SPILLED ENERGY IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We reinterpret the final softmax classifier over the vocabulary of Large Language Models (LLM) as an Energy-based Model (EBM). This allows us to decompose the chain of probabilities used in sequence-to-sequence modeling as multiple EBMs that interact together at inference time. Our decomposition offers a principled approach to measuring where the "energy spills" in LLM decoding, empirically showing that spilled energy correlates well with factual errors, inaccuracies, biases, and failures. Similar to Orgad et al. (2025), we localize the "exact" token associated with the answer, yet, unlike them, who need to train a classifier and ablate which activations to feed to it, we propose a method to detect hallucinations *completely training-free that naturally generalizes across tasks and LLMs* by using the output logits across subsequent generation steps. We propose two ways to detect hallucinations: the first one that measures the difference between two quantities that we call **spilled energy**, measuring the difference between energy values across two generation steps that mathematically should be equal; the other is **marginal energy**, which we can measure at a single step. Unlike prior work, our method is training-free, mathematically principled, and demonstrates strong cross-dataset generalization: we scale our analysis to state-of-the-art LLMs, including LLaMa-3, Mistral, and Qwen-3, evaluating on nine benchmarks and achieving competitive performance with robust results across datasets and different LLMs.

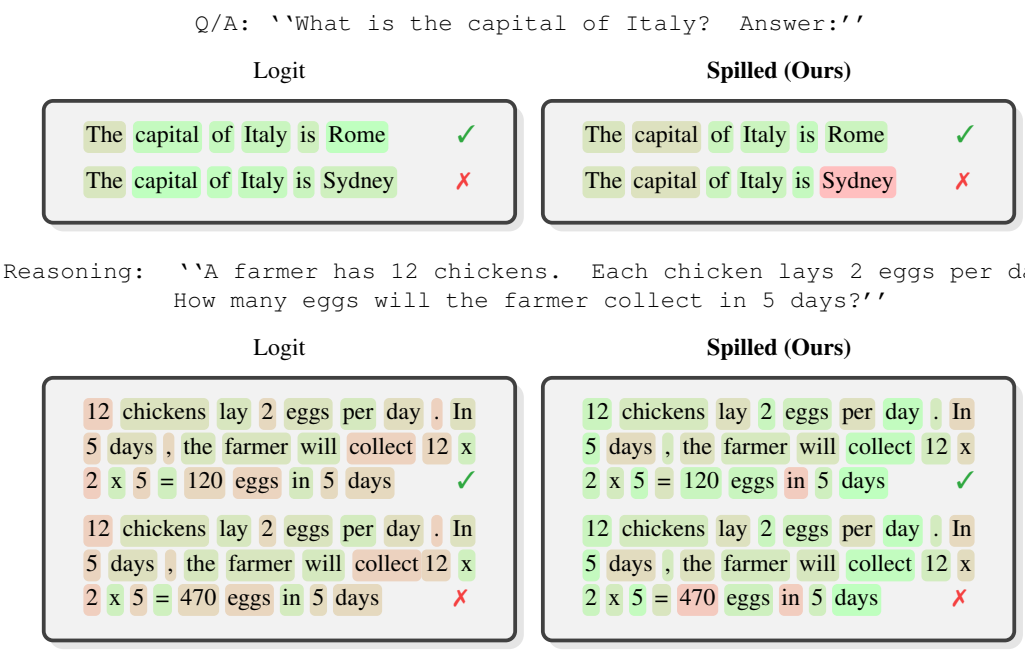


Figure 1: Color-coded comparison of hallucination detection with LLaMa-3 8B using logit confidence and **our spilled energy**. Our method generalizes well across topics (e.g., Q&A, reasoning) and diverse LLMs. ✓ indicates a correct answer and ✗ an incorrect one. While our approach focuses on the exact answer tokens (e.g. Rome/Sydney and 120/470, see Section 4.2), here we apply min–max normalization to the full answer for visualization, as truthful hallucination.

1 INTRODUCTION

The widespread adoption of Large Language Models (LLMs) across various domains has brought increasing attention to their critical limitation: their tendency to generate incorrect or misleading information—commonly referred to as "hallucinations." This issue supports the idea that LLMs are just stochastic parrots (Bender et al., 2021) answering in a way that is statistically plausible with respect to the input prompt despite not having a real understanding of it. On the other side, recent reasoning capabilities proper to ChatGPT 4o (OpenAI-Team, 2023) or Deepseek (Liu et al., 2024) offer counter evidence to actually support this. Ongoing research seeks to characterize and categorize hallucinations, setting them apart from other error types (Liu et al., 2022; Ji et al., 2023; Huang et al., 2023b; Rawte et al., 2023). At the same time, recent discussions have introduced terms such as confabulations (Millidge, 2023) and fabrications (McGowan et al., 2023), sometimes attributing a form of "intention" to LLMs—though the very idea of LLM "intentionality" and other human-like qualities remains contested (Salles et al., 2020; Serapio-García et al., 2023; Harnad, 2024). Research on LLM hallucinations can be categorized into two main branches: the first one is the extrinsic branch, where the hallucinations are measured with respect to the interpretation that humans give to those errors (Bang et al., 2023; Ji et al., 2023; Huang et al., 2023b; Rawte et al., 2023). The second branch was started by Kadavath et al. (2022b), proposing to study the hallucinations *within* the model itself. Following Kadavath et al. (2022b), the work in Li et al. (2024) proposes Inference-Time Intervention (ITI) as a way to improve the "truthfulness" of LLMs at inference time. ITI functions by altering model activations at inference time, steering them along specific directions within a restricted set of attention heads. Our work is also different from Yin et al. (2023), since we care about detecting errors in LLMs, whereas they introduce an automated methodology to detect when LLMs are aware that they do not know how to answer.

In this work, we follow the definition of hallucinations given by Orgad et al. (2025) as any form of error produced by an LLM—including factual mistakes, biased outputs, breakdowns in common-sense reasoning, and related issues. Like them, we also confirm that the truthfulness signal is concentrated in the "exact answer tokens." Nevertheless, unlike them, we abandon the idea of using a probe classifier (Belinkov, 2022) trained for each task and dataset. Given that LLMs are foundational models, user interactions typically occur *in the wild*, making it difficult to predict which probe classifier is best suited for detecting hallucinations in real-world scenarios. Furthermore, in this setting, classifier weights should not only be updated dynamically for each task, but the optimal token–layer combination is also dataset-dependent, which conflicts with the broad LLM applicability. Indeed, in the work by Orgad et al. (2025), the article reports:

> "We find that probing classifiers do not generalize across different tasks."

In our paper, we propose to solve this problem with a training-free method that generalizes better across different tasks and is mathematically principled using the framework of Energy-based Models (EBMs). Fig. 1 reports a qualitative comparison across tasks, comparing to the logit confidence. Additional samples are shown in Appendix D.4.

We reinterpret the final softmax classifier over the vocabulary of LLM as an EBM, taking inspiration from what Grathwohl et al. (2020) did five years ago for classifiers. This perspective enables us to decompose the sequence-to-sequence probability chain into multiple interacting EBMs that operate jointly during inference. Through this decomposition, we introduce the notion of "spilled energy" in LLM decoding and show empirically that such spill strongly correlates with errors. Given that our method is solely based on the mathematics of EBMs and the chain rule of probability, we do not have to train or tune our detector, striking a good generalization across tasks and LLMs. Building on this foundation, our contributions are as follows:

- ◇ Training-free, LLM hallucination detection generalizing across tasks using the EBM framework. We introduce a method for detecting hallucinations that requires no additional training, in contrast to prior work that relies on trained classifiers and ablations of model activations. Our approach directly reads values inside the LLM, enabling natural generalization across tasks and performing better than logit-based detection.
- ◇ Two energy-based metrics. We define two complementary measures of energy spills: (i) delta energy $\Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$, which captures discrepancies between energy values across two time steps that

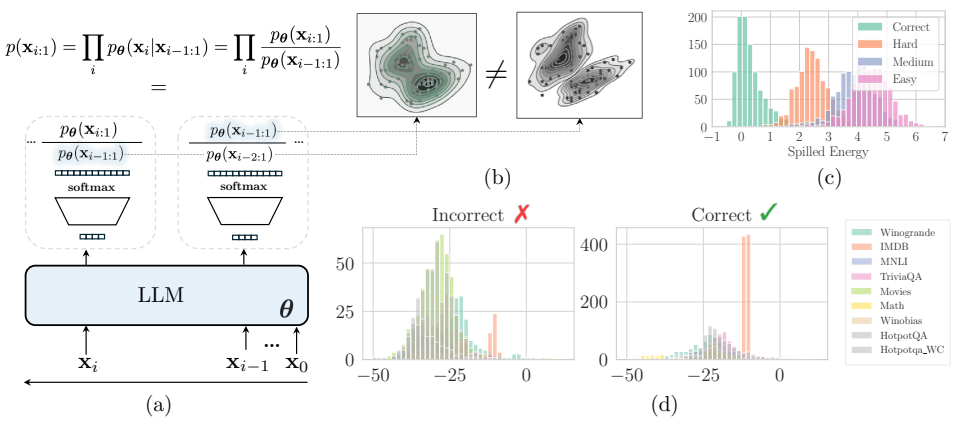


Figure 2: **How energy spills in LLMs**. (a) Language Modeling $p(\mathbf{x}_{i:1})$ is attained as a decomposition problem following the chain rule of probability, implemented as autoregressive: we recursively apply a discriminative classifier over the vocabulary $\mathcal{V}$ to attain generative modeling with larger context size i.e. $p(\mathbf{x}_i|\mathbf{x}_{i-1:1})$. (b) We reinterpret each discriminative classifier as a generative EBM, finding a connection between two quantities that should be the same across time steps yet are different. We call this difference "the spilled energy" $\Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$ in Eq. (8). (c) Given that we simply read values inside the LLM, our approach is training-free and correlates well with hallucinations on a synthetic math dataset with increasing difficulty; (d) histograms of spilled energy values, for incorrect and correct answers on all nine datasets using $\min$ pooling for Llama-3-Instruct. The two distributions are easily separable by using a simple threshold, resulting in a generalization across real-world tasks.

should be mathematically equivalent, and (ii) marginal energy $E^m_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$, which can be evaluated at a single time step.

- ◇ Scalable and generalizable analysis. Our framework is mathematically principled, training-free, and exhibits strong cross-dataset generalization. We scale our analysis to state-of-the-art LLMs, including Llama 3-8B-Instruct and Mistral-7B-Instruct, and demonstrate competitive performance across nine benchmarks, showing robustness across datasets and architectures.

Fig. 2(a) illustrates the core idea of our method: rather than using a naïve approach, such as simply recording the logit or training a probe classifier at the activations of the answer token, we first reinterpret the LLM as an autoregressive EBM via the chain rule of probabilities. We then further decompose each conditional probability, incorporating insights from Grathwohl et al. (2020). At the time step of the exact token $i-1$, we extract the energy, which corresponds to the logit, and compare it with the marginal energy at the next time step i, corresponding to the denominator of the softmax. According to the chain rule, these two quantities should be identical; however, they differ in the LLM implementation—Fig. 2(b). We find that the discrepancy, which we term spilled energy $\Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$, correlates strongly with instances where the LLM produces an incorrect output—see Fig. 2(c). Moreover, its detection signal separates well correct and incorrect classes across datasets, reflecting the model's confidence, as shown in Fig. 2(d).

2 Related Work

EBM applications to Trustworthy AI. EBMs have been applied to improve the reliability and interpretability of Deep Nets. For example, Energy-Based Out-of-Distribution Detection (OOD) (Liu et al., 2020) uses the energy score as a more robust alternative to the softmax confidence. At the same time, Grathwohl et al. (2020) presents how to reinterpret a discriminative classifier as EBM to train models both discriminative and generative. Following this work, Zhu et al. (2021) gives new insights into the role of energy when training EBMs and robust classifiers using adversarial training. Instead, Mirza et al. (2024; 2025) explain adversarial attacks by reinterpreting the softmax classifier as an EBM, showing that these perturbations correspond to shifts in the underlying energy landscape.

Foundations of Hallucination in LLMs. LLMs are prone to diverse errors—including bias, reasoning failures, and generation of factually incorrect information unsupported by reliable sources. Karpowicz (2025) frames hallucination and imagination as mathematically identical phenomena, both emerging from a necessary violation of information conservation. Also Xu et al. (2025) provides a formal learning-theoretic proof that hallucinations are unavoidable. They define a *formal world* in which both the LLM and the ground-truth are computable functions, showing through classic results in computability theory, that no LLM can learn all such functions. As a consequence, hallucination is not just a practical artifact but a fundamental limitation of LLMs, valid even under idealized conditions. Recently Kalai et al. (2025) showed that hallucinations come from the statistical problem of the pretraining methodology: minimizing the cross entropy naturally causes errors because it does not train the model to express uncertainty and say "I do not know." Kalai et al. (2025) proposes to change the evaluation practices to not reward models for guessing, but rather to mimic the human exams that penalize only wrong answers.

Detecting and Mitigating LLM Hallucinations. Orgad et al. (2025) train classifiers on the internal representations of the LLMs to predict, based on the features, the correctness of the answer. Given an LLM in a white-box setting, an input prompt, and the generated response $\hat{y}$, the classifier's task is to predict whether $\hat{y}$ is a hallucination. Orgad et al. suggested that LLMs may encode more factual knowledge in their latent subspaces than is revealed in their outputs. Gekhman et al. (2025) propposed a framework for studying hidden knowledge. Finally, Santilli et al. (2025) point out that uncertainty quantification in language models is often evaluated using metrics like AuROC. This shares biases between detection methods and correctness functions (e.g., length effects) that systematically distort results. One way to mitigate hallucinations is to act at the decoding stage, where the output generation can be steered Subramani et al. (2022). Steering vectors provide a straightforward way to control a model by adding a fixed vector to its activations (Dunefsky & Cohan, 2025). Fu et al. (2025) introduced DeepConf, a test-time method that leverages model-internal confidence signals to filter out low-quality reasoning traces during or after generation. Kuhn et al. (2023b); Fadeeva et al. (2024); Farquhar et al. (2024), and its follow-up by Kossen et al. (2025) in which they approximate the semantic entropy in a more efficient way. Constrained decoding approaches Li et al. (2023); Peng et al. (2023) modify token selection policies. Similarly, reinforcement learning with fact-based rewards Ouyang et al. (2022) has been used to bias decoding trajectories toward verifiable outcomes. Incorrect answers may also be given due to an ambiguous prompt: Kuhn et al. (2023a)'s CLAM framework uses few-shot prompts to classify a question's ambiguity and then asks the user to clarify.

3 Background and Preliminaries

3.1 Energy-Based Models

We give an overview of Energy-based Models (EBMs) and their use in discriminative classifiers.

EBMs. Energy-Based Models are a class of probabilistic models in which the probability distribution over data points $\mathbf{x}$ is defined in terms of an energy function $E_{\boldsymbol{\theta}}(\mathbf{x})$. The energy function, parameterized by a neural network $\boldsymbol{\theta}$ (Lecun et al., 2006), assigns a scalar energy to each configuration of $\mathbf{x}$, where lower energy values correspond to higher likelihood. The resulting probability distribution is given by $p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))}{Z_{\boldsymbol{\theta}}}$ where $Z_{\boldsymbol{\theta}}$ denotes the partition function (normalizing constant), defined as $Z_{\boldsymbol{\theta}} = \sum_{\mathbf{x}} \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))$ for discrete $\mathbf{x}$, or equivalently $Z_{\boldsymbol{\theta}} = \int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))\, d\mathbf{x}$ for continuous $\mathbf{x}$. Standard neural networks are often deterministic function approximators, mapping $\mathbf{x} \mapsto y$, EBMs instead define a full probability distribution over data or latent variables.

One of the strengths of EBMs is their flexibility in modeling arbitrary distributions without being tied to a specific parametric form. This flexibility comes from the fact that the energy function $E(\mathbf{x})$ can be defined in various ways. Training involves learning the parameters of the energy function such that the probability distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ matches the empirical distribution of the data. This is typically done using techniques like contrastive divergence, score matching, or maximum likelihood.

Notation. Let $\mathcal{V}$ denote the vocabulary of the LLM, i.e., the set of all tokens that can be processed as input and generated at each decoding step, with size $|\mathcal{V}| = V$. We shorten the sequence of tokens $\{\mathbf{x}_N, \ldots, \mathbf{x}_1\}$ as $\mathcal{X} = \{\mathbf{x}_{N:1}\}$, and $\mathbf{x}_i \in \mathcal{V}$ denotes the token in the i-th position along the sequence. We model the LLM as a function $\boldsymbol{\theta} : \mathbb{R}^{N \times V} \to \mathbb{R}^{V}$, implemented by a transformer, or any other sequence-to-sequence mechanism. For a sequence $\{\mathbf{x}_{i:1}\}$ as input, we write $\boldsymbol{\theta}\big(\mathbf{x}_{i:1}\big)[k]$ to denote the

predicted logit assigned to the k-th token class in $\mathcal{V}$ for the $i+1$ token in the sequence, as is standard in autoregressive LLM training (Ouyang et al., 2022).

3.2 Autoregressive Large Language Models

Generative modeling has been pursued through a variety of approaches beyond autoregression (AR). Variational Autoencoders (VAEs) (Kingma & Welling, 2014) learn a probabilistic latent variable model by encoding inputs into a latent space and decoding samples back to the data domain. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) frame generation as a min-max game between a generator and a discriminator. The diffusion process has been incorporated into neural nets (Sohl-Dickstein et al., 2015) and, more recently, Diffusion Models (Ho et al., 2020) have emerged as a powerful class of generative models. While these paradigms differ in how they approximate the data distribution, AR models are special in their kind and take a more direct route by factorizing the joint probability of sequences into conditionals, making them especially suitable for language modeling. We now focus on the AR formulation that underlies most LLMs. Textual data is segmented into a sequence of tokens $\mathcal{X} = \{\mathbf{x}_i, \dots, \mathbf{x}_1\}$, and a language modeling objective is employed to maximize the likelihood of such data (Radford & Narasimhan, 2018). In other words, we model the joint probability of tokens in the sequence $\mathcal{X}$, through a conditional probability parameterized by $\boldsymbol{\theta}$:

$$p(\mathbf{x}_{i:1}) = p(\mathbf{x}_i \mid \mathbf{x}_{i-1:1}) \dots p(\mathbf{x}_2 \mid \mathbf{x}_1)\, p(\mathbf{x}_1) = \prod_i \underbrace{p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{x}_{i-1:1})}_{\text{discriminative model}}\, p_{\boldsymbol{\theta}}(\mathbf{x}_1). \tag{1}$$

What we find interesting about this factorization is that, although it seeks to attain *generative modeling*, i.e., $p(\mathbf{x}_{i:1})$, it actually uses recursively *discriminative classifiers*, parametrized by a transformer network $\boldsymbol{\theta}$, that predicts a discrete distribution of the next token $\mathbf{x}_i$ over the vocabulary $\mathcal{V}$, given previous tokens $\mathbf{x}_{i-1:1}$. This is used to model each conditional probability.

4 How Energy Spills in LLMs

When predicting the token at position i, the conditional probability modeled by $\boldsymbol{\theta}$ can be decomposed using the probabilities of the sequences. As a result, the marginal term from step i cancels out with the sequence probability from the decomposition at the previous step $i-1$, which means we have:

$$p(\mathbf{x}_{i:1}) = \prod_i p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{x}_{i-1:1}) = \prod_i \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})}{p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})} \implies \dots \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})}{\underbrace{p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})}_{\text{step } i}} \frac{\overbrace{p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})}^{\text{step } i-1}}{p_{\boldsymbol{\theta}}(\mathbf{x}_{i-2:1})} \cdots = p(\mathbf{x}_{i:1}). \tag{2}$$

This indeed confirms that Eq. (1) results in the correct formulation for language modeling, which is $p(\mathbf{x}_{i:1})$. Following the mathematics, these quantities should cancel out along the sequence, but we will now show that, in practice, *this constraint is not explicitly optimized for, and we can exploit it for hallucination detection.*

4.1 Interpreting LLMs as Energy-based models (EBMs)

Let us continue the expansion from Eq. (2). Writing the conditional as the ratio between the joint distribution in the numerator and the marginal distribution in the denominator, we note that this ratio is actually implemented in LLMs as a softmax classifier that digests the embedding of the prior sentence $\mathbf{x}_{i-1:1}$ and predicts the next token $\mathbf{x}_i$, thus this chain of equality holds true. We can then apply the "trick" from Grathwohl et al. (2020) as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{x}_{i-1:1}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})}{p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})} = \frac{\exp \boldsymbol{\theta}(\mathbf{x}_{i-1:1})\,[\mathtt{id}(\mathbf{x}_i)]}{\sum_{k=1}^{V} \exp \boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]} \text{ where } \mathtt{id}: \{0,1\}^V \mapsto [1, \dots, V]. \tag{3}$$

$\mathtt{id}$ is the map that takes as input a one-hot encoding vector $\mathbf{x}_i$ for a word token at position i in the text and outputs its index in the vocabulary. A typical cross-entropy loss only optimizes with the

supervision provided by the ground-truth token, through the vocabulary index $\texttt{id}(\mathbf{x}_i)$. This loss ignores all other quantities or constraints related to the complete sequence $\mathcal{X}$, i.e., ignores all the time steps higher than $i+1$.

We can write the conditional probability of Eq. (3) as a ratio of two EBMs as:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{x}_{i-1:1}) = \log \frac{\exp(-E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}))}{\exp(-E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1}))} \frac{\widetilde{Z}(\boldsymbol{\theta})}{Z(\boldsymbol{\theta})} = -E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) + E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1}). \tag{4}$$

Following Zhu et al. (2021), the partition functions simplify since $\log \widetilde{Z}(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta})$[1].

$E^{\ell}_{\boldsymbol{\theta}}$, $E^{m}_{\boldsymbol{\theta}}$ are computed from the output of the model, but with two big differences: $E^{\ell}_{\boldsymbol{\theta}}$ as a single *logit* extracted using the $\texttt{id}$ of the sampled token, $E^{m}_{\boldsymbol{\theta}}$ by *marginalizing* over all $\texttt{ids}$ in the vocabulary.

The two energies can be derived from the softmax of the logits, by connecting Eq. (4) and Eq. (3):

$$-\log p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{x}_{i-1:1}) = -\log \left(\frac{\exp(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\texttt{id}(\mathbf{x}_i)])}{\sum_k \exp(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k])} \right) = \tag{5}$$

$$= \underbrace{-\boldsymbol{\theta}(\mathbf{x}_{i-1:1})\,[\texttt{id}(\mathbf{x}_i)]}_{E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})} + \underbrace{\log \sum_{k=1}^{V} \exp \boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]}_{-E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})} \tag{6}$$

where $\boldsymbol{\theta}(\mathbf{x}_{i-1:1})$ produces the logits over the entire vocabulary $\mathcal{V}$, and $\texttt{id}(\mathbf{x}_i)$ allows us to extract the logit of the sampled token at decoding step i.

We can think of $E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$ as the energy of the sampled tokens $\{\mathbf{x}_{i:1}\}$, and $E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})$ as the energy $E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$, marginalized over all possible $\mathbf{x}_i$. Considering the decoding at step i in Eq. (4), we get:

$$E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) = -\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\texttt{id}(\mathbf{x}_i)], \quad E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1}) = -\log \sum_{k=1}^{V} \exp \boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]. \tag{7}$$

Using the chain rule and Eq. (6), we can write the negative log-likelihood in terms of energies as:

$$-\log p(\mathbf{x}_{N:1}) = -\log \prod_i p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{x}_{i-1:1}) = \sum_i E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) - E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})$$

without considering the base case $p_{\boldsymbol{\theta}}(\mathbf{x}_1)$. Now, if we develop the above equation as done for Eq. (2), we write the total energy of a sequence of length N as $E_{\boldsymbol{\theta}}(\mathbf{x}_{N:1})$. Observe that the two energies, not interacting at the same step but at steps i and $i-1$, **should be equal, but they are measured in the LLM at different generation steps and from different components.**

$$E_{\boldsymbol{\theta}}(\mathbf{x}_{N:1}) = \textstyle\sum_{i=1}^{N-1} E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i+1:1}) - E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) \;=\; \ldots \overbrace{E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i+1:1})}^{\text{timestep } i+1} \underbrace{- E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})}_{\Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})} + \overbrace{E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})}^{\text{timestep } i} - E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1}) \ldots$$

At timestep $i+1$, first $-E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$ is measured, taking the denominator in the softmax as in the right part of Eq. (6), whereas at timestep i, the second $E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$ is taken, reading the logit in the softmax, left part of Eq. (6). We thus define the discrepancy between the two quantities as **spilled energy**:

Definition 4.1 (Spilled Energy $\Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$)**.** The spilled energy in an LLM is the difference between two energies that, in principle, should be equal, but given that they are measured i) at different time steps ii) in different components, could be different.

$$\Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) \triangleq -E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) + E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) = \underbrace{-\log \sum\nolimits_k \exp(\boldsymbol{\theta}(\mathbf{x}_{i:1})[k])}_{\text{timestep } i+1} + \underbrace{\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\texttt{id}(\mathbf{x}_i)]}_{\text{timestep } i} \tag{8}$$

Since both terms on the right side should be equal to $E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$, delta values should always be zero when we are correctly modeling the energy at timestep i. A shorter explanation for why spilled energy needs to be zero is given in Appendix A.3.

[1]For a formal proof, please see Appendix A.1.

4.2 Detecting hallucinations with spilled energy

EBMs have previously been used to assess neural network credibility (Liu et al., 2020), and calibration for LLMs has been explored by the Anthropic team (Kadavath et al., 2022b). However, dominant training-free baselines such as logits or "p(true)" remain weak. We likewise adopt a training-free approach, but rely on Eq. (8) and its variants as discriminants.

We feed the prompt $\{\mathbf{x}_{i-1}, \ldots, \mathbf{x}_1\}$ to the LLM $\boldsymbol{\theta}$ and obtain the completion $\{\mathbf{x}_N, \ldots, \mathbf{x}_i\}$. Following Orgad et al. (2025), we focus on the "exact answer" tokens—those in $[i+1, N]$ that contain the precise answer (e.g., `Rome` in Fig. 1), denoted $[u, w] \subseteq [i+1, N]$. For instance, it would be the tokens associated with `Rome` in the question in Fig. 1. We identify this span by prompting the LLM for a brief answer. When the answer spans multiple tokens, we apply a pooling strategy, which we ablate in Section 5. We propose measuring two values that correlate well with hallucinations:

1. the marginal energy $E^m_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$;
2. the spilled energy $\Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$ by definition of Eq. (8).

We also attempt to combine the two metrics into scaled spilled energy ΔE_s, where the spilled energy is multiplied by the absolute value of the marginal energy as $\Delta E_s(\mathbf{x}_{i:1}) = |E^m_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})| \Delta E_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$. The metrics proposed here are independent, new for LLMs, and can all be tested efficiently. These measures can be computed over the full sequence, but for error detection, as discussed in Section 5.2, we must extract the values in the localized exact interval $[u, w]$ to avoid false positives. Note that $E^\ell_{\boldsymbol{\theta}}(\mathbf{x}_{i:1})$ is the classic baseline which in literature is referred to as "logits" or "logits confidence".

5 Experiments

To evaluate spilled energy, we consider two complementary settings. First, a controlled synthetic environment, where we generate both correct and incorrect multi-digit arithmetic solutions. Second, established real-world benchmarks, where errors arise naturally across diverse reasoning and comprehension tasks. Together, these experiments test whether insights from the clean synthetic setup transfer to the complexity of open-domain language understanding.

5.1 Spilled Energy under Synthetic Arithmetic

Experimental Setting. We first evaluate spilled energy in a controlled setting: multi-digit arithmetic problems with more than 14 digits. For each instance, we generate both correct and incorrect solutions. We tested three different LLMs: Llama-3 8B (Dubey et al.), Qwen-3 8B (Qwen-Team), and Mistral-7B-Instruct v0.3 (Jiang et al.). Incorrect solutions are obtained by introducing random numerical errors of varying magnitude. Specifically, we define three error ranges that differ in their difficulty of detection:

- ◇ **Easy**: random offset in the range $[1000, 10000]$, which are typically easier to identify.
- ◇ **Medium**: random offset in the range $[100, 1000]$, where detection requires closer inspection.
- ◇ **Hard**: random offset in $[1, 10]$, much harder to detect since they appear plausible at first glance.

This design allows us to systematically probe whether spilled energy can distinguish between correct and incorrect generations across different levels of error subtlety.

Results. We observe that spilled energy values separate correct from incorrect solutions with high reliability across all error ranges and across all LLMs. In particular, spilled energy consistently assigns lower values to correct generations and higher values to incorrect ones, producing a clear margin of separation. Compared to standard baselines such as *logits*, spilled energy achieves superior discriminative power, especially for errors in the more challenging range $[1, 10]$, see Fig. 3. We offer more results in Fig. 5. Larger, better-detailed ROC and histograms are in Figs. 6 and 7 respectively.

5.2 Cross-dataset Results in Real-World Benchmarks

Experimental Setting. We evaluate our methods on a diverse set of established NLP benchmarks, including Math (Hendrycks et al.), TriviaQA (Joshi et al.), HotpotQA (Yang et al.), Winogrande

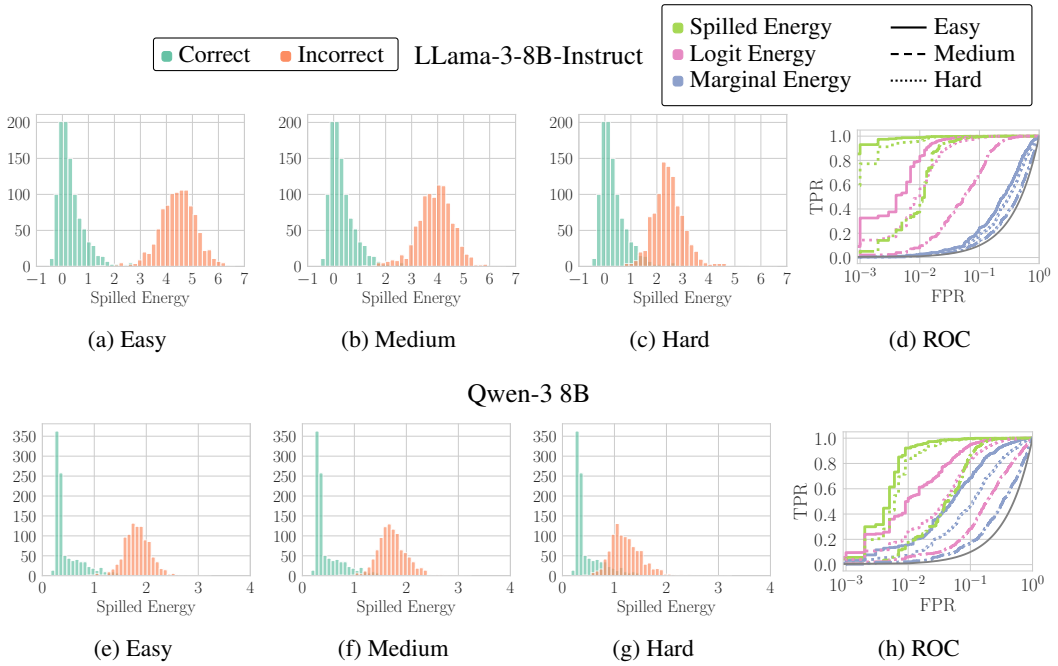
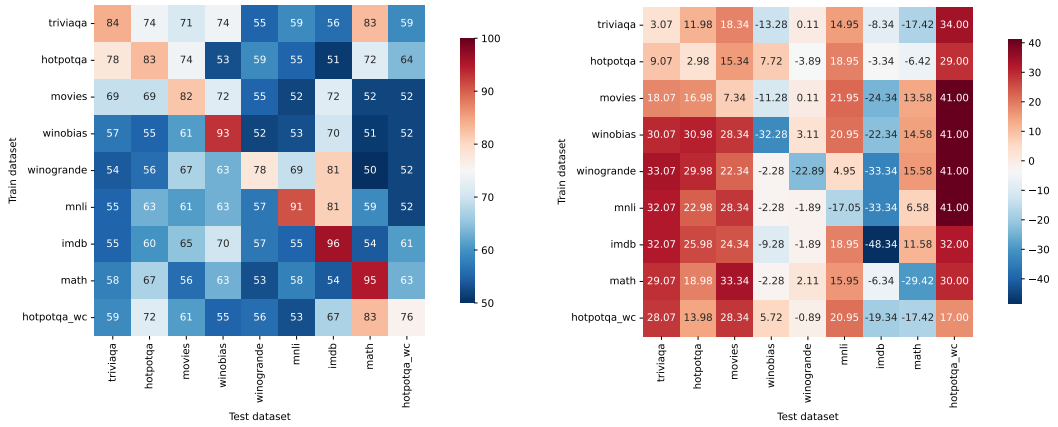


Figure 3: Histograms of Spilled Energy values across models (rows) on Math Sums with different error ranges in the answer (columns, decreasing range left to right, making it harder to detect errors). All sums are performed on 13-digit integers. In the fourth column, we show ROC curves for Hallucination Detection across the error ranges (colors) and methods (line styles).



(a) Results by Orgad et al.

(b) Spilled Energy Improvement over Orgad et al.

Figure 4: (a) AuROC performance as percentages of probing classifiers on exact answer tokens by Orgad et al. for LlaMA-3-Instruct. (b) depicts the performance difference between our Spilled ΔE with Min pooling and theirs. Positive values indicate cases where Spilled ΔE outperforms Orgad et al.. This comparison highlights the generalization capabilities of our method, compared to probing classifiers. Legend: low performance high performance.

(Sakaguchi et al.), Winobias (Zhao et al.), Movies (Tapaswi et al.), MNLI (Williams et al.) and IMDB (Maas et al.). These datasets span a wide range of reasoning and error-detection tasks, allowing us to test whether the patterns observed in the synthetic arithmetic setting extend to real-world, open-domain scenarios. Here too, we evaluate multiple LLMs that are either instruction-aligned or not aligned, such as LLaMA-3 (Dubey et al.), and Mistral (Jiang et al.). As emphasized by Orgad et al., it is essential to first localize the tokens most relevant to the final answer before applying error detection. Since exact answer tokens may consist of multiple tokens, we further adopt a pooling

Table 1: Hallucination detection performance, in terms of AuROC, across nine benchmarks and four different LLMs. We measure the generalization across all tasks by computing the average.

	Pool	HotpotQA	HotpotQA-WC	IMDB	Math	MNLI	Movies	TriviaQA	Winobias	Winogrande	Average
LLaMA-Instruct Dubey et al. (2024)											
p(true)	—	58.31±0.32	51.66±1.05	50.72±1.20	49.53±2.16	52.33±0.98	59.30±0.85	45.99±0.51	45.47±1.58	48.33±0.68	51.29±04.86
Orgad et al.	Mean	66.56±9.10	59.00±8.14	*69.78*±14.76	*66.56*±17.04	60.56±12.53	66.44±8.06	63.22±11.11	**67.33**±11.97	**58.00**±7.79	64.16±03.90
Logit E^{ℓ}	Max	72.85±2.12	91.11±1.52	42.08±5.07	57.81±3.82	25.52±3.00	43.97±1.38	68.89±1.96	39.95±2.41	49.40±2.16	54.62±18.97
Marginal E^{m}	Max	*76.72*±1.38	30.74±3.45	**85.63**±2.39	27.08±5.06	**89.90**±1.25	**96.17**±0.63	*80.13*±1.87	57.67±2.94	47.47±1.83	*65.72*±24.39
Marginal E^{m}	Min	75.91±1.62	**97.57**±0.75	14.37±2.39	**70.55**±2.43	61.21±3.24	72.21±1.60	73.38±1.86	47.19±2.71	53.98±2.30	62.93±21.89
Spilled ΔE_s	Max	53.65±1.40	36.28±2.99	55.80±4.32	35.44±3.41	58.81±2.58	70.30±1.49	48.70±2.44	36.53±2.98	44.32±1.70	48.87±11.26
Spilled ΔE	Min	**85.98**±1.09	*93.00*±1.61	47.66±4.06	65.58±3.02	*73.95*±1.97	*89.34*±1.04	**87.07**±1.33	*60.72*±2.74	*55.11*±2.05	**73.16**±15.64
LLaMA Dubey et al. (2024)											
p(true)	—	52.83±0.71	49.33±0.86	52.30±0.58	58.63±1.26	53.78±0.70	60.76±0.69	62.94±0.51	50.02±1.24	53.47±0.54	54.90±04.77
Orgad et al.	Mean	61.22±9.95	56.78±8.70	**72.67**±13.91	*69.67*±15.07	60.33±13.77	*64.00*±8.40	66.44±8.20	**60.89**±12.60	**53.56**±4.36	*62.84*±05.71
Logit E^{ℓ}	Max	53.47±2.13	49.02±1.79	48.27±1.32	57.38±6.09	91.76±0.91	57.42±1.43	52.77±2.58	50.74±1.51	51.17±1.83	56.89±12.70
Marginal E^{m}	Max	*78.00*±1.30	*76.90*±1.09	48.29±1.16	68.77±8.33	10.93±1.42	**80.70**±1.98	67.49±1.69	51.91±2.32	*51.28*±2.47	59.36±20.69
Marginal E^{m}	Min	58.39±2.79	59.20±1.95	*51.71*±1.16	34.13±8.78	*97.42*±0.51	50.37±2.43	*69.88*±1.40	49.05±2.20	49.00±2.30	57.68±16.75
Spilled ΔE_s	Min	77.75±1.52	79.44±2.05	43.39±1.82	*72.87*±6.10	**99.97**±0.08	61.56±2.95	77.55±1.62	*52.34*±2.57	48.17±1.62	68.12±17.15
Spilled ΔE	Min	**79.04**±1.78	**80.83**±1.87	43.22±1.67	**74.36**±5.54	**99.97**±0.08	61.97±2.81	**78.54**±1.57	*52.11*±2.58	48.21±1.62	**68.69**±17.48
Mistral-Instruct Jiang et al. (2023)											
p(true)	—	56.67±0.80	53.41±0.68	48.84±0.78	51.63±1.29	54.93±0.53	60.64±0.47	63.59±0.57	56.34±0.92	56.92±0.57	55.88±04.45
Orgad et al.	Mean	64.78±10.56	56.78±7.95	**82.67**±11.63	**68.78**±11.43	64.22±12.12	64.89±11.55	65.44±12.10	**61.00**±12.23	**61.44**±11.31	65.56±06.84
Logit E^{ℓ}	Max	77.24±1.66	83.84±1.66	22.28±2.54	57.67±3.29	78.98±1.58	76.89±1.49	80.35±1.88	45.53±2.60	48.17±1.97	63.44±19.99
Marginal E^{m}	Max	64.63±1.97	33.42±1.90	*81.33*±2.32	26.52±2.28	17.62±1.20	*86.60*±1.20	65.46±2.25	*56.41*±4.44	*51.14*±1.71	53.68±22.53
Marginal E^{m}	Min	*87.58*±1.35	**97.94**±0.62	18.67±2.27	*67.58*±3.37	**97.96**±0.55	84.90±1.37	*87.75*±1.73	49.19±3.97	48.49±1.86	*71.12*±25.68
Spilled ΔE_s	Max	49.13±2.50	36.37±2.40	46.45±2.56	29.05±2.57	53.79±1.55	55.24±2.17	46.73±1.98	53.30±3.66	51.20±1.84	46.81±8.24
Spilled ΔE	Min	**91.12**±1.10	*97.47*±0.78	59.77±2.57	66.63±3.46	*95.95*±0.83	**94.99**±0.93	**91.75**±1.01	50.74±3.15	49.00±1.92	**77.49**±19.42
Mistral Jiang et al. (2023)											
p(true)	—	54.21±0.76	51.68±0.76	50.40±0.50	45.86±2.05	51.94±0.50	49.12±0.63	58.00±0.67	53.76±1.17	47.29±0.55	51.36±03.73
Orgad et al.	Mean	61.78±9.27	57.44±6.95	**76.22**±12.82	65.78±15.27	56.67±11.83	64.22±8.91	64.33±10.40	**58.00**±12.29	**54.56**±4.36	62.11±06.21
Logit E^{ℓ}	Max	49.54±1.42	52.47±1.61	32.72±2.89	57.21±3.89	92.49±1.15	30.52±2.00	39.73±2.03	46.53±3.80	44.41±2.42	49.51±17.28
Marginal E^{m}	Max	83.57±1.13	*86.83*±1.70	45.31±2.49	62.26±4.29	96.03±0.83	**99.27**±0.24	**92.26**±1.31	51.31±3.35	*54.49*±2.48	*74.59*±19.91
Marginal E^{m}	Min	**87.52**±1.31	**90.91**±1.58	54.69±2.49	**86.21**±1.96	**98.80**±0.35	*94.41*±0.62	*83.66*±2.16	*52.15*±1.74	46.37±2.02	**77.19**±19.05
Spilled ΔE_s	Max	60.54±1.81	60.18±1.84	43.47±2.76	71.93±3.62	45.94±2.40	78.84±1.53	67.92±1.32	57.24±3.72	51.88±1.90	59.77±11.08
Spilled ΔE	Min	*84.24*±1.18	83.74±1.41	*57.43*±2.99	*78.26*±2.93	*96.69*±0.62	84.47±1.17	81.27±1.83	50.62±1.72	48.72±1.75	73.94±16.18

strategy across the localized span to obtain a final score per sentence. We compare spilled and marginal energy against baselines such as the probing classifiers of Orgad et al., logit confidence of Varshney et al. and p(true) of Kadavath et al..

Ablation of the exact answer token. We provide an ablation experiment on the impact of selecting the exact answer tokens. Table 2 reports average AuROC over 9 benchmarks and 3 LLMs with the exact answer, along with another column that offers the improvement provided by using the exact answer. Like prior work, we confirm that searching for the exact answer provides a notable boost: the improvement is very pronounced ($\sim 24\%$) for spilled and marginal energy, while the logit baseline receives a modest increase of 9%.

Cross-dataset results. We next evaluate in the more general setting of cross-dataset transfer, which better reflects real-world usage. For methods requiring training, we report the average performance on each dataset when trained separately on each remaining datasets (e.g., performance on IMDB is the average accuracy of classifiers trained on each of the other nine datasets). Fig. 4 shows a confusion matrix of cross-dataset performance, where the rows represent the training dataset and the columns represent the testing dataset, and where red means good performance and blue low accuracy. The model tested is LlaMA-3-Instruct. Fig. 4a shows that probing classifiers, as soon as they go out-of-distribution from the dataset on which they are trained, perform only marginally better than random guessing. The sharp drop observed in the off-diagonal elements supports our premise that this standard, in-distribution setup significantly overestimates the utility of trained probes for broad LLM deployment. Meanwhile, Fig. 4b displays the improvement of Spilled ΔE over the probing classifier, where a positive red result means improvement of our method. Ours exhibits greater performance across most datasets without requiring training. The generalization is proved with a strong increment over the off-diagonal. Moreover, in some cases, such as TriviaQA, HotpotQA, and Movies, we have improvements *even on the diagonal*. Other confusion matrices are available in Appendix D.2.

Table 1 summarizes results across nine benchmarks. The result reported in each cell is the average of the accuracies of Fig. 4a within a column. Spilled energy consistently outperforms *logit* confidence, and substantially surpasses the probing classifiers of Orgad et al. (2025). While this latter performs well when trained and tested on the same dataset, their performance drops sharply under cross-dataset evaluation, as reflected in their higher standard deviations. By contrast, ours requires no training and

Table 3: Hallucination detection performance on the Gemma Model Instruct for different parameters of the model, 1B and 4B.

	Pool	IMBD	Movies	TriviaQA	Winogrande	Winobias	MNLI	Math	HotpotQA	HotpotQA-WC	Average
Gemma-Instruct 4B Kamath et al. (2025)											
Logit E^{ℓ}	Max	50.09±0.45	60.88±3.96	53.95±2.10	49.77±0.15	**54.43**±2.80	27.00±2.16	*78.64*±3.47	62.84±1.97	64.49±2.02	55.79±13.24
Marginal E^{m}	Max	49.14±2.70	*83.02*±1.56	*84.14*±1.39	**51.49**±1.97	47.97±1.80	**100.00**±0.00	74.57±3.60	*83.70*±0.77	**85.95**±2.03	*73.33*±17.94
Marginal E^{m}	Min	50.86±2.70	51.29±3.30	55.33±1.80	48.12±1.89	51.91±2.10	99.01±0.50	76.03±3.27	62.59±1.49	71.84±2.72	63.00±15.75
Spilled ΔE_s	Max	**50.89**±1.65	50.77±5.72	56.08±2.48	*50.59*±1.72	*53.53*±2.81	95.61±0.56	43.94±3.21	50.87±1.87	51.21±1.68	55.94±14.35
Spilled ΔE	Min	**50.89**±1.65	**86.13**±4.28	**89.01**±1.06	50.18±1.97	53.10±3.05	*99.66*±0.21	**82.29**±2.46	**89.10**±1.75	*82.70*±1.35	**75.89**±17.98
Gemma-Instruct 1B Kamath et al. (2025)											
Logit E^{ℓ}	Max	*46.33*±0.82	48.12±11.45	58.89±1.61	50.50±2.45	**53.49**±3.71	49.28±2.12	**65.12**±6.62	62.24±3.62	*75.67*±1.96	56.63±9.13
Marginal E^{m}	Max	45.42±1.78	**94.15**±8.44	**83.66**±1.82	50.23±3.83	49.93±1.56	**98.17**±0.39	*64.21*±6.67	**86.87**±1.39	**82.33**±1.27	**72.77**±19.33
Marginal E^{m}	Min	**54.58**±1.78	28.93±14.50	39.80±2.54	49.84±4.38	*50.39*±1.80	56.33±1.60	63.20±4.27	41.58±2.85	61.56±1.61	49.58±10.47
Spilled ΔE_s	Max	45.17±2.37	33.27±11.49	49.01±1.67	*52.27*±3.56	49.91±2.59	77.48±1.92	40.49±4.17	49.18±3.93	35.77±2.13	48.06±12.13
Spilled ΔE	Min	45.02±2.45	*82.82*±12.91	*80.73*±2.16	**52.48**±3.75	49.77±2.82	*92.93*±1.79	56.82±6.90	*85.64*±2.23	71.86±1.77	*68.67*±16.84

generalizes robustly across diverse benchmarks. We observe that instruction-tuned models tend to amplify the margin by which spilled energy outperforms other methods, whereas on non-aligned Mistral, spilled energy may rank slightly behind marginal energy. We also compare pooling strategies and find that min pooling yields the best overall performance across methods. Table 3 shows our method generalizes to Gemma over different LLM size, 1B and 4B.

Impact of Instruction Tuning. We observe a difference in the behavior in the base models and their instruction-tuned ones. While instruction-tuning generally improves generation quality, it can degrade the calibration of classical confidence metrics, as described in Huang et al. (2023a); Ho et al. (2025). For instance, examining the average performance in Table 1, the logit baseline E_{θ}^{ℓ} decreases from 56.89% to 54.62% for LLaMA-3, indicating that fine-tuning may lead to overconfidence. In contrast, Spilled Energy (ΔE_{θ}) consistently benefits from instruction tuning, showing improved detection rates across both LLaMA-3 (68.69% to 73.16%) and Mistral (73.94% to 77.49%).

	Pool	Average % w/ exact answer	Exact answer increase
Logit E^{ℓ}	Max	56.12	+9.23
Orgad et al.	Mean	63.67	–
Marginal E^{m}	Min	67.23	+20,02
Marginal E^{m}	Max	63.34	+3,62
Spilled ΔE	Min	**73.32**	**+24.06**

Table 2: Improvement in the AuROC with the exact answer. Average across 4 LLMs and 9 benchmarks.

Variance and Generalization. A notable observation in Table 1 is the higher standard deviation associated with marginal and spilled energy compared to the probing classifiers in the average column. This variance is not a weakness but a reflection of the method's training-free nature. Since ΔE_{θ} relies on the intrinsic energy landscape of the LLM, its magnitude and sensitivity are naturally dependent on the specific domain (e.g., the sharp energy peaks in *Math* and *HotpotQA* versus the flatter distributions in *Winobias* and *IMDB*). Probing classifiers, by contrast, have high-variance when cross-testing yet the average of cross-testing results is mostly constant just above random chance ($\approx 62 - 64\%$).

Limitations. A current limitation of spilled energy is that it sometimes produces false positives on tokens that are not semantically informative, as shown in Appendix D.4. We observe this effect most prominently on punctuation tokens (e.g., commas, periods) and on words at the beginning of sentences. In these cases, the probability mass over the next token is naturally spread across many plausible options, leading to inflated spilled energy values even in otherwise correct generations. This highlights the importance of accurately identifying the *exact answer tokens*, as detection is most reliable when restricted to the parts of the output that carry the semantic content of the answer.

6 Conclusion

We reinterpreted the softmax layer of LLMs as an EBM, which lets us define *spilled energy*: the discrepancy between energy values that should be equal across consecutive time steps. We show theoretically and empirically that this discrepancy provides a strong, training-free signal for detecting hallucinations and errors in LLM outputs. Through synthetic arithmetic experiments, we demonstrate that spilled energy reliably separates correct from incorrect generations, outperforming baselines such as logits and marginal energy. Across diverse real-world NLP benchmarks, spilled energy generalizes robustly without requiring additional classifiers or task-specific training, unlike probing methods that struggle with transfer. Overall, spilled energy offers a principled and practical framework for error detection in LLMs and a new perspective on the internal energy dynamics of autoregressive models.

Ethics Statement

This work adheres to the ICLR Code of Ethics. Our study focuses on methodological contributions to error and hallucination detection in Large Language Models. We do not train new models or collect additional data; instead, we rely exclusively on publicly available datasets and widely used benchmark models for evaluation.

We note that part of our evaluation includes the Math dataset, which was publicly accessible at the time of experimentation but has since been taken down following a copyright claim. We emphasize that this dataset was used solely for evaluation purposes of our method, and only prior to the date of the takedown. No redistribution of the dataset was made, and our reported results are limited to demonstrating methodological effectiveness.

Our work does not involve personally identifiable information, sensitive content, or human subjects, and does not raise foreseeable risks of harm. We believe the proposed approach contributes positively to research on trustworthy AI by providing a training-free and generalizable framework for error detection in language models.

Reproducibility Statement

We are committed to ensuring the reproducibility of our results. All experimental details, including model configurations, evaluation protocols, and datasets used, are described in the main text and Appendix B. Upon acceptance of this work, we will publicly release the code implementing our method, along with instructions to reproduce all reported experiments. This will allow the community to verify our findings and build upon our work.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023. 2

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/. 2

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021. 2

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The LLaMa 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024. 7, 8, 9

Jacob Dunefsky and Arman Cohan. One-shot optimized steering vectors mediate safety-relevant behaviors in LLMs. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=teW4nIZ1gy. 4

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. Fact-checking the output of large language models via token-level uncertainty quantification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9367–9385, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.558. URL https://aclanthology.org/2024.findings-acl.558/. 4

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, June 2024. doi: 10.1038/s41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0. 4

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence, 2025. URL https://arxiv.org/abs/2508.15260. 4

Zorik Gekhman, Eyal Ben-David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. Inside-out: Hidden factual knowledge in LLMs. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=f7GG1MbsSM. 4

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 5

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020. 2, 3, 5

Stevan Harnad. Language writ large: Llms, chatgpt, grounding, meaning and understanding. *arXiv preprint arXiv:2402.02243*, 2024. 2

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021. 7

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 5

Zhengyi Ho, Siyuan Liang, and Dacheng Tao. Review of hallucination understanding in large language and vision models, 2025. URL https://arxiv.org/abs/2510.00034. 10

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232, 2023a. 10

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023b. 2

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 Computing Surveys*, 55(12):1–38, 2023. 2

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, 2023. URL https://arxiv.org/abs/2310.06825. 7, 8, 9

Mandar Joshi, Eunsol Choi, Daniel S 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, 2017. 7

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, 2022a. 9

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022b. 2, 7

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate. Technical report, OpenAI and Georgia Tech, September 2025. Technical Report. 4

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786. 10

Michał P. Karpowicz. On the fundamental impossibility of hallucination control in large language models, 2025. URL https://arxiv.org/abs/2506.06382. 4

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 5

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth A Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs, 2025. URL https://openreview.net/forum?id=YQvvJjLWX0. 4

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with generative language models, 2023a. URL https://arxiv.org/abs/2212.07769. 4

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=VD-AYtP0dve. 4

Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. *A tutorial on energy-based learning*. MIT Press, 2006. 4

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024. 2

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/. 4

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 2

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. A token-level reference-free hallucination detection benchmark for free-form text generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6723–6737, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.464. URL https://aclanthology.org/2022.acl-long.464. 2

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020. 3, 7

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015. 8

Alessia McGowan, Yunlai Gui, Matthew Dobbs, Sophia Shuster, Matthew Cotter, Alexandria Selloni, Marianne Goodman, Agrima Srivastava, Guillermo A Cecchi, and Cheryl M Corcoran. Chatgpt and bard exhibit spontaneous citation fabrication during psychiatry literature search. *Psychiatry Research*, 326:115334, 2023. 2

Beren Millidge. LLMs confabulate not hallucinate. *Beren's Blog*, March 2023. URL https://www.beren.io/2023-03-19-LLMs-confabulate-not-hallucinate/. 2

Mujtaba Hussain Mirza, Maria Rosaria Briglia, Senad Beadini, and Iacopo Masi. Shedding more light on robust classifiers under the lens of energy-based models. In *ECCV*, 2024. 3

Mujtaba Hussain Mirza, Maria Rosaria Briglia, Filippo Bartolucci, Senad Beadini, Giuseppe Lisanti, and Iacopo Masi. Understanding adversarial training with energy-based models, 2025. URL https://arxiv.org/abs/2505.22486. 3

OpenAI-Team. Gpt-4 technical report, 2023. URL https://cdn.openai.com/papers/gpt-4.pdf. 2

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *ICLR*, 2025. 1, 2, 4, 7, 8, 9, 10, 18, 19, 24, 25

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155. 4, 5

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. Check your facts and try again: Improving large language models with external knowledge and automated feedback, 2023. URL https://arxiv.org/abs/2302.12813. 4

Qwen-Team. Qwen3: Think deeper, act faster, 2025. URL https://qwen.ai/blog?id=1e3fa5c2d4662af2855586055ad037ed9e555125. Accessed: 2025-09-23. 7

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. In *OpenAI Technical Report*, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. 5

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2541–2573, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.155. URL https://aclanthology.org/2023.emnlp-main.155/. 2

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021. 8

Arleen Salles, Kathinka Evers, and Michele Farisco. Anthropomorphism in ai. *AJOB neuroscience*, 11(2):88–95, 2020. 2

Andrea Santilli, Adam Golinski, Michael Kirchhof, Federico Danieli, Arno Blaas, Miao Xiong, Luca Zappella, and Sinead Williamson. Revisiting uncertainty quantification evaluation in language models: Spurious interactions with response length bias results. In *ACL*, 2025. 4

Greg Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Peter Romero, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. Personality traits in large language models. *arXiv preprint arXiv:2307.00184*, 2023. 2

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 5

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.48. URL https://aclanthology.org/2022.findings-acl.48/. 4

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4631–4640, 2016. doi: 10.1109/CVPR.2016.501. 8

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation, 2023. 9

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101. 8

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models, 2025. URL https://arxiv.org/abs/2401.11817. 4

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William 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, 2018. 7

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuan-Jing Huang. Do large language models know what they don't know? In *ACL*, 2023. 2

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 15–20, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2003. URL https://aclanthology.org/N18-2003/. 8

Yao Zhu, Jiacheng Ma, Jiacheng Sun, Zewei Chen, Rongxin Jiang, Yaowu Chen, and Zhenguo Li. Towards understanding the generative capability of adversarially robust classifiers. In *ICCV*, pp. 7708–7717, 2021. 3, 6, 16

A Appendix

A.1 Partition Functions Proof used in Eq. (4)

We extend the proof of Zhu et al. to the sequence-to-sequence setting by treating next-token prediction as a multi-class classification problem. At step i, the input is the prefix $\{\mathbf{x}_{i-1:1}\}$, and the model outputs logits over the vocabulary $\mathcal{V}$ of size V. For notational consistency, we define the following energy terms:

$$\begin{cases} E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) = -\log\big(\exp\left(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\mathtt{id}(\mathbf{x}_i)]\right)\big)\,, \\ E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1}) = -\log\Big(\sum_{k=1}^{V}\exp\left(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]\right)\Big)\,. \end{cases} \tag{9}$$

The probability of the sequence up to position i can be expressed as

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}) = \frac{\exp(-E^{\ell}_{\boldsymbol{\theta}}(\mathbf{x}_{i:1}))}{Z_{\boldsymbol{\theta}}}, \tag{10}$$

where $Z_{\boldsymbol{\theta}}$ is the global partition function (normalizing constant), defined over all possible continuations of all prefixes:

$$Z_{\boldsymbol{\theta}} = \sum_{\mathbf{x}_{i-1:1}}\sum_{\mathbf{x}_i}\exp(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\mathtt{id}(\mathbf{x}_i)]) = \sum_{\mathbf{x}_{i-1:1}}\sum_{k=1}^{V}\exp(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k])\,. \tag{11}$$

Similarly, the probability of the prefix $\mathbf{x}_{i-1:1}$ can be written using the marginal energy:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1}) = \frac{\exp(-E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1}))}{\widetilde{Z}_{\boldsymbol{\theta}}}, \tag{12}$$

where $\widetilde{Z}_{\boldsymbol{\theta}}$ is the corresponding normalizing constant:

$$\widetilde{Z}_{\boldsymbol{\theta}} = \sum_{\mathbf{x}_{i-1:1}}\exp(-E^{m}_{\boldsymbol{\theta}}(\mathbf{x}_{i-1:1})) = \sum_{\mathbf{x}_{i-1:1}}\exp\left(\log\sum_{k=1}^{V}\exp\big(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]\big)\right). \tag{13}$$

By expanding the logarithm in Eq. (13), we obtain

$$\widetilde{Z}_{\boldsymbol{\theta}} = \sum_{\mathbf{x}_{i-1:1}}\sum_{k=1}^{V}\exp(\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k])\,, \tag{14}$$

which is identical to Eq. (11). Hence, the two partition functions coincide:

$$Z_{\boldsymbol{\theta}} = \widetilde{Z}_{\boldsymbol{\theta}}. \tag{15}$$

A.2 The Role of Temperature in Spilled Energy

We now analyze how the temperature parameter τ affects the definition of spilled energy. Starting from Eq. (3), the probability of the next token under temperature scaling is

$$\log p_\theta(\mathbf{x}_i \mid \mathbf{x}_{i-1:1}) = \log \frac{\exp\big(\frac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\mathrm{Id}(\mathbf{x}_i)]\big)}{\sum_k \exp\big(\frac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]\big)} \tag{16}$$

$$= \frac{1}{\tau}\,\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\mathrm{Id}(\mathbf{x}_i)] - \log \sum_k \exp\big(\tfrac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]\big)\,. \tag{17}$$

Accordingly, the spilled energy becomes

$$\Delta E_\theta(\mathbf{x}_{i:1}) = \frac{1}{\tau}\,\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\mathrm{Id}(\mathbf{x}_i)] - \log \sum_{k=1}^{|V|} \exp\big(\tfrac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_i,\ldots,\mathbf{x}_1)[k]\big)\,. \tag{18}$$

Limit case $\tau \to \infty$. When the temperature tends to infinity, the logits are scaled down towards zero, making all tokens equally likely:

$$\lim_{\tau\to+\infty} \Delta E_\theta(\mathbf{x}_{i:1}) = \lim_{\tau\to\infty} \frac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\mathrm{Id}(\mathbf{x}_i)] - \log \sum_{k=1}^{|V|} \exp\big(\tfrac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]\big) \tag{19}$$

$$= 0 - \log \sum_{k=1}^{|V|} \exp(0) \tag{20}$$

$$= -\log |V|. \tag{21}$$

Thus, for $\tau \to \infty$ the model degenerates into a uniform random classifier over the vocabulary.

Interpretation. Varying τ perturbs the balance between the two energy terms, introducing a systematic error in ΔE_θ. From the perspective of the Boltzmann distribution, scaling by $\frac{1}{\tau}$ corresponds to injecting or removing energy from the system. At high temperatures ($\tau \to \infty$), the system approaches maximum entropy, where all tokens have equal probability. At low temperatures ($\tau \to 0^+$), the distribution collapses onto the maximum logit token, making the model highly deterministic.

Error accumulation. As we generate tokens sequentially, we accumulate deviations in ΔE_θ:

$$\log p_\theta(\mathbf{x}_{i-1:1}) = \frac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[\mathrm{Id}(\mathbf{x}_i)] - \log \sum_k \exp\big(\tfrac{1}{\tau}\boldsymbol{\theta}(\mathbf{x}_{i-1:1})[k]\big) + \sum_{j=1}^{i} \Delta E_\theta(\mathbf{x}_{j:1}). \tag{22}$$

Hence, temperature scaling not only modifies the probabilities but also reshapes the cumulative error landscape traced by spilled energy.

A.3 Why Spilled Energy should be zero?

TL;DR Consider Eq. (2) in our paper and the simplification that occurs between the two probabilities between step i and step $i-1$: that simplification occurs because the probability in the denominator at step i is the same as the probability in the numerator at step $i-1$ in order to perform language modeling correctly. We measure those inside and LLMs in terms of energy, and the spilled energy is the amount by which they differ.

Please see the definition below. Let us assume a sequence of three tokens $\mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0$. If we do language modeling with autoregression, minimizing the negative log-likelihood, we have:

$$-\log p(\mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0) = -\log \underbrace{p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{x}_0)}_{\text{step 2}}\, p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_0)$$

Now, every conditional probability on the right side is implemented with a transformer ending in a softmax discriminative classifier. Equations (3) and (5) in our paper allow us to re-interpret:

$$\textbf{step 2:} \quad -\log p(\mathbf{x}_2|\mathbf{x}_1,\mathbf{x}_0) = -\log \frac{p(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0)}{p(\mathbf{x}_1,\mathbf{x}_0)} = -\log \left[\frac{\exp\left(\theta(\mathbf{x}_1,\mathbf{x}_0)[id(\mathbf{x}_2)]\right)}{\sum_k^V \exp\left(\theta(\mathbf{x}_1,\mathbf{x}_0)[k]\right)} \right] = \quad (23)$$

$$= E^l(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0) - E^m(\mathbf{x}_1,\mathbf{x}_0). \quad (24)$$

In other words, we reinterpret:

- ◇ the numerator $p(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0)$ as the energy $E^l(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0)$, which is the **logit (l)** of the softmax at timestep 2;
- ◇ The denominator as the energy $E^m(\mathbf{x}_1,\mathbf{x}_0)$ obtained with the **marginalization (m)** across the vocabulary V. This value can be read "read" simply by taking the denominator of the softmax at timestep 2. Please remember this term.

It is better to indicate them as energies (since they are not probabilities), and given their logarithmic properties, we obtain a difference. We use the notation l for logits and m for marginalization.

Now, **when we go across steps and we connect two-time steps, this is where the magic happens:**

step 1: $-\log p(\mathbf{x}_1|\mathbf{x}_0) = -\log \frac{p(\mathbf{x}_1,\mathbf{x}_0)}{p(\mathbf{x}_0)} = E^l(\mathbf{x}_1,\mathbf{x}_0) - E^m(\mathbf{x}_0)$.

We see that at timestep 1, the value $E^l(\mathbf{x}_1,\mathbf{x}_0)$ **appears again, but measured at the logit level.**

In other words, across the time-steps 2 and 1, the quantity $E(\mathbf{x}_1,\mathbf{x}_0)$ is measured twice:

- ◇ at timestep 2, as the marginalization
- ◇ at timestep 1, as the logit.

In the architecture or in the loss, there is no mechanism that forces this to be the same, but they should be equal, given the language modeling objective. This is the same as saying that in Equation (2) of our paper, the probabilities across time steps need to be simplified as we indicate.

In other words, this:

$$p(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0) = p(\mathbf{x}_2|\mathbf{x}_1,\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0)p(\mathbf{x}_0)$$

implies:

$$E(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0) = E^l(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0) \underbrace{-E^m(\mathbf{x}_1,\mathbf{x}_0) + E^l(\mathbf{x}_1,\mathbf{x}_0)}_{\text{should be zero}} \underbrace{-E^m(\mathbf{x}_0) + E^l(\mathbf{x}_0)}_{\text{should be zero}}$$

To model the energy of a sequence $E^l(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0)$ correctly, then:

- ◇ $-E^m(\mathbf{x}_1,\mathbf{x}_0) + E^l(\mathbf{x}_1,\mathbf{x}_0) = 0$ (spilled energy at timestep 2 if non-zero)
- ◇ $-E^m(\mathbf{x}_0) + E^l(\mathbf{x}_0) = 0$ (spilled energy at timestep 1 if non-zero)

so that $E(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0) = E^l(\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_0)$.

B REPRODUCIBILITY

For comparability, we adopt the same experimental setting as Orgad et al. (2025), whose implementation is publicly available at https://github.com/technion-cs-nlp/LLMsKnow. This ensures that our baselines and evaluation procedures follow an established and validated protocol.

In addition, we will release our own codebase, which includes:

- ◇ computation of the proposed energy-based measures;
- ◇ scripts for reproducing the synthetic arithmetic preliminary experiments.

The code and instructions will be made available upon acceptance of this work to facilitate full reproducibility of our results.

B.1 Exact Answer Token Detection Details

To analyze the **spilled energy** specifically on the tokens carrying the semantic weight of the answer, we must first localize the "exact answer" span $[u, w]$ within the longer generated sequence $\hat{y}$. We adopt the methodology proposed by Orgad et al. (2025), utilizing a combination of heuristics and an auxiliary instruction-tuned LLM to perform this extraction.

Extraction Strategy Depending on the nature of the task, we employ two strategies to identify the exact answer substring s:

- ◇ **Heuristic Matching:** For tasks with a closed set of possible labels (e.g., classification tasks or multiple-choice QA), we perform string matching to locate the label within the generation.
- ◇ **LLM-based Extraction:** For open-ended generation tasks (e.g., TriviaQA, Math), where the answer form varies, we employ an instruction-tuned model (Mistral-7B-Instruct) to extract the short answer from the long-form generation.

Prompting for Extraction Following Orgad et al. (2025), we prompt the auxiliary model with the original question q and the generated long answer $\hat{y}$ using the following template:

Prompt for Exact Answer Extraction

```
Extract from the following long answer the short answer, only the
relevant tokens.  If the long answer does not answer the question,
output NO ANSWER.

Q: [Question 1]
A: [LLM long answer 1]
Exact answer:  [Short exact answer 1]

Q: [Question 2]
A: [LLM long answer that does not answer the question]
Exact answer:  NO ANSWER

Q: [Question]
A: [LLM long answer]
Exact answer:
```

Verification and Token Mapping To ensure robustness, we verify that the extracted string s is a valid substring of the original generation $\hat{y}$. If the extraction is invalid or the model outputs "`NO ANSWER`," we retry the extraction up to five times. If a valid substring is still not found, the sample is excluded from the analysis to avoid identifying incorrect tokens.

Once the substring s is validated, we map it to the corresponding token indices $[u, w]$ in the original sequence. The spilled energy analysis is then performed specifically over this interval, or pooled across it (e.g., via min-pooling) as described in Section 5.2.

Answer Extraction Performance For answer localization, we achieve accuracy comparable to the results of Orgad et al. (2025). We report in Table 4 the extraction success rate across the full datasets using Mistral-7B-Instruct. Note that some datasets have been excluded (e.g., IMDB, Winobias, Winogrande) since they have a finite set of possible answers that can be used to easily locate the exact answer within the model's generation.

Table 4: Answer Extraction Success Rate across tasks for Mistral-Instruct.

Dataset	Success Rate (%)
TriviaQA	90.29
HotpotQA	87.37
Movies	93.61
MNLI	92.99
Math	87.59
HotpotQA-WC	92.38

C LLM USAGE

Large language models were used exclusively for text polishing and minor exposition refinements. All substantive research content, methodology, and scientific conclusions were developed entirely by the authors.

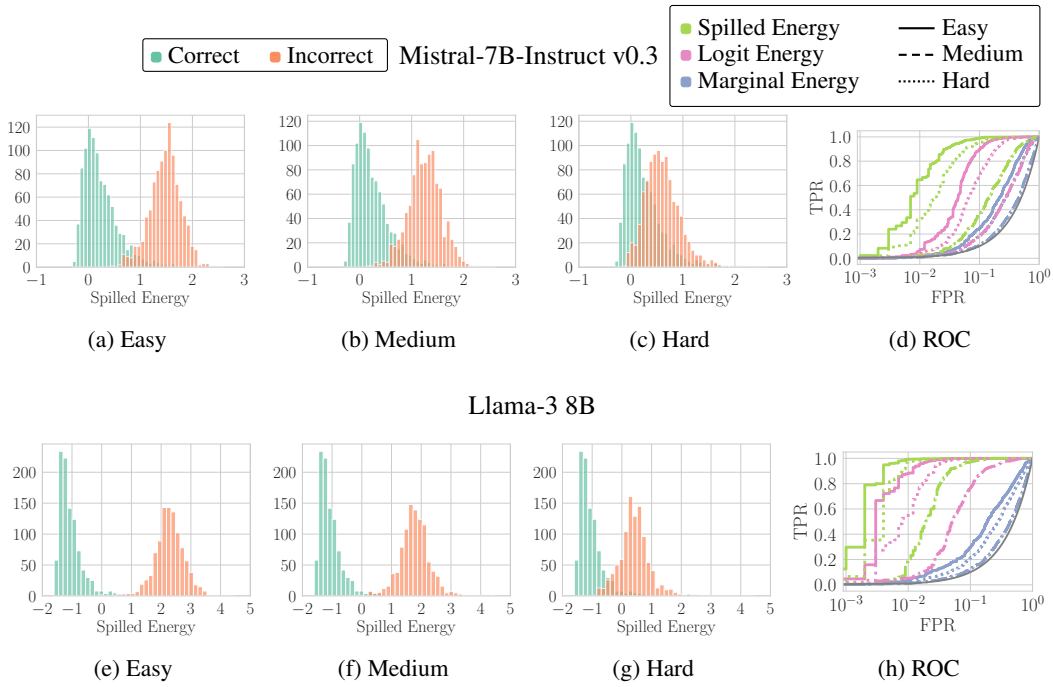
D SUPPLEMENTARY MATERIAL

This supplementary material is intended to complement the main paper by providing further motivation for our assumptions and design choices, as well as additional ablation studies or additional plots, such as ROCs and histograms, that could not fit in the main paper.

D.1 ADDITIONAL RESULTS FOR SYNTHETIC ARITHMETIC

In Fig. 5 we augmented Fig. 3 in the main paper, adding also the results for **Mistral-7B-Instruct v0.3** and **LLaMa-3-8B**. The same findings of the figure in the paper also translate to this LLM, meaning that our method generalizes across LLMs.

Fig. 6 and Fig. 7 also extend and provide more details of Fig. 3 in the main paper by showing, respectively, the histograms and the ROC at a better resolution and displayed in different frames. Also, we have added results for Mistral-7B-Instruct v0.3 and LLaMa-3-8B.



(a) Easy (b) Medium (c) Hard (d) ROC

(e) Easy (f) Medium (g) Hard (h) ROC

Figure 5: Histograms of Spilled Energy values across models (rows) on Math Sums with different error ranges in the answer (columns, decreasing range left to right, making it harder to detect errors), as described in Section 5.1. In the fourth column, we show ROC curves for Hallucination Detection across the error ranges (colors) and methods (line styles).

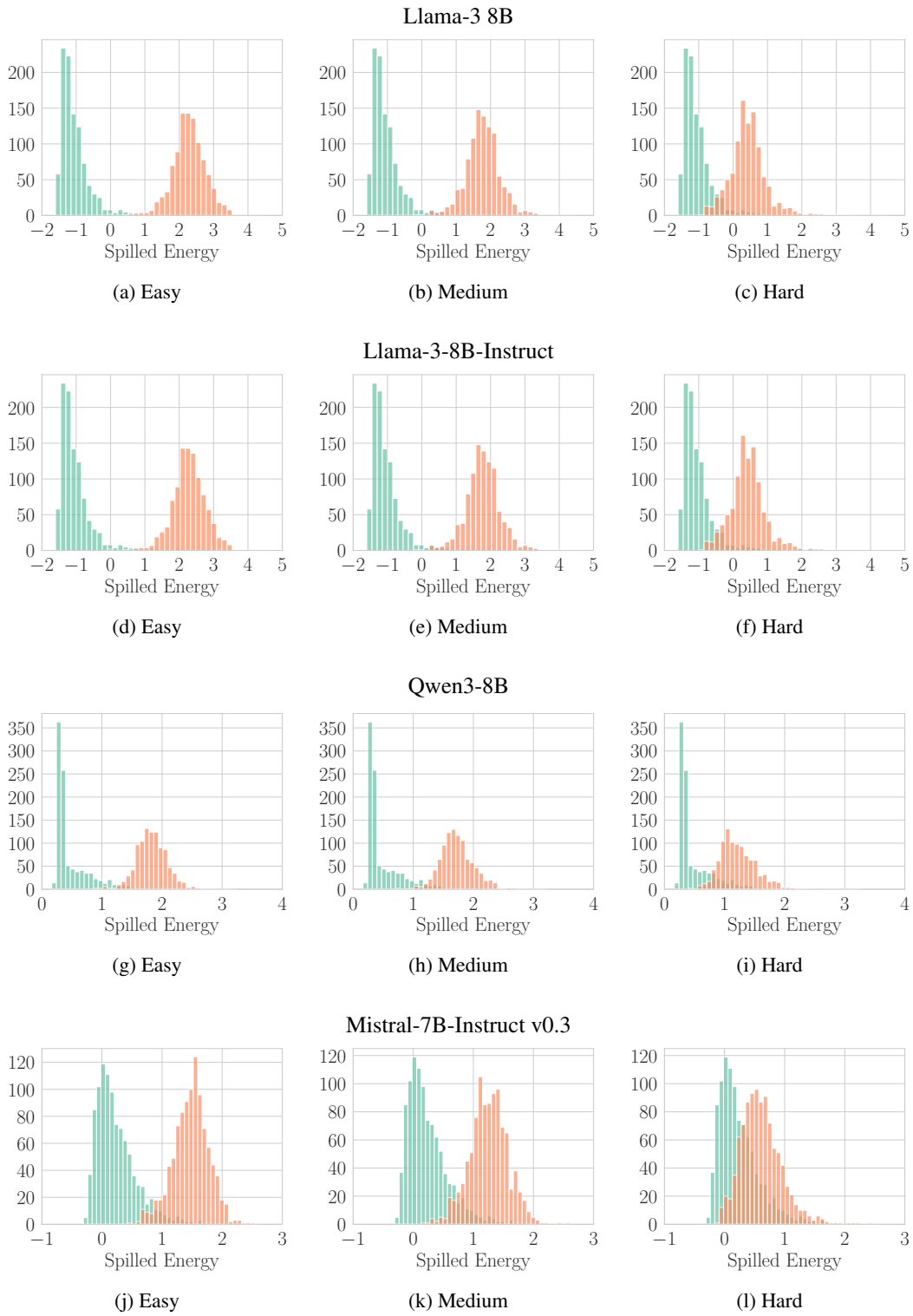
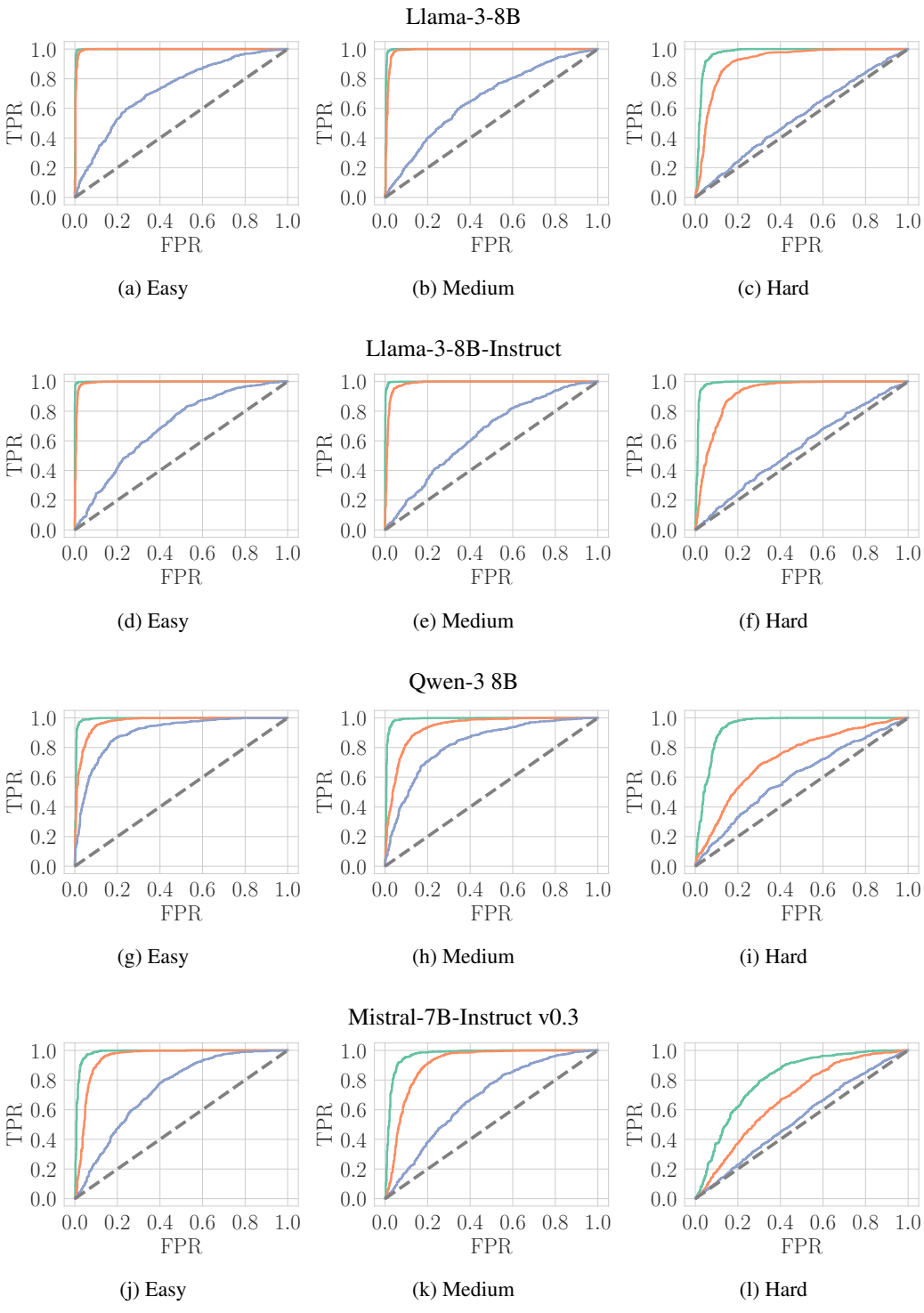


Figure 6: Histograms of Spilled Energy values for `Correct` and `Incorrect` answers across models on Math Sums, increasing difficulty from left to right. We compute sums on 13-digit integers, for incorrect answers we add a random offset sampled uniformly from the error interval: Easy $\sim \mathcal{U}(1e3, 1e4)$ - Medium $\sim \mathcal{U}(1e2, 1e3)$ - Hard $\sim \mathcal{U}(1, 10)$; for more details see Section 5.1.



(a) Easy (b) Medium (c) Hard

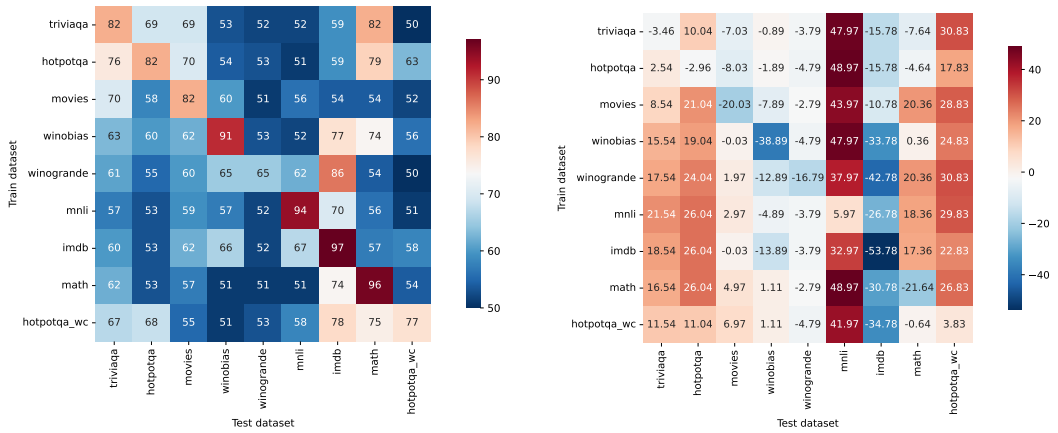
(d) Easy (e) Medium (f) Hard

(g) Easy (h) Medium (i) Hard

(j) Easy (k) Medium (l) Hard

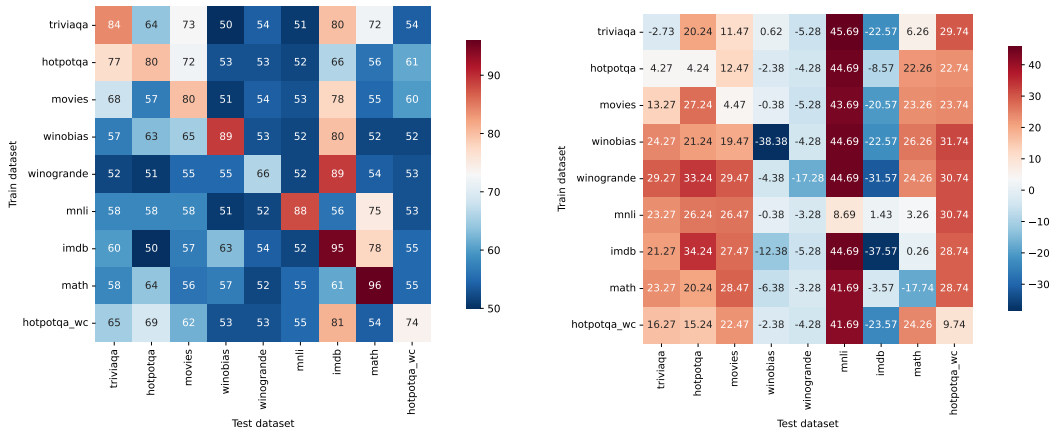
Figure 7: ROC curves for Hallucination Detection across models (rows) on Math Sums with different error ranges in the answer (columns, decreasing range left to right). All sums are performed on 13-digit integers. Legend: **Spilled (ours)** Spilled ΔE Logit E^{ℓ} Marginal E^{m}

D.2 Cross-testing Comparison with heatmaps



(a) (b)

Figure 8: Fig. 8a presents the cross-dataset performance of the method proposed by Orgad et al. (2025) using Llama-3. Fig. 8b depicts the performance difference between their method and our Spilled ΔE with Min pooling. Positive values indicate cases where Spilled ΔE outperforms the method of Orgad et al. (2025). All the numbers are expressed as percentages.



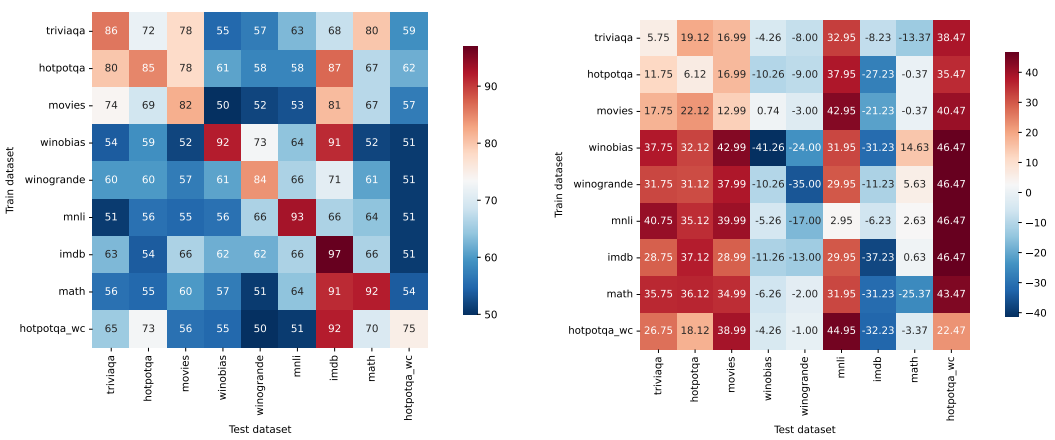
(a) (b)

Figure 9: Fig. 9a presents the cross-dataset performance of the method proposed by Orgad et al. (2025) using Mistral. Fig. 9b depicts the performance difference between their method and our Spilled ΔE with Min pooling. Positive values indicate cases where Spilled ΔE outperforms the method of Orgad et al. (2025). All the numbers are expressed as percentages.

D.3 Additional results for Cross-testing with Real World Benchmarks

Table 5 shows how our method compares with the baselines methods, Orgad et al. (2025) and Logit E^{ℓ}. This table was obtained by using various pooling methods in the pooling frame from which we measure the hallucination. More details below alongside the examples based on Fig. 11:

- ◇ **Min**: minimum energy value in the pooling frame. Energy Measured: -3
- ◇ **Max**: maximum energy value in the pooling frame. Energy Measured: 11
- ◇ **Mean**: mean among all the energies in the pooling frame. Energy Measured: 2.08
- ◇ **Last Token**: energy on the last token of the pooling frame. Energy Measured: -3



(a) (b)

Figure 10: Fig. 10a presents the cross-dataset performance of the method proposed by Orgad et al. (2025) using Mistral-Instruct. Fig. 10b depicts the performance difference between their method and our Spilled ΔE with Min pooling. Positive values indicate cases where Spilled ΔE outperforms the method of Orgad et al. (2025). All the numbers are expressed as percentages.

	Pool	HotpotQA	HotpotQA-WC	IMDB	Math	MNLI	Movies	TriviaQA	Winobias	Winogrande	Average
					LLaMA-Instruct						
Orgad et al. (2025)	Mean	66.56±9.10	59.00±8.14	*69.78*±14.76	*66.56*±17.04	60.56±12.53	66.44±8.06	63.22±11.11	**67.33**±11.97	**58.00**±7.79	64.16±3.90
Spilled ΔE	Min	**85.98**±1.09	*93.00*±1.61	47.66±4.06	65.58±3.02	*73.95*±1.97	*89.34*±1.04	**87.07**±1.33	*60.72*±2.74	*55.11*±2.05	**73.16**±15.64
Marginal E^m	Max	*76.72*±1.38	30.74±3.45	**85.63**±2.39	27.08±5.06	**89.90**±1.25	**96.17**±0.63	*80.13*±1.87	57.67±2.94	47.47±1.83	*65.72*±24.39
Marginal E^m	Min	75.91±1.62	**97.57**±0.75	14.37±2.39	**70.55**±2.43	61.21±3.24	72.21±1.60	73.38±1.86	47.19±2.71	53.98±2.30	62.93±21.89
Logit E^ℓ	Max	72.85±2.12	91.11±1.52	42.08±5.07	57.81±3.82	25.52±3.00	43.97±1.38	68.89±1.96	39.95±2.41	49.40±2.16	54.62±18.97
Spilled ΔE	Max	54.34±1.58	47.68±2.81	52.34±4.06	40.33±3.05	56.44±2.81	68.56±1.87	47.54±2.40	38.40±2.61	44.97±1.51	50.07±8.66
					LLaMA						
Orgad et al. (2025)	Mean	61.22±9.95	56.78±8.70	**72.67**±13.91	69.67±15.07	60.33±13.77	64.00±8.40	66.44±8.20	**60.89**±12.60	**53.56**±4.36	62.84±5.71
Logit E^ℓ	Min	**87.93**±1.01	**91.24**±0.80	51.73±1.32	42.99±5.68	97.01±0.43	**99.86**±0.16	**84.53**±0.87	49.29±1.46	48.52±1.78	**72.57**±22.36
Spilled ΔE	Min	*79.04*±1.78	*80.83*±1.87	43.22±1.67	**74.36**±5.54	**99.97**±0.08	61.97±2.81	*78.54*±1.57	52.11±2.58	48.21±1.62	*68.69*±17.48
Spilled ΔE_s	Min	77.75±1.52	79.44±2.05	43.39±1.82	*72.87*±6.10	**99.97**±0.08	61.56±2.95	77.55±1.62	*52.34*±2.57	48.17±1.62	68.12±17.15
Marginal E^m	Max	78.00±1.30	76.90±1.09	48.29±1.16	68.77±8.33	10.93±1.42	*80.70*±1.98	67.49±1.69	51.91±2.32	51.28±2.47	59.36±20.69
Marginal E^m	Min	58.39±2.79	59.20±1.95	51.71±1.16	34.13±8.78	97.42±0.51	50.37±2.43	69.88±1.40	49.05±2.20	49.00±2.30	57.68±16.75
Logit E^ℓ	Max	53.47±2.13	49.02±1.79	48.27±1.32	57.38±6.09	91.76±0.91	57.42±1.43	52.77±2.58	50.74±1.51	51.17±1.83	56.89±12.70
Logit E^ℓ	ALT	43.56±1.95	39.74±1.73	48.27±1.32	57.41±6.06	91.71±0.94	43.11±1.57	43.62±2.57	50.74±1.51	51.17±1.83	52.15±14.88
Logit E^ℓ	Last Token	43.56±1.95	39.74±1.73	48.27±1.32	57.41±6.06	91.71±0.94	43.11±1.57	43.62±2.57	50.74±1.51	51.17±1.83	52.15±14.88
Marginal E^m	ALT	61.59±1.88	58.64±1.60	48.29±1.16	67.93±9.32	10.75±1.44	61.39±1.80	49.73±1.45	51.19±2.59	51.44±2.50	51.22±15.61
Marginal E^m	Last Token	61.59±1.88	58.64±1.60	48.29±1.16	67.93±9.32	10.75±1.44	61.39±1.80	49.73±1.45	51.19±2.59	51.44±2.50	51.22±15.61
Marginal E^m	Mean	58.27±2.50	58.64±1.58	48.29±1.16	68.32±8.35	6.12±0.70	66.55±3.22	45.67±1.38	51.80±2.29	51.29±2.46	50.55±17.33
					Mistral-Instruct						
Orgad et al. (2025)	Mean	64.78±10.56	56.78±7.95	**82.67**±11.63	*68.78*±11.43	64.22±12.12	64.89±11.55	65.44±12.10	**61.00**±12.23	**61.44**±11.31	65.56±6.84
Spilled ΔE	Min	**91.12**±1.10	*97.47*±0.78	59.77±2.57	66.63±3.46	*95.95*±0.83	**94.99**±0.93	**91.75**±1.01	50.74±3.15	49.00±1.92	**77.49**±19.42
Marginal E^m	Min	*87.58*±1.35	**97.94**±0.62	18.67±2.27	67.58±3.37	**97.96**±0.55	84.90±1.37	*87.75*±1.73	49.19±3.97	48.49±1.86	*71.12*±25.68
Logit E^ℓ	Max	77.24±1.66	83.84±1.66	22.28±2.54	57.67±3.29	78.98±1.58	76.89±1.49	80.35±1.88	45.53±2.60	48.17±1.97	63.44±19.99
Marginal E^m	Max	64.63±1.97	33.42±1.90	*81.33*±2.32	26.52±2.28	17.62±1.20	*86.60*±1.20	65.46±2.25	*56.41*±4.44	51.14±1.71	53.68±22.53
Logit E^ℓ	Last Token	55.77±2.38	71.26±2.28	22.28±2.54	**71.21**±2.42	47.78±2.26	42.93±1.96	58.36±3.52	45.65±2.94	48.30±2.04	51.50±14.26
Logit E^ℓ	ALT	55.77±2.38	71.26±2.28	22.28±2.54	**71.21**±2.42	47.78±2.26	42.93±1.96	58.36±3.52	45.65±2.94	48.30±2.04	51.50±14.26
					Mistral						
Orgad et al. (2025)	Mean	61.78±9.27	57.44±6.95	**76.22**±12.82	65.78±15.27	56.67±11.83	64.22±8.91	64.33±10.40	58.00±12.29	**54.56**±4.36	62.11±6.21
Marginal E^m	Min	**87.52**±1.31	**90.91**±1.58	54.69±2.49	**86.21**±1.96	**98.80**±0.35	*94.41*±0.62	*83.66*±2.16	52.15±1.74	46.37±2.02	**77.19**±19.05
Marginal E^m	Max	83.57±1.13	*86.83*±1.70	45.31±2.49	62.26±4.29	96.03±0.83	**99.27**±0.24	**92.26**±1.31	51.31±3.35	*54.49*±2.48	*74.59*±19.91
Spilled ΔE	Min	*84.24*±1.18	83.74±1.41	*57.43*±2.99	*78.26*±2.93	*96.69*±0.62	84.47±1.17	81.27±1.83	50.62±1.72	48.72±1.75	73.94±16.18
Spilled ΔE	Max	61.50±1.88	63.60±1.68	42.57±2.99	76.27±3.42	47.01±2.48	81.84±1.60	68.07±1.30	**58.71**±3.69	51.13±1.87	61.19±12.30
Spilled ΔE_s	Max	60.54±1.81	60.18±1.84	43.47±2.76	71.93±3.62	45.94±2.40	78.84±1.53	67.92±1.32	57.24±3.72	51.88±1.90	59.77±11.08

Table 5: Hallucination detection performance, in terms of AuROC, across nine benchmarks and different LLMs. We measure the generalization across all tasks by computing the average.

◇ **After Last Token**: energy of the first token after the pooling method. Energy Measured: 1

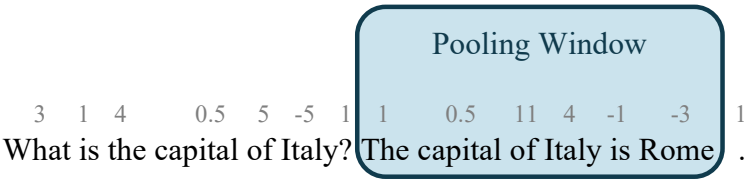


Figure 11: Example of the Pooling Window

D.4 Additional Qualitative Results

In this section, we offer additional results of the detection performance following what is shown in Fig. 1. We report both success cases and failure cases. While it is difficult to draw conclusions and predict when, why, and on which topics spilled energy may work or not, we noticed that it appears to perform reliably on knowledge-based factual content but exhibits difficulties with reasoning tasks and numerical information, despite working well on math questions as demonstrated in Section 5.1. Further investigation is required to better understand and validate these patterns.

D.4.1 Success Cases

`Question: ''Which planet is known as the Red Planet ?''`

Logits: The Red Planet is Mars . ✓
Ours: The Red Planet is Mars . ✓

Logits: The Red Planet is Jupiter . ✗
Ours: The Red Planet is Jupiter . ✗

`Question: ''What is the largest mamm al in the world ?''`

Logits: The largest mamm al in the world is the Blue Whale ✓
Ours: The largest mamm al in the world is the Blue Whale ✓

Logits: The largest mamm al in the world is the House Cat . ✗
Ours: The largest mamm al in the world is the House Cat . ✗

`Question: ''Who painted the Mona Lisa?''`

Logits: The Mona Lisa was painted by Leonardo da Vinci . ✓
Ours: The Mona Lisa was painted by Leonardo da Vinci . ✓

Logits: The Mona Lisa was painted by Pablo Esc obar . ✗
Ours: The Mona Lisa was painted by Pablo Esc obar . ✗

`Question: ''What gas do plants breathe in for photosyintesis ?''`

Logits: They breathe in carbon dioxide ✓
Ours: They breathe in carbon dioxide ✓

Logits: They breathe in oxygen ✗
Ours: They breathe in oxygen ✗

Question: ``In which continent is Egypt Located ?''

Logits: Egypt is located in Africa ✓
Ours: Egypt is located in Africa ✓

Logits: Egypt is located in Europe ✗
Ours: Egypt is located in Europe ✗

Question: ``What is the fastest land animal ?''

Logits: The fastest land animal is the che et ah ✓
Ours: The fastest land animal is the che et ah ✓

Logits: The fastest land animal is the lion ✗
Ours: The fastest land animal is the lion ✗

Question: ``What is the hardest natural substance on Earth ?''

Logits: The hardest natural substance is diamond ✓
Ours: The hardest natural substance is diamond ✓

Logits: The hardest natural substance is gold ✗
Ours: The hardest natural substance is gold ✗

Question: ``Which ocean is the largest ?''

Logits: The largest ocean is the Pacific Ocean ✓
Ours: The largest ocean is the Pacific Ocean ✓

Logits: The largest ocean is the Indian Ocean ✗
Ours: The largest ocean is the Indian Ocean ✗

D.4.2 Failure Cases

Question: ``Who was the first person to walk on the mood ?''

Logits: Neil Armstrong ✓
Ours: Neil Armstrong ✓

Logits: Buzz Ald rin ✗
Ours: Buzz Ald rin ✗

Reasoning: ''Marta has 2 sisters and 2 brothers. How many sisters has one of the brothers of Marta?''

Logits: Mart a has | 2 sisters and | 2 brothers . Each brother has the same | 2 sisters . So , one of Mart a 's brothers has | 2 sisters . ✓

Ours: Mart a has | 2 sisters and | 2 brothers . Each brother has the same | 2 sisters . So , one of Mart a 's brothers has | 2 sisters . ✓

Logits: Mart a has | 2 sisters and | 2 brothers . Each brother has the same | 3 sisters . So , one of Mart a 's brothers has | 3 sisters . ✗

Ours: Mart a has | 2 sisters and | 2 brothers . Each brother has the same | 3 sisters . So , one of Mart a 's brothers has | 3 sisters . ✗

Reasoning: ''If there are 3 cars and each car has 4 wheels , how many wheels are there in total ? ''

Logits: Each car has | 4 wheels . So , for | 3 cars , the total number of wheels is | 3 x | 4 = | 12 wheels . ✓

Ours: Each car has | 4 wheels . So , for | 3 cars , the total number of wheels is | 3 x | 4 = | 12 wheels . ✓

Logits: Each car has | 8 wheels . So , for | 3 cars , the total number of wheels is | 3 x | 8 = | 14 wheels . ✗

Ours: Each car has | 8 wheels . So , for | 3 cars , the total number of wheels is | 3 x | 8 = | 14 wheels . ✗

Reasoning: ''What is the square root of 64?''

Logits: The square root of | 64 is | 8 ✓
Ours: The square root of | 64 is | 8 ✓

Logits: The square root of | 64 is | 10 ✗
Ours: The square root of | 64 is | 10 ✗

Question: ''What blood tyoe is known as the universal donor ?''

Logits: O negative ✓
Ours: O negative ✓

Logits: AB positive ✗
Ours: AB positive ✗