.	AP	AUROC	R@1%	P@1%	F1@1%
LSTM +volume	0.022^*	$<10^{-3*}$	0.766	0.766	0.766
LSTM −volume	0.003^*	$<10^{-3*}$	0.429	0.429	0.429
Transformer +volume	0.022^*	$<10^{-3*}$	0.429	0.431	0.429
Transformer −volume	0.429	$<10^{-3*}$	0.851	0.851	0.851

Table 30: Dynamics-ablation significance: Benjamini–Hochberg-adjusted one-sided p-values for the RSNN against each replacement, paired cluster bootstrap (B=10,000), corrected within this 20-test family. * marks $p_{\mathrm{BH}} < 0.05$. Tested in the opposite direction, no replacement is significantly better than the RSNN on any metric; the strongest such comparison reaches only $p = 0.15$ uncorrected.

Removal	AP	AUROC	R@1%	P@1%	F1@1%
− volume modulation	$<10^{-3*}$	$<10^{-3*}$	0.062	0.076	0.062
− box-geometry channels	0.005^*	$<10^{-3*}$	0.386	0.400	0.386
− concretization channel	$<10^{-3*}$	$<10^{-3*}$	0.004^*	0.004^*	0.004^*

Table 31: Component-ablation significance — Benjamini–Hochberg-adjusted one-sided p-values for the full model against each ablated variant, paired cluster bootstrap (B=10,000), corrected within this 15-test family. * marks $p_{\mathrm{BH}} < 0.05$.

Reported instantiations. The three thresholded metrics appear in different instantiations across the paper. Main tables report them at a 1% false-positive-rate budget: with $\mathrm{FPR}_k = \mathrm{FP}_k/n_-$, the budget selects the largest prefix k^* with $\mathrm{FPR}_{k^*} \leq 0.01$ (a step convention with no interpolation across the tie block; if no prefix qualifies, all three metrics are zero). Appendix sweep tables report recall, precision, and F1 at budgets of 5%, 1%, 0.5%, and 0.1%. Appendix operating-point tables fix the threshold on the validation split at $\leq 1\%$ FPR and apply it unchanged to test, reporting the achieved test FPR alongside precision, recall, and F1. Ablation tables report the same five metrics as the main tables, at times adding a Δ column per metric relative to the full model.

E.2 Latency and peak inference memory

Both quantities are measured in the deployment configuration: the frozen generator streams token-by-token at batch size one with a KV cache, and the detector consumes only the tensors of the forward pass that predicts the scored token. Every measurement is the detector's added cost over the bare generator, which makes the numbers comparable across host models.

Per-token detection latency. The wall-clock time of the detector's per-step feature construction and scoring, recorded with device synchronization after every step; we report the median of all test tokens, alongside the generator's own step time for reference. Work incurred once per response — a recurrent forward over the collected signal sequence, or ICR's attention-map reassembly — is divided by response length and reported as amortized per-token cost.

Peak inference memory. The peak accelerator memory allocated while streaming the most memory-demanding test responses (the four longest prompt-plus-response token sequences), measured once with the generator alone and once per detector group; we report the difference. Detector parameter counts accompany the table.

F HUGGING FACE CHATBOT

To demonstrate the LatticeSpike framework in an interactive setting, we deploy a hallucination-aware chatbot through an anonymous Hugging Face Space. The demonstration integrates Meta Llama 3.2 1B Instruct as the underlying language model with the proposed LatticeSpike RSNN as the hallucination detector. The anonymous deployment allows reviewers to directly interact with the system while preserving author anonymity.

Figure 8 shows the deployed interface. For each user query, the LLM first generates a response, after which LatticeSpike analyzes the internal causal LLM signals at the token level. In the current demonstration, each user message is evaluated independently rather than maintaining detector state across multiple conversational turns. This design pro-

Corpus	Generator	Read layer	Batch	Epochs	α	Params
THRIVE-L	Llama-3.2-1B-Instruct	8	2	15	0.75	412K
THRIVE-G	Gemma-2-2B-it	13	2	15	0	461K
RAGTruth	Llama-2-7B-chat	16	10	15	0.25	805K
TriviaQA	Qwen3-8B	18	4	10	1	806K

Table 32: Per-corpus configuration. Read layer is $\lfloor L/2 \rfloor$ of the generator's L layers, where the box input h_t is read; batch size counts responses; α is the objective-mix weight of Section 3.3; Params is the detector's parameter count, which grows with the generator's hidden width and vocabulary.

vides a direct view of how the detector responds to the internal model dynamics associated with individual generated tokens.

The interface exposes several standard generation parameters to facilitate interactive evaluation. The maximum number of generated tokens can be adjusted between 16 and 128, with a default value of 64. The sampling temperature can be varied between 0 and 1.5, with a default value of 0.7, while nucleus sampling (`top-p`) can be varied between 0.1 and 1.0, with a default value of 0.95. These controls allow the behavior of LatticeSpike to be inspected under different decoding conditions.

For every generated response, the visualization identifies tokens whose LatticeSpike anomaly score exceeds the learned detection threshold. Tokens associated with a hallucination warning are highlighted directly within the generated text, while RSNN burst-spike events are separately indicated by the interface. This provides a correspondence between the generated language, token-level anomaly scores, and the underlying spiking response of the detector.

Figure 9 illustrates an example interaction with the deployed system. The user asks, "How many e's are in 17?", to which the language model generates the response "There are 1 e in 17". The generated statement is incorrect because the character `e` does not occur in the string `17`. LatticeSpike identifies anomalous activity around the tokens responsible for the erroneous statement and returns the verdict **Potential Hallucination Detected**.

For this example, the anomaly detection threshold is 11.8914, while the maximum observed token-level anomaly score is 12.4648. Two of the nine generated tokens exceed the anomaly threshold, and two of the nine tokens are also associated with RSNN burst-spike activity. The maximum observed spike density is $\rho = 0.5215$, compared with the RSNN spike-density threshold of $\rho_{\mathrm{th}} = 0.1500$.

The visualization provides two complementary views of the detection process. The left panel plots the anomaly score associated with each generated token, with the horizontal dashed line representing the decision threshold. Tokens whose anomaly scores exceed this threshold are highlighted, making it possible to localize the detector response within the generated sequence. The right panel displays the distribution of anomaly scores across the generated tokens, together with the same decision threshold. The textual output additionally reports the corresponding flagged token positions.

This example demonstrates the intended use of LatticeSpike as a *token-level warning mechanism*: rather than returning only a single sequence-level hallucination score, the deployed system exposes where elevated anomaly activity occurs during generation and relates these events to the spiking dynamics of the RSNN.

G THRIVE Construction and Validation Details

G.1 Source Composition and Prompt Regimes

Open-prompt sources. The open-prompt portion of THRIVE is constructed from eight complementary factual QA benchmarks: HaluEval (Li et al., 2023), HaluBench (Ravi et al., 2024), PopQA (Mallen et al., 2023), NQ-Open (Lee et al., 2019; Kwiatkowski et al., 2019), SimpleQA-Verified (Haas et al., 2025; Wei et al., 2024), TruthfulQA (Lin et al., 2022), FreshQA (Vu et al., 2024), and HotpotQA (Yang et al., 2018). The frozen prompt snapshot contains 9,970 examples. The sources were selected to cover different factual regimes, including open-domain recall, long-tail entities, misconception-prone questions, temporally sensitive facts, and multi-hop reasoning. We apply source-specific cleaning before generation so that the reference answer itself is as reliable as possible. In particular, HaluEval contributes gold answers rather than intentionally hallucinated alternatives; HaluBench contributes faithful/PASS examples rather than perturbed failures; PopQA questions with conflicting reference answers are removed; SimpleQA-Verified is preferred over the unverified release; and FreshQA examples are retained only while their references remain within the

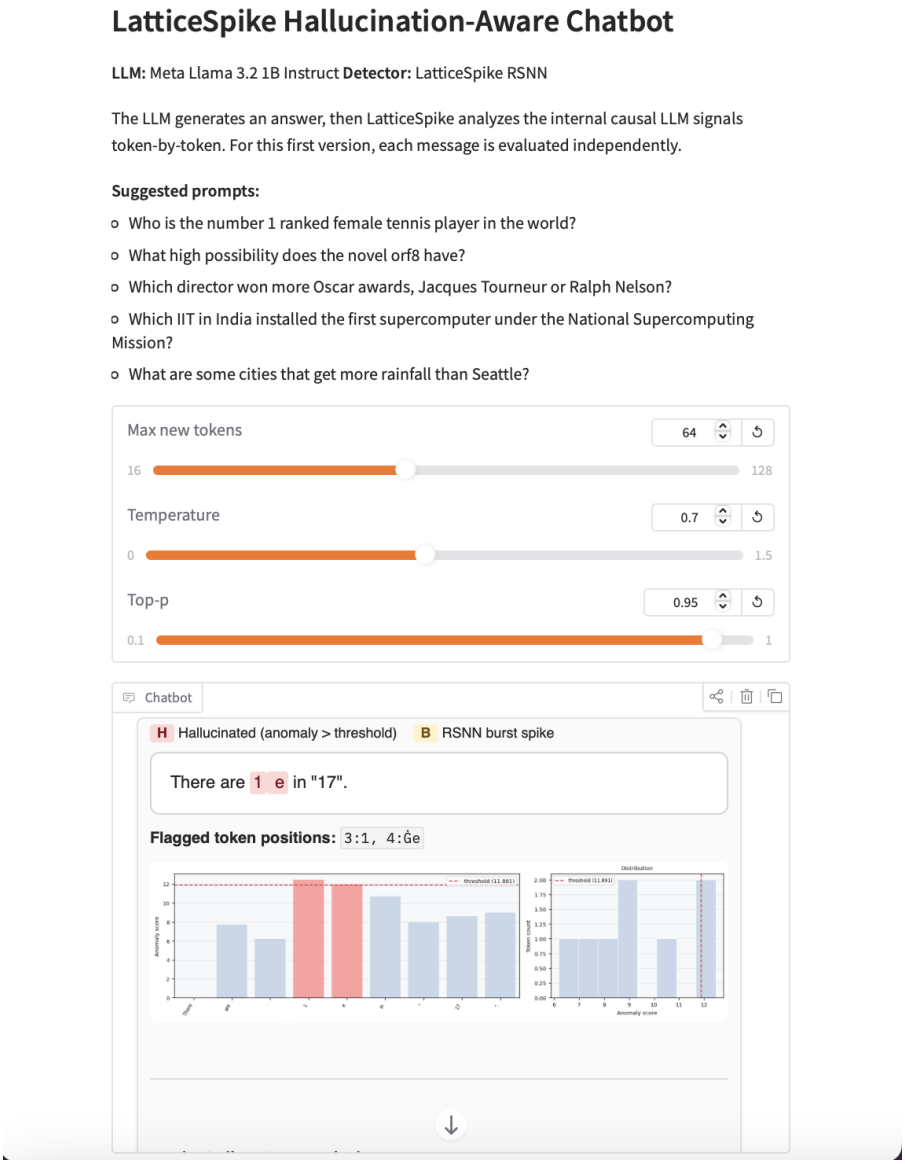


Figure 8: Anonymous Hugging Face deployment of the LatticeSpike hallucination-aware chatbot. The demonstration combines Meta Llama 3.2 1B Instruct with the LatticeSpike RSNN detector. Users can control the maximum generation length, temperature, and top-p sampling parameters and inspect token-level hallucination warnings and RSNN burst-spike activity for generated responses. Each message is analyzed independently in the current deployment.

source-provided review window. These filters are intended to reduce reference noise rather than to make the downstream hallucination task artificially easier.

Answer-forced RAGTruth arm. We construct a separate 981-example arm from RAGTruth questions and retrieved passages. The passage and question are preserved, but the response is regenerated by each THRIVE generator and judged using the THRIVE annotation procedure. We do not transfer RAGTruth's original answer spans because they refer to a different response and are defined primarily in terms of support from the retrieved passage. THRIVE instead evaluates the factual correctness of the newly generated answer, using the available reference and context as evidence. We keep this arm separate from the open-prompt data because its instruction encourages the model to answer rather than abstain, which changes the generation regime and can alter hallucination prevalence independently of model quality.



Figure 9: Example of token-level hallucination detection in the interactive LatticeSpike demonstration. For the query "How many e's are in 17?", the LLM incorrectly generates "There are 1 e in 17". LatticeSpike flags the response as a potential hallucination and localizes elevated anomaly activity to individual generated tokens. The left visualization shows token-level anomaly scores relative to the detection threshold (11.8914), while the right visualization shows the distribution of anomaly scores. In this example, the maximum anomaly score is 12.4648, 2/9 tokens exceed the anomaly threshold, 2/9 exhibit RSNN burst-spike activity, and the maximum spike density is $\rho = 0.5215$ compared with a threshold of $\rho_{\text{th}} = 0.1500$.

G.2 Generation and Token Alignment

Matched generators. THRIVE is instantiated with two generators: Llama-3.2-1B-Instruct (THRIVE-L) and Gemma-2-2B-it (THRIVE-G). Both models receive the same prompt population so that differences in hallucination behavior are not confounded with differences in source composition. This matched construction also enables cross-generator experiments in which a detector trained on one generator is evaluated on the other, providing a stronger test of whether the learned signal reflects hallucination rather than model-specific style.

Generation protocol. Answers are generated greedily with a maximum of 512 new tokens. The relatively permissive cap avoids systematically truncating longer responses, which are particularly important for studying the onset of hallucination over time. For contextual examples, the passage is truncated before the question and generation cue, preserving the part of the prompt required to answer correctly. We retain the generated text, token sequence, token-level log-probabilities, character alignment, finish reason, and generation provenance. These quantities are recorded for analysis and reproducibility; factuality labels are not derived from model confidence.

Native token coordinate system. All supervision is defined on the exact token sequence emitted by the generator. Token text is reconstructed from the generated sequence rather than by independently decoding each subword, which avoids errors when visible characters span byte-level tokens. The final assembly verifies that token counts, token positions, and the stored answer are mutually consistent and that the original generation is not changed by annotation or recovery. This invariant is important because even a small mismatch between the stored answer and the generator's token grid would shift token-level labels and the estimated hallucination onset.

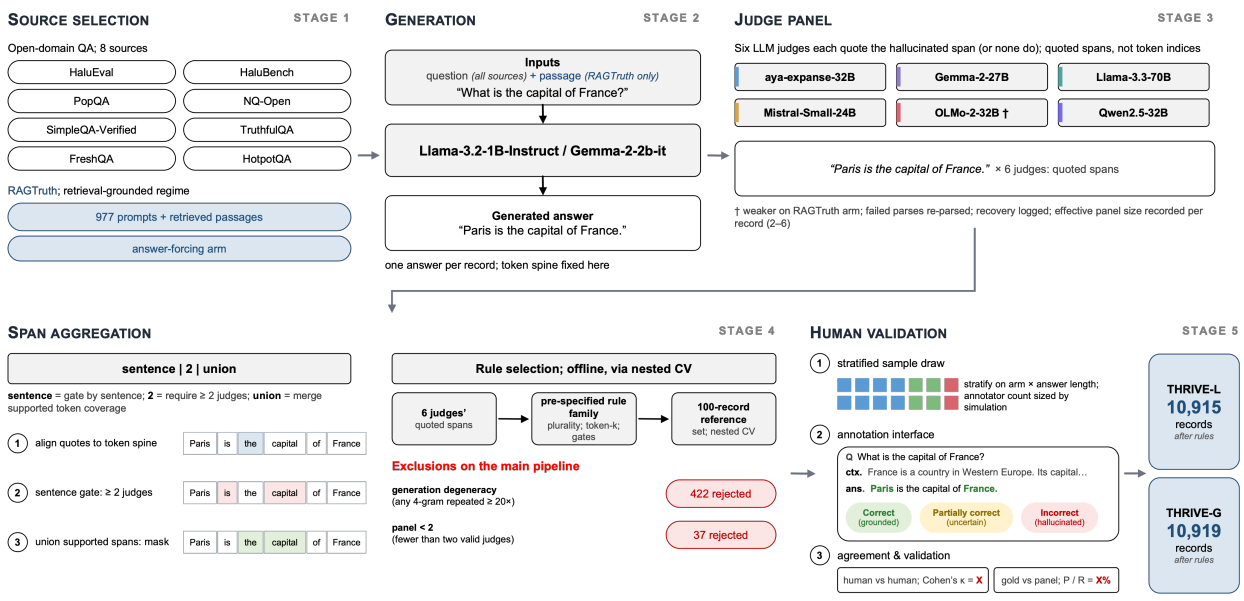


Figure 10: Overview of the THRIVE construction pipeline.

G.3 Inter-model Annotation

Judge panel. Each generated answer is evaluated by a heterogeneous panel consisting of Aya-Expanse-32B (Dang et al., 2024), Gemma-2-27B-IT (Gemma Team et al., 2024), Llama-3.3-70B-Instruct (Grattafiori et al., 2024), Mistral-Small-24B-Instruct-2501 (Mistral AI, 2025), OLMo-2-32B-Instruct (Team OLMo et al., 2025), and Qwen2.5-32B-Instruct (Qwen Team et al., 2024). The panel intentionally spans several model families rather than repeatedly sampling one teacher model, reducing dependence on a single model's factual priors or annotation style. Judging is performed in BF16 for the 24–32B models. Llama-3.3-70B is loaded in NF4 because full-precision inference does not fit on a single 80 GB accelerator; this difference is treated as an implementation constraint and recorded as part of the experimental provenance.

Quote-based annotation. Judges are asked to identify minimal verbatim spans corresponding to false claims or explicit uncertainty instead of returning token indices. This design follows an observed failure mode in earlier versions of the pipeline: judges could agree on the same factual error yet select slightly different subword boundaries, causing token-level voting to suppress otherwise consistent semantic judgments. By asking for text spans, the language model performs the semantic decision while deterministic alignment code performs the token bookkeeping. Repeated phrases may be disambiguated by occurrence. We deliberately avoid fuzzy semantic matching when the returned phrase is not present in the answer, because doing so would create token supervision that the judge did not explicitly provide.

Label semantics. The three labels distinguish grounded content, hallucinated content, and explicit uncertainty or abstention expressed by the generated answer. The uncertainty label is not used merely because a judge is unsure whether a claim is correct. Similarly, a claim is not treated as hallucinated simply because it is absent from a short reference answer. This distinction is important for open-domain QA, where a correct response can contain true information beyond the supplied reference.

G.4 Aggregation, Recovery, and Exclusions

Consensus aggregation. The default aggregation operates at the sentence level before assigning token spans. A sentence is treated as hallucination-positive when at least two resolved judges identify hallucinated content within it; the supported spans from those judges are then unioned inside that sentence. The rationale is that judges are generally more consistent about whether a proposition is false than about the exact subword boundaries that express it. Requiring agreement at the sentence level therefore suppresses isolated judge errors without demanding identical token-level selections. A stricter span-oriented aggregation and the earlier token-level plurality rule are retained for

Statistic	THRIVE-L	THRIVE-G
Pre-filter records	10,951	10,951
Final records	10,915	10,919
Open-prompt arm	9,938	9,939
Answer-forced arm	977	980
Generated answer tokens	898,586	242,160
Records containing H	4,698 (43.0%)	4,470 (40.9%)
Records containing U	2,809 (25.7%)	569 (5.2%)
Generation-degeneracy exclusions	35	29
< 2 resolving-judge exclusions	1	3
Total excluded	36	32

Table 33: Final THRIVE release statistics. The two arms are constructed from the same prompt matrix but filtered independently. Record-level H and U counts describe the released automatic labels and should not be interpreted as human-estimated corpus prevalence.

ablation and sensitivity analysis, but they represent alternative views of the same underlying judge outputs rather than additional annotations.

Failed judgments and recovery. A judge output that cannot be parsed or aligned is treated as a missing vote, not as evidence that the answer is grounded. Failed outputs are first retried with the same model and prompt. When a judge has returned a verbatim phrase that occurs multiple times in the answer, the remaining ambiguity can be resolved deterministically by applying the same placement policy to every exact occurrence. By contrast, genuinely paraphrased responses remain unresolved rather than being force-matched. This policy favors traceable missingness over introducing labels that were never explicitly produced by the judge.

Deterministic exclusions. We remove only failure modes that can be expressed as reproducible rules. Responses exhibiting severe generation loops are excluded when the same 4-gram repeats at least 20 times. We also remove records with fewer than two resolving judges because they cannot satisfy the default consensus rule; retaining them would systematically convert annotation failure into an all-grounded label. In the finalized THRIVE-L construction, these criteria remove 36 of 10,951 generated records, leaving 10,915 examples. The same rules are applied independently to THRIVE-G rather than copying the Llama exclusions, since degeneration and judge resolution can differ across generators.

Corpus Scale and Release Statistics. Table 33 summarizes the finalized release after applying the deterministic exclusion rules described. The two generators begin from the same 10,951-prompt matrix but are filtered independently because degeneration and judge resolution can differ by generator.

Panel depth and provenance. Most released records are supported by all six judges: 9,498 THRIVE-L and 9,641 THRIVE-G records have six resolving judges, while 1,250 and 1,179, respectively, have five. Only 25 THRIVE-L and 18 THRIVE-G records have fewer than four resolving judges. Model revisions are pinned, and judge decoding is greedy. The three aggregation rules used in the release were fixed on a separate 100-record development sample before construction of the finalized corpora and were then applied unchanged to both generator arms.

G.5 Human Validation and Statistical Analysis

G.5.1 Sampling and Annotation Design

Sampling rationale. Human validation is conducted separately for THRIVE-L and THRIVE-G because the two generators can differ in both hallucination prevalence and span characteristics. Two annotators independently label each sampled record using the same annotation guide, with automatic panel labels hidden throughout review. The Llama arm contains 334 reviewed records and the Gemma arm 339. Sampling is stratified jointly by prompt regime and answer length rather than drawn uniformly, ensuring adequate representation of long answers and the answer-forced regime. The samples were designed to estimate corpus-level hallucination prevalence to roughly ± 5–6 percentage points after weighting strata back to their population frequencies.

Inter-annotator reference. Token-level hallucination annotation is not deterministic even for humans: annotators may agree that a response is wrong but select slightly different boundaries for the offending phrase. We therefore report

Arm / level	κ	P	R	F1
THRIVE-L / record	0.886	93.2	94.9	94.0
THRIVE-L / token	0.574	65.4	68.0	66.7
THRIVE-G / record	0.803	87.1	87.8	87.4
THRIVE-G / token	0.634	74.5	64.1	68.9

Table 34: Human–human agreement. Precision, recall, and F1 are percentages. These values provide an empirical inter-annotator reference for interpreting automatic panel performance; they are not assumed to be a strict performance ceiling.

Arm	Human ref.	Record level				Token level			
		κ	P	R	F1	κ	P	R	F1
THRIVE-L	A	0.697	78.3	93.7	85.3	0.373	53.4	46.4	49.6
THRIVE-L	B	0.619	75.7	88.8	81.7	0.285	47.1	39.3	42.9
THRIVE-G	GA	0.667	74.1	86.2	79.7	0.278	49.5	28.8	36.4
THRIVE-G	GB	0.711	76.9	88.7	82.4	0.357	53.3	36.0	43.0

Table 35: Automatic panel performance against each human annotator independently. Precision, recall, and F1 are percentages. Record-level agreement is substantially stronger than token-level agreement, consistent with the greater ambiguity of exact span localization.

human–human agreement as an empirical inter-annotator reference rather than as a strict upper bound on automatic performance. This is particularly important for THRIVE because its intended use includes token-level monitoring and hallucination onset, where exact boundary agreement is stricter than record-level factuality.

Human agreement is substantially stronger at the record level than at the token level in both generators (Table 34). This pattern empirically motivates THRIVE's use of sentence-level semantic agreement before assigning token spans: deciding whether a proposition is hallucinated is more reproducible than selecting its exact subword boundaries.

G.5.2 Panel Agreement with Human Annotators

Evaluation against independent annotators. We score the automatic panel separately against each annotator instead of collapsing the two annotators into a single synthetic gold label before evaluation. This avoids hiding genuine human disagreement and exposes the remaining gap between record-level detection and exact span localization.

The automatic panel approaches human–human agreement more closely at the record level than at the token level. On THRIVE-L, the best panel–human token F1 is 49.6%, compared with 66.7% between the two humans; on THRIVE-G the corresponding values are 43.0% and 68.9%. We report these values directly rather than normalizing by human–human agreement, since inter-annotator agreement is an empirical reference rather than a mathematical upper bound.

G.5.3 Hallucination Onset

Agreement on $t^\star$. We also measure whether annotators identify the same first hallucinated token. Exact $t^\star$ agreement is 78/150 (52.0%) for THRIVE-L and 83/108 (76.9%) for THRIVE-G after advancing both annotations past leading markup. Exact onset agreement is intentionally strict: two annotators can identify the same false claim while differing by one or two tokens at its left boundary. In the released automatic labels, 25.7% of panel-derived $t^\star$ positions fall on leading markup such as an opening bold marker or quotation mark, typically immediately before the substantive claim; onset comparisons should therefore use the same normalization rule and should be interpreted jointly with token-level agreement.

G.5.4 Corpus Hallucination Prevalence

Post-stratified estimation. Because the human-review samples are disproportionately allocated across prompt-regime and answer-length strata by design, their raw positive fractions are not unbiased estimates of corpus prevalence. We therefore weight each stratum back to its population frequency and apply the finite-population correction when reporting uncertainty.

Arm	Annotator	Hallucination rate
THRIVE-L	A	38.7% ± 5.6
THRIVE-L	B	39.4% ± 5.5
THRIVE-G	GA	39.2% ± 5.6
THRIVE-G	GB	38.9% ± 5.6

Table 36: Post-stratified, finite-population-corrected estimates of the share of records containing at least one human-identified hallucinated token. The estimates are descriptive of THRIVE's prompt mixture and generation policy, not universal hallucination rates for the underlying models.

The four estimates are tightly aligned, supporting the narrow claim that roughly four in ten responses contain a human-identified hallucination under the THRIVE prompt mixture. They should not be generalized beyond this mixture of sources, prompt regimes, generation policy, and exclusion rules.

G.5.5 Token Scoring Protocol

Formatting-neutral token evaluation. Token-level scoring excludes positions consisting only of markup or punctuation on both the human and automatic sides. This masking is necessary because quote-to-token alignment can sweep adjacent Markdown markers, quotation marks, or terminal punctuation into an otherwise valid semantic span. Roughly 20% of panel-labeled hallucinated positions in the validation analysis are such formatting-only tokens; scoring them as factual disagreements lowers token precision by approximately 21 points. The mask excludes these positions from evaluation rather than relabeling them and is applied symmetrically to human and automatic annotations.

G.5.6 Uncertainty Labels and Known Disagreement Mode

Uncertainty prevalence. The released automatic labels contain explicit uncertainty or abstention (U) in 2,809 THRIVE-L records (25.7%) and 569 THRIVE-G records (5.2%). This difference is descriptive of the generated corpora and should not be interpreted as an intrinsic uncertainty rate of either base model.

Hallucination–uncertainty boundary. A systematic disagreement remains for responses that decline to answer by making a false claim about an external source, for example asserting that a census does not contain information that is in fact available. The automatic panel marks such cases as U on approximately 83% of a roughly 162-record diagnostic subset, whereas a strict reading of the human labeling guide assigns H because the answer makes a false factual claim about the world. This differs from a statement that the supplied passage does not contain the information, which can correctly constitute abstention. We therefore treat the H–U boundary as a known limitation of the current automatic labels rather than claiming equivalent validation strength for all three classes.

G.5.7 Additional Construction Robustness

Judge resolution. The overwhelming majority of records are judged by five or six resolving models, but resolution is not perfectly uniform across judges and regimes. In particular, OLMo resolves only 54% of the THRIVE-G answer-forced arm. These failures are treated as missing votes rather than grounded judgments, and the released record-level judge depth is retained to support sensitivity analyses.

Human-review asymmetry. Both human-review passes were blind to automatic labels, but the THRIVE-L annotation pass included a self-review against the labeling guide before finalization whereas the THRIVE-G pass did not. This procedural difference is a limitation when comparing panel–human token agreement across generator arms and may contribute to the lower THRIVE-G localization scores.

Interpretation. Together, the validation results support two distinct conclusions. First, THRIVE's automatic labels are substantially more reliable for record-level hallucination detection than for exact token localization, mirroring the human agreement pattern. Second, token-span and onset labels should be treated as fine-grained weak supervision with independently quantified annotation noise rather than as exact human-gold boundaries. This is why the benchmark reports record-, token-, and onset-level behavior separately.

Source	Citation	Records	License
HaluEval	(Li et al., 2023)	4,399	Apache-2.0
NQ-Open	(Lee et al., 2019; Kwiatkowski et al., 2019)	3,365	CC BY-SA 3.0
PopQA	(Mallen et al., 2023)	2,995	MIT
HotpotQA	(Yang et al., 2018)	2,402	CC BY-SA 4.0
HaluBench	(Ravi et al., 2024)	1,993	CC BY-NC-2.0
RAGTruth	(Niu et al., 2024)	1,957	MIT
SimpleQA-Verified	(Haas et al., 2025; Wei et al., 2024)	1,991	MIT
TruthfulQA	(Lin et al., 2022)	1,596	Apache-2.0
FreshQA	(Vu et al., 2024)	1,136	Apache-2.0
Total		21,834	collection

Table 37: THRIVE source composition across both arms, with the license under which each source's material is redistributed.

G.5.8 LICENSING AND REDISTRIBUTION

Licensing and redistribution. THRIVE is released as a collection. Source questions, reference answers, and grounding passages are reproduced verbatim, and each record remains governed by the license of the benchmark named in its `source` field (Table 37). We license only the layer we produced (which we refer to as the annotation layer), comprising the generated answers, the panel labels, the span alignment, and the generation and judging metadata, under CC BY-SA 4.0. Because the HaluBench-derived records carry a CC BY-NC-2.0 license, the corpus as a whole is not available for commercial use; users requiring commercial terms must exclude records whose `License` is `halubench`.