Estimating Knowledge in Large Language Models Without Generating a Single Token

Anonymous ACL submission

Abstract

To evaluate knowledge in large language models (LLMs), current methods query the model and then evaluate its generated responses. In this work, we ask whether evaluation can be done *before* the model has generated any text. Concretely, is it possible to estimate how knowledgeable a model is about a certain entity, only from its internal computation? We study this question with two tasks: given a subject entity, the goal is to predict (a) the ability of the model to answer common questions about the entity, and (b) the factuality of responses generated by the model about the entity. Experiments with a variety of LLMs show that KEEN, a simple probe trained over internal subject representations, succeeds at both tasks — strongly correlating with both the QA accuracy of the model per-subject and FActScore, a recent factuality metric in open-ended generation. Moreover, KEEN naturally aligns with the model's hedging behavior and faithfully reflects changes in the model's knowledge after fine-tuning. Lastly, we show a more interpretable yet equally performant variant of KEEN, which highlights a small set of tokens that correlates with the model's lack of knowledge. Being simple and lightweight, KEEN can be leveraged to identify gaps and clusters of entity knowledge in LLMs, and guide decisions such as augmenting queries with retrieval.

1 Introduction

The standard approach for evaluating knowledge in large language models (LLMs) relies on querying the model, letting it generate responses, and then evaluating the responses. This evaluation can be done using various methods, including comparing responses to gold answers (Touvron et al., 2023; Cohen et al., 2023a), measuring response consistency over multiple generations (Cohen et al., 2023b; Manakul et al., 2023; Kuhn et al., 2023), checking the support of responses in external evidence (Gao et al., 2023; Bohnet et al., 2022), or estimating the model's uncertainty per-response (Yu et al., 2024; Yuksekgonul et al., 2024; Li et al., 2023; Snyder et al., 2023; Liu et al., 2022).

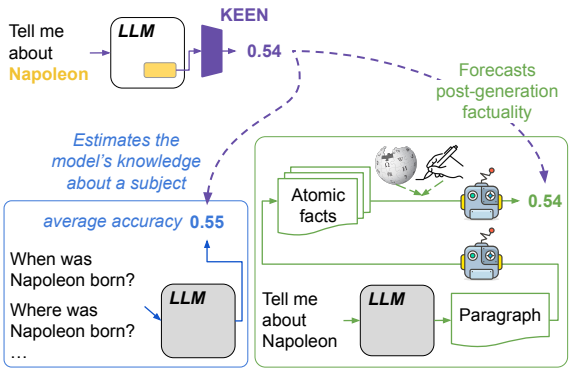

Figure 1: We show that simple probes (KEEN), trained over hidden model representations, quantify the model's knowledge about a given subject entity — estimating the model's question-answering accuracy on entity-related questions (bottom left) and forecasting the factuality of model-generated texts about the entity (right).

In this work, we take a step back and ask whether it is possible to evaluate the model's knowledge *before* it generates any text, using only its internal computation. This view is analogous to human studies that show the effectiveness of non-verbal communication for assessing witness credibility in the courtroom (Remland, 1994; Denault et al., 2024). Concretely, we propose to evaluate how knowledgeable an LLM is about a given subject entity (e.g. Napoleon or Empire State Building), by considering only how it processes the name of that entity, and *before* it generates a single token.

We formalize this problem as entity knowledge estimation (§2) and devise two concrete tasks. Given an entity, the goal is to predict: (a) how many common questions about the subject entity the model will answer correctly (Figure 1, bottom left), and (b) how many of the claims in a model generated response about the subject are factually correct (Figure 1, right).

To tackle entity knowledge estimation, we capitalize on findings from recent interpretability works which show that, during inference, the hidden representations of an input entity capture many attributes related to it (Geva et al., 2023; Meng et al., 2024), and often these attributes can be extracted with linear functions (Hernandez et al., 2024). Therefore, we propose (§3) to estimate how knowledgeable a model is about a given entity by training simple probes, called KEEN (Knowledge Estimation of ENtities), over the model's representations of the entity (Figure 1, upper left).

We evaluate KEEN in two experimental settings (§4) of factual question answering (QA) and open-ended generation (OEG) of biographies. In the QA setting, we derive a set of questions per-subject for subjects in PopQA (Mallen et al., 2023) and evaluate how well KEEN predicts the model's average accuracy per-subject across these questions. In the OEG setting, we evaluate the correlation of KEEN with FActScore (Min et al., 2023), a post-generation hallucination detector. In both settings and across models of different sizes and families — GPT2 (Radford et al., 2019), Pythia (Biderman et al., 2023), LLaMA2 (Touvron et al., 2023), and Vicuna (Chiang et al., 2023) — KEEN consistently shows a strong correlation between 0.58-0.68 with model accuracy and 0.66-0.77 with factuality. Moreover, KEEN probes trained on entity representations show substantially stronger correlation with model accuracy and factuality than probes trained on commonly-used intrinsic features, such as fully-connected scores and self-attention activations, and external features, such as entity-popularity.

Further analyzing the utility and features of KEEN (§5), we show that KEEN faithfully correlates with the model's hedging behavior, i.e., the score predicted by KEEN decreases as the fraction of per-entity questions that a model hedges on increases. In addition, KEEN faithfully reflects changes in the model's knowledge following fine-tuning: training LLaMA2 on Wikipedia articles about certain entities increases their KEEN score while scores for other entities tend to decrease. Lastly, we show that training KEEN on the vocabulary projections of entity representations (nostalgebraist, 2020; Geva et al., 2021) increases the probe's interpretability without performance cost, identifying a small set of tokens that signal a lack of entity knowledge.

To conclude, we present KEEN, a simple and lightweight approach for quantifying how knowledgeable a model is about a given entity from intrinsic properties, which well-estimates the accuracy and factuality of model outputs about the entity. We also show that KEEN scores are reflective of both hedging behavior and changes in entity-based knowledge over fine-tuning. KEEN could be used to inform developer decisions such as whether to augment queries with retrieval, discard certain queries (e.g. by abstaining), enhance models with external tools, or identify "holes" in the model's knowledge to apply further training on. We release our code and data at `https://anonymized`.

2 Entity Knowledge Estimation

Our goal is to evaluate how much knowledge an LLM captures about an entity from how it processes the entity's name alone, without obtaining model responses and evaluating them post-generation. This view is motivated by growing evidence from interpretability works which find that, during model inference, knowledge is centralized in the hidden representations corresponding to named entities (Meng et al., 2024; Geva et al., 2023; Li et al., 2021).

Given a subject entity s (e.g. Napoleon or Empire State Building) and a model M, our goal is to estimate two related quantities: (a) the performance of M on queries about s, and (b) the probability that M will generate incorrect facts given any query about s. These two quantities are expected to be related, as they are both influenced by and reflect the amount of knowledge M captures about s.

To evaluate entity knowledge, we propose two concrete evaluation settings:

Question Answering (QA) For a subject entity s and a set of common question-answer pairs $\mathcal{Q} = \{\langle q_i, a_i \rangle\}_{i=1}^{n}$ about s, denote by $\hat{a}_i$ the answer predicted by a model M for the query q_i. Given only the subject s, our goal is to estimate the average accuracy of M over $\mathcal{Q}$, denoted as $y_{QA}^{(s)} := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[\hat{a}_i = a_i]$.

Open-Ended Generation (OEG) For a general information-seeking query q about a subject s (e.g. *"Tell me facts about Napoleon"* or *"Generate a paragraph about Napoleon"*), let $\mathcal{R} = \{\langle c_i, a_i \rangle\}_{i=1}^{m}$ be the set of claims in the response generated by M, each with a 0/1 label indicating its correctness with respect to external evidence. Claims can be extracted and evaluated for correctness using various automatic methods (e.g., Nenkova and Passonneau, 2004; Shapira et al., 2019; Zhang and Bansal,

2021). Given only the subject s, the task is to predict the portion of factually correct claims in $\mathcal{R}$, denoted as $y^{(s)}_{OEG} := \frac{1}{m}\sum_{i=1}^{m} a_i$.

A naive solution for both tasks would be to first obtain queries about s, feed them to M, and evaluate the answers M generates. Here we seek an efficient solution, which estimates the knowledge of M about s, without iteratively executing M.

3 KEEN

Geva et al. (2023) showed that for a given subject in the input, LLMs construct an information-rich representation of the subject that encodes many of its attributes. Furthermore, subject attributes can be extracted from the subject representation with a simple linear function (Hernandez et al., 2024). We capitalize on these findings and propose to train a simple probe over the model's representations of subjects to predict how much knowledge the model captures about them. In our following formulation (and the rest of the paper), we focus on widely-adopted transformer-based auto-regressive language models.

Notation Assuming a language model with L layers, a hidden dimension d, a vocabulary $\mathcal{V}$, and an unembedding matrix $W_U \in \mathbb{R}^{|\mathcal{V}|\times d}$. Let $\mathbf{h}_{\ell,i}$ be the hidden representation at position i and layer ℓ, omitting normalization, $\mathbf{h}_{\ell,i}$ is computed as:

$$\mathbf{h}_{\ell,i} = \mathbf{h}_{\ell-1,i} + \mathbf{a}_{\ell,i} + \mathbf{m}_{\ell,i}$$

where $\mathbf{a}_{\ell,i}$ and $\mathbf{m}_{\ell,i}$ denote the outputs from the ℓ-th multi-head self-attention and MLP sublayers, respectively (Vaswani et al., 2017).

3.1 Features

Let $t^{(s)}_1, ..., t^{(s)}_{s_r}$ be the sequence of s_r input tokens corresponding to a given subject s (e.g. `N, ap, oleon` for the subject Napoleon tokenized with GPT2). We use the representations at the last subject position (s_r), denoted as $\mathbf{h}^{(s)}_{1,s_r}, ..., \mathbf{h}^{(s)}_{L,s_r}$, to construct a feature vector $\mathbf{z}^{(s)} \in \mathbb{R}^{d_z}$.[1]

We use the following sets of features for $\mathbf{z}^{(s)}$:

- **Hidden states (HS)**: We take the subject representation from multiple upper-intermediate layers, where attributes of the subject are often extracted during inference (Geva et al., 2023; Meng et al., 2024) and are easier to disentangle (Huang et al., 2024; Hernandez et al., 2024). To account for variations in the inference pass of different subjects, we choose 3 consecutive layers $\mathcal{L} = \left\{\frac{3}{4}L + k \mid k \in \{-1, 0, 1\}\right\}$, from which we extract the hidden states $\{\mathbf{h}^{(s)}_{\ell,s_r} \mid \ell \in \mathcal{L}\}$. Then, we normalize these vectors (see details below) and average them into a d-dimensional feature vector.
- **HS with vocabulary projection (VP)**: We take the same hidden states as in HS, but instead of using them as-is, we use their projections to the vocabulary (nostalgebraist, 2020; Geva et al., 2021). Namely, we normalize and average the vectors $\{W_U f_L(\mathbf{h}^{(s)}_{\ell,s_r}) \mid \ell \in \mathcal{L}\}$ into a $|\mathcal{V}|$-dimensional feature vector, where f_L is the layer norm applied at the last layer of the model. While VP is not expected to improve performance, it could enhance interpretability, as the learned weight for each token signifies feature importance in quantifying subject-related knowledge.
- **HS with top-k of vocabulary projection (VP-k)**: Since the vocabulary space is typically large, in order to make the probe more interpretable and efficient, we perform feature selection over the trained VP probe to extract the k most influential tokens from the vocabulary projections. We then normalize and average the obtained $3 * k$ features (k for each layer) to train a new smaller probe over k-dimensional feature vectors.

For each of HS, VP, and VP-k, we apply Min-Max normalization before averaging the extracted vectors, which scales each feature to be within $[0, 1]$. For example, after extracting the hidden states $\{\mathbf{h}^{(s)}_{\ell,s_r} \mid \ell \in \mathcal{L}\}$ for some subject s, we normalize the values of every entry $i \in [d]$ and layer $\ell \in \mathcal{L}$ over a set of subjects $\mathcal{S}$. Let $\hat{\mathbf{h}}^{(s)}_{\ell,s_r} \in \mathbb{R}^d$ be the normalized $\mathbf{h}^{(s)}_{\ell,s_r}$, so the feature vector for HS is defined as $\mathbf{z}^{(s)} = \frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}} \hat{\mathbf{h}}^{(s)}_{\ell,s_r} \in \mathbb{R}^d$.

3.2 Probing

We define the following probe for predicting the model's QA accuracy $y^{(s)}_{QA}$ or response factuality $y^{(s)}_{OEG}$ given the features $\mathbf{z}^{(s)}$ for a subject s:

$$f(\mathbf{z}) := \sigma(\boldsymbol{\theta} \cdot \mathbf{z}) \quad (1)$$

Where σ is the sigmoid function and $\boldsymbol{\theta} \in \mathbb{R}^{d_z}$ is a single linear transformation. The sigmoid non-linearity is necessary to aid the model in learning

[1] In practice, we obtain the hidden representations using the query: "`This document describes [s]`". This is to avoid placing the subject in the first position of the input, which often encodes biases that could affect performance on our task (Xiao et al., 2024; Geva et al., 2023).

scores in the range $[0, 1]$.[2]

For each of the two tasks $T \in \{\text{QA, OEG}\}$, we optimize $\boldsymbol{\theta}$ over features and labels collected for a set of subjects $\mathcal{S}$ by minimizing the MSE loss:

$$\mathcal{L}_{MSE}(\boldsymbol{\theta}) = \|y_T^{(s)} - \sigma(\boldsymbol{\theta} \cdot \mathbf{z}^{(s)})\|_2^2$$

For more details on the probes' training, see §A.

4 Experiments

In this section, we evaluate KEEN and baselines that rely on different intrinsic and external features. We observed that the VP-50 probe obtained comparable performance while being significantly more interpretable (discussed in B.2) so we focus on evaluating the VP and VP-50 variants of this probe.

4.1 Experimental Setting

Data For the QA task, we sample 3,472 subject entities from PopQA (Mallen et al., 2023) and generate a set of 5.3 questions on average per subject. To generate questions, we take subject-relation-object triplets from Wikidata (Vrandečić and Krötzsch, 2014) and convert them into question-answer pairs with handwritten templates. For instance, the triplet (Napoleon, place of birth, France) will be converted to the question *"Where was Napoleon born?"* and the answer *"France"*. In addition, we augment each such example with multiple variants that cover different answer granularities (Yona et al., 2024), accounting for both answer and subject aliases, and handling cases with multiple answers. We consider a model's prediction for a given subject-relation pair as correct if it contains an exact match with any answer alias in at least one question variation.

For the OEG setting, we use the FActScore dataset (Min et al., 2023), which includes model-generated biographies, extracted claims, and claim labels which indicate whether the claim is supported or not-supported by the subject's Wikipedia page. We compare our results to the FActScore scores of the same generating model.

Examples for the two tasks are shown in Table 1. For both settings, we randomly split each dataset into disjoint sets of subjects: 65% train, 15% development, and 20% test. Importantly, the FActScore dataset and QA train set have a negligible number of overlapping subjects, 1 (0.2%), which allows us to test transfer learning between the two settings.

[2]We also experimented with linear probes and found that they tended to converge to scores in a narrow range around 0.5, failing to capture the signals in the inputs.

Baselines We evaluate three baselines that utilize intrinsic features and external features. For intrinsic features, we take the two best variants reported by Snyder et al. (2023), which trained binary hallucination detectors for QA. These detectors use the outputs from the self-attention and MLP modules as features, which were also considered by other recent methods for similar tasks (Yu et al., 2024; Yuksekgonul et al., 2024; Li et al., 2023).

- **Entity popularity (Pop.)**: It has been established that LLM performance is influenced by entity popularity (Mallen et al., 2023; Kandpal et al., 2023; Yona et al., 2024). We follow previous works (e.g., Chen et al., 2021; Mallen et al., 2023; Cohen et al., 2024) and approximate entity popularity using statistics from Wikipedia. Concretely, we use the total number of monthly views of the entity's page between the years 2000-2023.
- **Self-attention outputs (ATTN)**: We train the same probe of KEEN (Eq. 1), while using $\mathbf{a}_{L,s_r}^{(s)}$ as the feature vector $\mathbf{z}^{(s)}$, i.e., the output of the last self-attention sublayer for the last input token (which is the last subject token in our setup).
- **Fully-connected activations (FC)**: Here we train a similar probe to ATTN, which sets $\mathbf{z}^{(s)}$ to $\mathbf{m}_{L,s_r}^{(s)}$, the output of the last MLP sublayer for the last input token.

Models We analyze 7 auto-regressive language models across various sizes, latent spaces, and training objectives: GPT2-XL (Radford et al., 2019), Pythia 6B and 12B (Biderman et al., 2023), LLaMA2 7B and 13B (Touvron et al., 2023), and Vicuna 13B (Chiang et al., 2023).[3] The vocabulary sizes range between 30K-50K tokens and the hidden state dimensions range from 4096-5120.

Evaluation For every model and subject s in our data, we feed the model a generic prompt "`This document describes [`s`]`" and extract the features used for all methods: KEEN and the above baselines. Using these features, we obtain predictions for our two tasks for every method. For the Pop. baseline, we simply take the corresponding popularity value of the subject. We report Pearson correlation and the MSE between the predicted and gold scores, for every task, model and method. Correlation results are provided in §4.2 and the MSE results are reported in §B.

[3]We also analyzed Vicuna 7B, but due to its poor accuracy in the QA setting and inconsistent behavior, we omitted it in the main results. Results for Vicuna 7B can be found in §B.

Input subject s	Task	Output $y^{(s)}_{QA}$ / $y^{(s)}_{OEG}$	Example ⟨question, model answer⟩ / ⟨claim, correctness label⟩ pairs from $\mathcal{Q}$ / $\mathcal{R}$
George Washington	QA	0.67	⟨In what city was George Washington born?, Westmoreland County⟩, ⟨What is the religion of George Washington?, Episcopal Church⟩ ⟨Who is the father of George Washington?, Augustine Washington⟩
	OEG	0.74	⟨George Washington was a military man., 1⟩, ⟨George Washington was the first President of the United States., 1⟩, ⟨He was educated at the College of William and Mary., 0⟩

Table 1: Example input subject and the expected outputs for the two tasks for Pythia 12B. The output labels were computed based on the average QA accuracy over 12 questions (0.67), and the FActScore score for 35 claims (0.74).

	GPT2 XL	Pythia 6B	Pythia 12B	LLaMA2 7B	LLaMA2 13B	Vicuna 13B
Pop.	0.30	0.32	0.28	0.27	0.25	0.26
FC	0.49	0.59	0.55	0.50	0.49	0.49
ATTN	0.53	0.63	0.60	0.58	0.50	0.52
VP-50	0.54	0.64	0.59	0.53	0.48	0.50
VP	**0.61**	**0.68**	**0.64**	**0.64**	**0.58**	**0.60**
HS	**0.60**	**0.68**	**0.64**	**0.64**	**0.58**	**0.60**

Table 2: Correlation with the average QA accuracy for the KEEN QA probes and baselines.

Model	Pop.	FC	ATTN	VP-50	VP	HS
Pythia 12B	0.36	0.61	**0.77**	0.72	0.75	**0.77**
Vicuna 13B	0.37	0.49	0.65	0.55	**0.66**	**0.66**

Table 3: Correlation with FActScore for the KEEN OEG probes and baselines.

Model	Pop.	FC	ATTN	VP-50	VP	HS
Pythia 12B	0.47	0.41	0.55	0.40	0.57	**0.60**
Vicuna 13B	0.40	0.52	0.50	0.48	0.61	**0.62**

Table 4: Transfer learning results, showing the correlation between FActScore and KEEN QA probes and baselines. Results are reported over all 500 subjects in the FActScore dataset.

4.2 Results

KEEN well-estimates the model's knowledge about the subject entity Table 2 and 3 show the QA and OEG results, respectively.

In both settings and across all models, KEEN probes trained on hidden representations and vocabulary projections demonstrate the strongest correlation of 0.60-0.68 with QA accuracy and 0.66-0.77 with FActScore. This shows that it is possible to predict how knowledgeable a model is about an entity from the entity's hidden representations.

Predicting factuality based on common intrinsic features (FC and ATTN) consistently underperforms with respect to KEEN, further supporting the finding that entity knowledge is centralized in entity representations during inference. Furthermore, the entity popularity baseline (Pop.) performs poorly on both tasks, with low correlation values of ≤ 0.32 in QA and ≤ 0.36 in OEG. This shows that while external statistics of popularity (such as Wikipedia page count) are useful in deriving general performance trends, they often fail to provide fine-grained entity-level predictions.

Surprisingly, for the Pythia models even the KEEN OEG VP-50 probe strongly correlates with FActScore, indicating that there is a relatively small set of tokens which are influential in increasing/decreasing predicted accuracy. We further analyze these tokens in §5.4 and provide intuition for interpreting them. Moreover, we discuss the trade-off between interpretability and score correlation in B.2.

KEEN QA probes generalize to predict factuality in OEG Since knowledge is centralized in the internal representations of entities, their use in estimating knowledge should transfer across different settings. Table 4 shows that the predictions of KEEN QA probes have a strong correlation of 0.60-0.62 with FActScore. Further, the correlation of KEEN QA probes with QA accuracy and FActScore are notably similar, e.g. 0.60 and 0.62 for Vicuna 13B KEEN QA HS probes, respectively. These results show that HS and VP features capture signals that generalize across settings, regardless of whether the task requires explicit (QA) or implicit (OEG) recall of factual knowledge by the model.

5 Analysis

In this section, we further look into the predictions and features of KEEN, evaluating its faithfulness with respect to model hedging (§5.1) and changes in the model's knowledge following training (§5.2). In addition, we analyze its errors (§5.3) and the features of its VP-50 variant (§5.4).

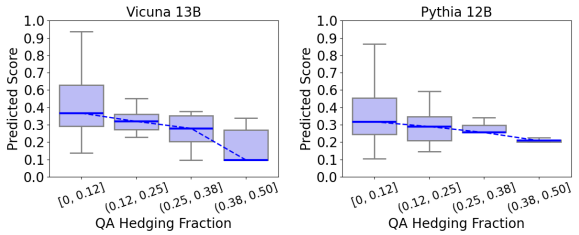

Figure 2: KEEN QA scores as a function of the fraction of per-subject queries that Vicuna 13B and Pythia 12B hedge on.

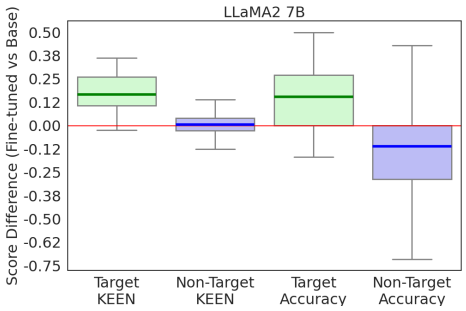

Figure 3: Changes in the KEEN QA score and average QA accuracy after fine-tuning LLaMA2 7B on paragraphs about a target subject, for target and non-target subjects. These results are aggregated over individually fine-tuning for 20 subjects.

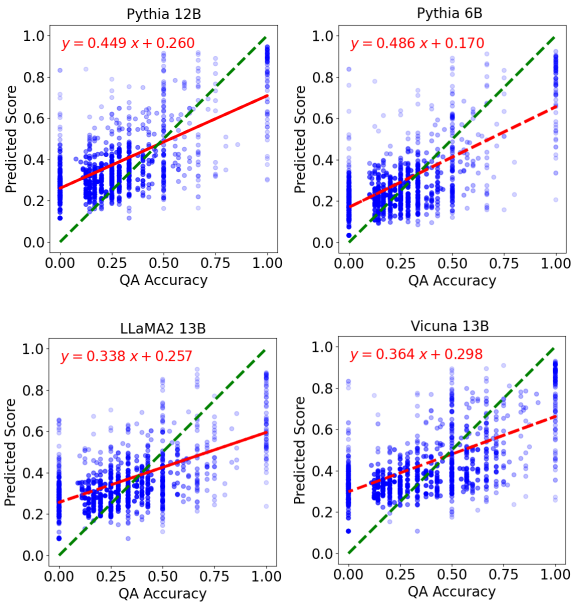

Figure 4: Predicted scores from the KEEN QA VP probe and the golden QA Accuracy scores are positively linearly related.

5.1 Correlation with Model Hedging

To prevent factually incorrect responses, LLMs are trained to hedge in cases of uncertainty, for example by generating "`I don't know`" (Ganguli et al., 2023). Therefore, it is expected that models generally hedge on entities they are less knowledgeable about. Since the KEEN QA probe score estimates entity-based knowledge, we hypothesize that it should correlate with the fraction of questions that a model hedges on about the entity.

Figure 2 confirms this hypothesis, showing that the KEEN QA VP score decreases as the fraction of queries the model hedges on increases. This implies that models may hedge based on features of the model's internal representations of the entity, similarly to KEEN.

5.2 Reflecting Changes in Model Knowledge

Our experiments so far evaluated KEEN while keeping the underlying LLM fixed. A natural question that arises is whether changes in the model's knowledge are reflected in changes in the KEEN score. We test this by fine-tuning LLaMA2-7B on paragraphs about a target subject and measuring changes in both the average QA accuracy and the KEEN QA score. Concretely, we sample 20 subjects from the QA test dataset and retrieve paragraphs from the Wikipedia page of each subject using BM25 (Robertson et al., 1995).[4] Then, we use LoRA (Hu et al., 2022) to fine-tune $< 0.5\%$ of the model's parameters, separately for each subject. After fine-tuning for a certain target subject, we compute the KEEN score for that subject, as well as for 256 non-target subjects from the QA test dataset. The KEEN QA probe trained over the model's hidden states before fine-tuning is used to compute these scores.

Figure 3 shows that on average, QA accuracy scores for the target entities increase by 0.16 and KEEN QA scores increase by 0.18, as models are fine-tuned on paragraphs related to them. Conversely, inline with works about catastrophic forgetting which find that models tend to forget information about entities observed in pre-training (Tirumala et al., 2022), the QA accuracy scores for non-target entities decrease after fine-tuning. However, the KEEN scores for non-target entities stay relatively constant. Since fine-tuning often doesn't erase residual information in LLMs (Patil et al., 2024), and KEEN relies on intermediate representations, a possible explanation for this discrepancy is that information is still encoded in the representations but the model fails to recall it.

[4] We use the Wikipedia dump from August 28, 2023.

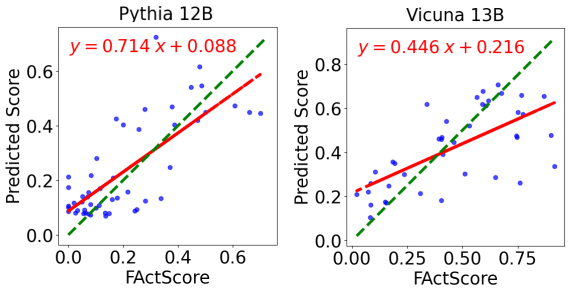

Figure 5: Predicted scores of the KEEN OEG VP probe versus FActScore scores. KEEN scores are positively linearly correlated with FActScore scores.

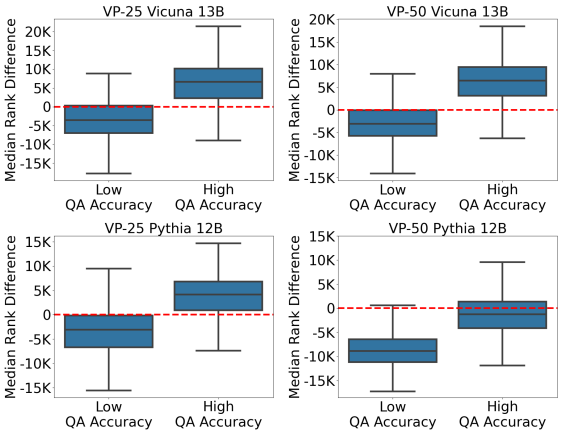

Figure 6: Difference, per subject, in the median rank of tokens with negative weight and tokens with positive weight. Pythia 12B VP-25 and VP-50 show the trade-off between interpretability and performance – though the median ranks of negative weight tokens are lower on average than positive weight tokens in VP-50, there is still a clear split in both accuracy groups.

5.3 Error Analysis

To better understand the limitations of KEEN, we plot the probes' predicted scores against the reference QA accuracy and FActScore scores.

Figure 4 shows that the KEEN VP QA probes tend to predict higher scores relative to QA accuracy for subjects that the model knows less about, although the KEEN scores for entities with QA accuracy between $[0, 0.5]$ do generally fall within a similar range of $[0.1, 0.5]$. For subjects that the model is more knowledgeable about, KEEN QA scores are more conservative, as seen by the cluster of scores below the $y = x$ line for QA accuracy values close to 1.0. Generally, KEEN scores have less variance than the QA accuracy scores since the slopes of the trend-lines are < 1, which may suggest that more complex predictors are needed to capture all the variance of QA accuracy. These trends are consistent across models of different families and sizes.

In §B, we include results for the other models, which follow the same trends in Figure 4 and Figure 5. We also provide the scatter plots for probes trained on the different KEEN features and baselines, all demonstrating the same linear relations.

5.4 Feature Analysis for VP-25 and VP-50

We analyze the most influential features of the KEEN QA VP probes to understand which tokens contribute most to predicting average QA accuracy. Our goal in this analysis is to identify tokens that either increase or decrease predicted QA accuracy, and to determine whether they are promoted in the representations of subjects with high and low QA accuracy, respectively. As a concrete example, for subjects with low QA accuracy, we expect tokens that decrease QA accuracy to generally be ranked lower in the subject representations than tokens that increase QA accuracy. Since the input normalization scheme described in §3.1 normalizes a token's logit in a given hidden state by its magnitude across the other subjects (not with respect to the other tokens in the hidden state), we can interpret the weight learned by the KEEN QA VP probe for each token as its direction and magnitude of influence on the predicted score.

	Weight	Example influential tokens
Pythia 12B	Pos	`analysis, Statistical, Players, Senator, Quantum, nationality, investments`
	Neg	`circadian, AMPK, lys, 16, jo, VERT, diese, see, Mort, ))*, dep, imi, ac`
Vicuna 13B	Pos	`athlet, kick, swing, developer, compiling, official, sales, GitHub, Movie`
	Neg	`sle, hurt, Circ, Alt, book, JK, ja, adow, istema, ppings, adjust, istol`

Table 5: Examples of the most influential tokens in the KEEN QA VP probes that were assigned positive and negative weights. These are some of the tokens that correspond to the features of KEEN QA VP-50 probes.

First, we identify the tokens associated with the largest absolute weights in the KEEN QA VP probes, as they are most influential on the predicted score. Next, we compare the median rank of tokens with negative weights to those with positive weights in the vocabulary projections of subjects with high QA accuracy (1.0) and low QA accuracy (0.0). Figure 6 shows that for low QA accuracy subjects, the median rank of negative weight tokens is generally lower than that of positive weight tokens. Conversely, for high QA accuracy subjects, the median

rank of negative weight tokens is generally higher than that of positive weight tokens. This opposing trend in the two accuracy groups indicates that there is a small set of tokens which hold signals for differentiating between subjects the model knows a lot about and those it knows less about.

We provide these important tokens in Table 5. Tokens assigned positive weight are related to meaningful concepts while tokens assigned negative weight are often numbers, abbreviations, or suffixes. A possible interpretation of this semantic difference between positive weight tokens and negative weight tokens is that the hidden states of low accuracy subjects encode less content, reflecting the model's lack of knowledge about them.

6 Related Work

Evaluation of knowledge and factuality of LLMs The common practice for estimating knowledge in LLMs is to query the model and then evaluate its outputs. This is often conducted though question-answering setups with gold labels (Roberts et al., 2020; Petroni et al., 2019; Cohen et al., 2023a, inter alia), by letting the model generate multiple responses and measuring response consistency (Cohen et al., 2023b; Manakul et al., 2023; Kuhn et al., 2023), checking whether the generated output is supported by external evidence (Gao et al., 2023; Bohnet et al., 2022; Min et al., 2023), or by estimating the model's uncertainty per-response (Zhang et al., 2023; Jesson et al., 2024). Unlike these methods, we focus on evaluating the model's entity knowledge beyond a single response, based on intrinsic features extracted before generating a single token.

Probing internal representations of LLMs Probing over internal representations has been used to predict model behavior, such as truthfulness (Marks and Tegmark, 2023; Azaria and Mitchell, 2023a), and properties of language, such as part-of-speech (Belinkov et al., 2017; Nikolaev and Padó, 2023), syntax (Hewitt and Manning, 2019), and sentence length (Adi et al., 2017) *for a specific input*. Probing has also been used to identify which hidden states are most influential on the performance of tasks, like classification (Alain and Bengio, 2017). Our use of probing differs from prior work because we estimate a property that captures model behavior over many inputs rather than a single input. Namely, the KEEN score provides a knowledge estimate relevant to any input concerning the entity. Further, KEEN focuses on estimating entity-specific knowledge and is useful in evaluating several model behaviors, including hedging, shifts in knowledge, and truthfulness.

Hallucination detection using intrinsic features Our work is closely related to methods that leverage intrinsic features for detecting factually incorrect claims, but has two core differences. The first being in our choice of features: we specifically use the hidden states corresponding to the named entity from the upper intermediate layers. In contrast, existing methods use various other features, like intermediate activation values (Azaria and Mitchell, 2023b), outputs from the self-attention modules (Yu et al., 2024; Yuksekgonul et al., 2024; Li et al., 2023; Snyder et al., 2023), soft-max prediction probabilities, and fully-connected scores (Snyder et al., 2023). Yu et al. (2024); Goloviznina and Kotelnikov (2024); Su et al. (2024) also examine the intermediate hidden representations, but for the purpose of identifying whether there exists a subspace of hidden states that lead to hallucinations. Similarly to our work, Yu et al. (2024) uses the hidden representations of subjects, but rather to train a binary hallucination detector. Unlike all these works that use internal representations to predict the factuality of a specific claim, we learn to estimate knowledge from a single internal representation of an entity, which is applicable to any claim pertaining to it.

7 Conclusion

We present the problem of estimating entity knowledge solely from the model's internal representations of the entity. We show that KEEN offers a simple and interpretable solution which correlates with model performance in both QA and OEG settings, as well as with current hallucination detection methods. Further, KEEN is also reflective of both hedging behavior and changes in knowledge throughout fine-tuning. From a broad perspective, our results demonstrate the potential of estimating model qualities and behavior for certain inputs based on intrinsic features, and call for future work to leverage simple and efficient methods like KEEN to improve the factuality and reliability of LLMs.

Limitations

While our approach successfully estimates the extent of the model's knowledge about a subject, it

does not identify specific gaps or clusters of knowledge. For instance, KEEN can estimate that the model will be 55% truthful when generating content about *Napoleon*, but it does not pinpoint that the model is unable to answer the specific question, *What military academy did Napoleon attend?*. An interesting direction for future work would be to develop a more fine-grained approach that predicts how knowledgeable the model is about specific aspects of the subject (e.g. military career of Napoleon) or identifies specific facts encoded in subject representations.

Another limitation is that this work focuses on estimating knowledge for entities, however not all subjects of questions are entities. For example, there is no clear subject for which we can apply KEEN in the question, *How does exercise influence mental health?*. KEEN also assumes that the subjects are already extracted for analysis. While identifying named entities in text is a well-studied task in NLP (Nadeau and Sekine, 2007), combining it with KEEN could make this approach more complex and computationally expensive.

Our evaluation focuses only on transformer-based auto-regressive LLMs. While this is one of the most popular and largest families of LLMs, it would be valuable to study the applicability of KEEN to other model architectures. Notably, Sharma et al. (2024) shows that factual recall in Mamba is similarly centered in the hidden states of the last subject token from the intermediate layers, so we expect our approach to generalize to other recurrent architectures.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes.

Amos Azaria and Tom Mitchell. 2023a. The internal state of an LLM knows when it's lying. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Amos Azaria and Tom Mitchell. 2023b. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Anthony Chen, Pallavi Gudipati, Shayne Longpre, Xiao Ling, and Sameer Singh. 2021. Evaluating entity disambiguation and the role of popularity in retrieval-based NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4472–4485, Online. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. 2023a. Crawling the internal knowledge-base of language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1856–1869, Dubrovnik, Croatia. Association for Computational Linguistics.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023b. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640, Singapore. Association for Computational Linguistics.

Vincent Denault, Chloé Leclerc, and Victoria Talwar. 2024. The use of nonverbal communication when assessing witness credibility: a view from the bench. *Psychiatry, Psychology and Law*, 31(1):97–120.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. 2023. The capacity for moral self-correction in large language models. *arXiv preprint arXiv:2302.07459*.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Valeriya Goloviznina and Evgeny Kotelnikov. 2024. I've got the "answer"! interpretation of llms hidden states in question answering. *Preprint*, arXiv:2406.02060.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. 2024. Ravel: Evaluating interpretability methods on disentangling language model representations. *arXiv preprint arXiv:2402.17700*.

Andrew Jesson, Nicolas Beltran-Velez, Quentin Chu, Sweta Karlekar, Jannik Kossen, Yarin Gal, John P. Cunningham, and David Blei. 2024. Estimating the hallucination rate of generative ai. *Preprint*, arXiv:2406.07457.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

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

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *Preprint*, arXiv:2310.06824.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2024. Locating and editing factual associations in gpt. In *loc*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Dmitry Nikolaev and Sebastian Padó. 2023. Investigating semantic subspaces of transformer sentence embeddings through linear structural probing. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 142–154, Singapore. Association for Computational Linguistics.

nostalgebraist. 2020. interpreting gpt: the logit lens.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2024. Can sensitive information be deleted from LLMs? objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *Conference on Empirical Methods in Natural Language Processing*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Language Models are Unsupervised Multitask Learners*.

Martin S Remland. 1994. The importance of nonverbal communication in the courtroom. *Atlantic Journal of Communication*, 2(2):124–145.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. Crowdsourcing lightweight pyramids for manual summary evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba. *Preprint*, arXiv:2404.03646.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. 2023. On early detection of hallucinations in factual question answering. *arXiv preprint arXiv:2312.14183*.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran HU, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. *Preprint*, arXiv:2403.06448.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Preprint*, arXiv:2205.10770.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu,

Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Narrowing the knowledge evaluation gap: Open-domain question answering with multi-granularity answers. *arXiv preprint arXiv:2401.04695*.

Lei Yu, Meng Cao, Jackie Chi Kit Cheung, and Yue Dong. 2024. Mechanisms of non-factual hallucinations in language models. *Preprint*, arXiv:2403.18167.

Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. 2024. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. In *The Twelfth International Conference on Learning Representations*.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6617–6632, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023. Enhancing uncertainty-based hallucination detection with stronger focus. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 915–932, Singapore. Association for Computational Linguistics.

A KEEN Training Details

Hyper-parameter tuning for KEEN probes All KEEN QA and OEG probes were trained with the AdamW optimizer with weight decay 0.01, and batch size of 32.

KEEN QA hyper-parameters were optimized over the configuration combinations presented in Table 6.

Hyper-parameter	Values
Learning Rate	$10^{-3}, 5e^{-3}, 5e^{-4}, 10^{-4}, 5e^{-5}$
Epochs	100, 1K, 3K, 5K

Table 6: KEEN QA probe hyper-parameter configurations

KEEN OEG hyper-parameters were optimized over the configuration combinations presented in Table 7.

Hyper-parameter	Values
Learning Rate	$5e^{-4}, 10^{-4}, 5e^{-5}, 10^{-5}$
Epochs	1K, 3K, 5K

Table 7: KEEN OEG probe hyperparameter configurations

The best hyper-parameters for QA and OEG KEEN probes are found in Table 8 and Table 9, respectively.

KEEN Probe	Hyper Param	GPT2 XL	Pythia 6B	Pythia 12B	LLaMA 7B	LLaMA 13B	Vicuna 7B	Vicuna 13B
HS	Epoch	3K	100	100	1K	1K	1K	100
	LR	10^{-5}	10^{-4}	10^{-4}	10^{-5}	10^{-5}	10^{-5}	10^{-4}
VP	Epoch	3K	100	3K	1K	1K	1K	3K
	LR	10^{-5}	10^{-4}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
VP-200	Epoch	3K	500	3K	3K	3K	3K	3K
	LR	10^{-5}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
VP-100	Epoch	3K	500	3K	3K	3K	3K	3K
	LR	10^{-5}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
VP-50	Epoch	3K	1K	3K	3K	3K	3K	3K
	LR	10^{-5}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
VP-25	Epoch	5K	3K	3K	3K	3K	3K	3K
	LR	10^{-5}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
VP-10	Epoch	-	-	-	3K	-	-	-
	LR	10^{-5}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
FC	Epoch	-	-	-	1K	-	-	-
	LR	-	-	-	10^{-5}	-	-	-
ATTN	Epoch	-	-	-	1K	-	-	-
	LR	-	-	-	10^{-5}	-	-	-

Table 8: Best hyper-parameters for KEEN QA probes and baselines.

Hyper-parameters for fine-tuning LLaMA2 7B The training details for the fine-tuning experiment in §5.2 are described in Table 10.

KEEN Probe	Hyper-parameter	Pythia 12B	Vicuna 13B
HS	Epoch	1K	1K
	LR	10^{-5}	1^{-4}
VP	Epoch	5K	3K
	LR	10^{-5}	$1e^{-5}$
VP-200	Epoch	5K	3K
	LR	10^{-4}	$1e^{-4}$
VP-100	Epoch	5K	5K
	LR	10^{-4}	$5e^{-4}$
VP-50	Epoch	5K	5K
	LR	10^{-4}	$5e^{-4}$
VP-25	Epoch	5K	5K
	LR	$5e^{-4}$	1^{-3}
VP-10	Epoch	5K	5K
	LR	10^{-3}	$5e^{-4}$
FC	Epoch	1K	1K
	LR	$5e^{-5}$	$5e^{-5}$
ATTN	Epoch	1K	1K
	LR	10^{-5}	$5e^{-5}$

Table 9: Best hyper-parameters for KEEN OEG probes and baselines.

Optimizer	LR	Epoch	Scheduler	Warm Up ratio	LoRA alpha	LoRA dropout	LoRA r
AdamW	$2e^{-4}$	100	Linear	0.03	16	0.1	64

Table 10: Hyper-parameters for fine-tuning LLaMA2 7B.

Resources All our experiments were conducted using the PyTorch package (Paszke et al., 2019) on a single A100 or H100 GPU.

B Additional Results

B.1 Mean Standard Error (MSE) for KEEN

Table 11, Table 12, Table 13, present the MSE for the KEEN OEG probes with FActScore scores, KEEN OEG probes with FActScore scores, the KEEN QA probes with FActScore scores, and the KEEN QA probes with average QA accuracy scores. The performance of these probes is discussed in §4.2.

Model	Freq.	FC	ATTN	VP-50	VP	HS
Pythia-12B	0.028	0.026	0.014	0.020	0.017	0.014
Vicuna-13B	0.052	0.075	0.049	0.052	0.040	0.039

Table 11: MSE for KEEN OEG Probes between predicted KEEN scores and FActScore scores.

B.2 Interpretability-Performance Tradeoff for KEEN VP-k

Figure 7, Figure 8, and Figure 9 demonstrate diminishing returns in increasing the parameter count beyond 50 tokens, suggesting that a small set of tokens contains significant signals for estimating entity knowledge. There is a clear trade-off between

Model	Freq.	FC	ATTN	VP-50	VP	HS
Pythia 12B	-	0.053	0.043	0.047	0.062	0.074
Vicuna 13B	-	0.046	0.053	0.052	0.049	0.050

Table 12: MSE for KEEN QA probes and FActScore.

	Freq.	FC	ATTN	HS	VP-50	VP
GPT2 XL	0.053	0.053	0.046	0.041	0.045	0.040
Pythia 6B	0.063	0.052	0.048	0.045	0.047	0.042
Pythia 12B	0.069	0.059	0.051	0.056	0.052	0.053
LLaMA2 7B	0.069	0.068	0.057	0.051	0.062	0.053
LLaMA2 13B	0.073	0.061	0.060	0.053	0.061	0.053
Vicuna 7B	0.015	0.010	0.010	0.011	0.015	0.010
Vicuna 13B	0.086	0.074	0.071	0.064	0.072	0.062

Table 13: MSE for KEEN QA Probes between predicted KEEN scores and QA accuracy scores.

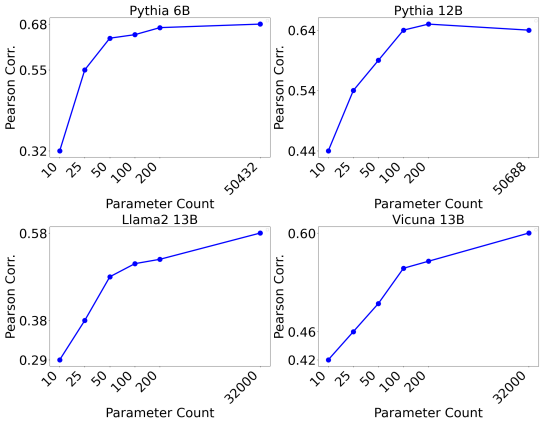

Figure 7: Correlation of KEEN QA VP probe scores with QA accuracy as a function of input parameter count.

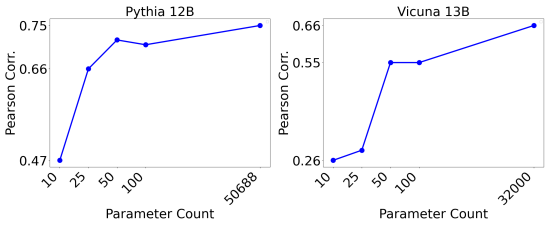

Figure 8: Correlation of KEEN OEG VP probe scores and FActScore as a function of input parameter count.

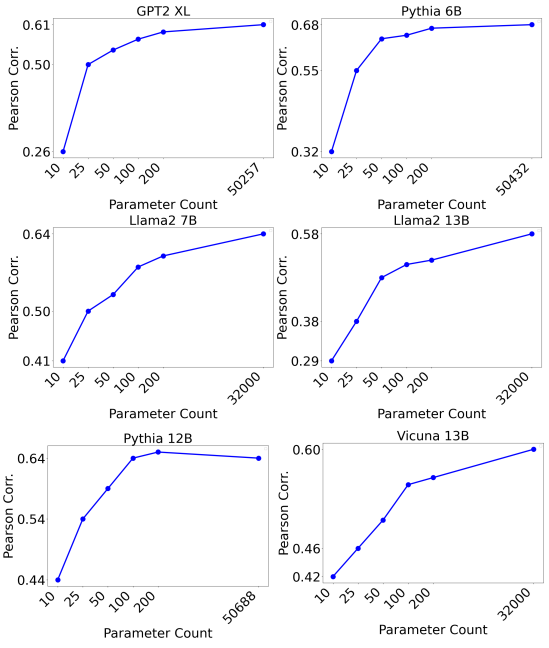

Figure 9: KEEN QA probe correlation with QA accuracy as a function of token count.

the greater interpretability of smaller KEEN VP probes and the reduced correlation. However, the KEEN VP-50 variant remains highly interpretable with a minimal number of tokens and does not suffer a substantial correlation decline. Consequently, we chose to focus on evaluating the KEEN VP and VP-50 variants.

B.3 QA correlation plots

Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, and Figure 16 show the results for the QA experiments in §4.2 for GPT2 XL, Pythia 6B, Pythia 12B, LLaMA 7B, LLaMA 13B, Vicuna 7B, and Vicuna 13B, respectively.

B.4 OEG correlation plots

Figure 17 and Figure 18 show the results for the OEG experiments in §4.2 for Vicuna 13B and Pythia 12B, respectively.

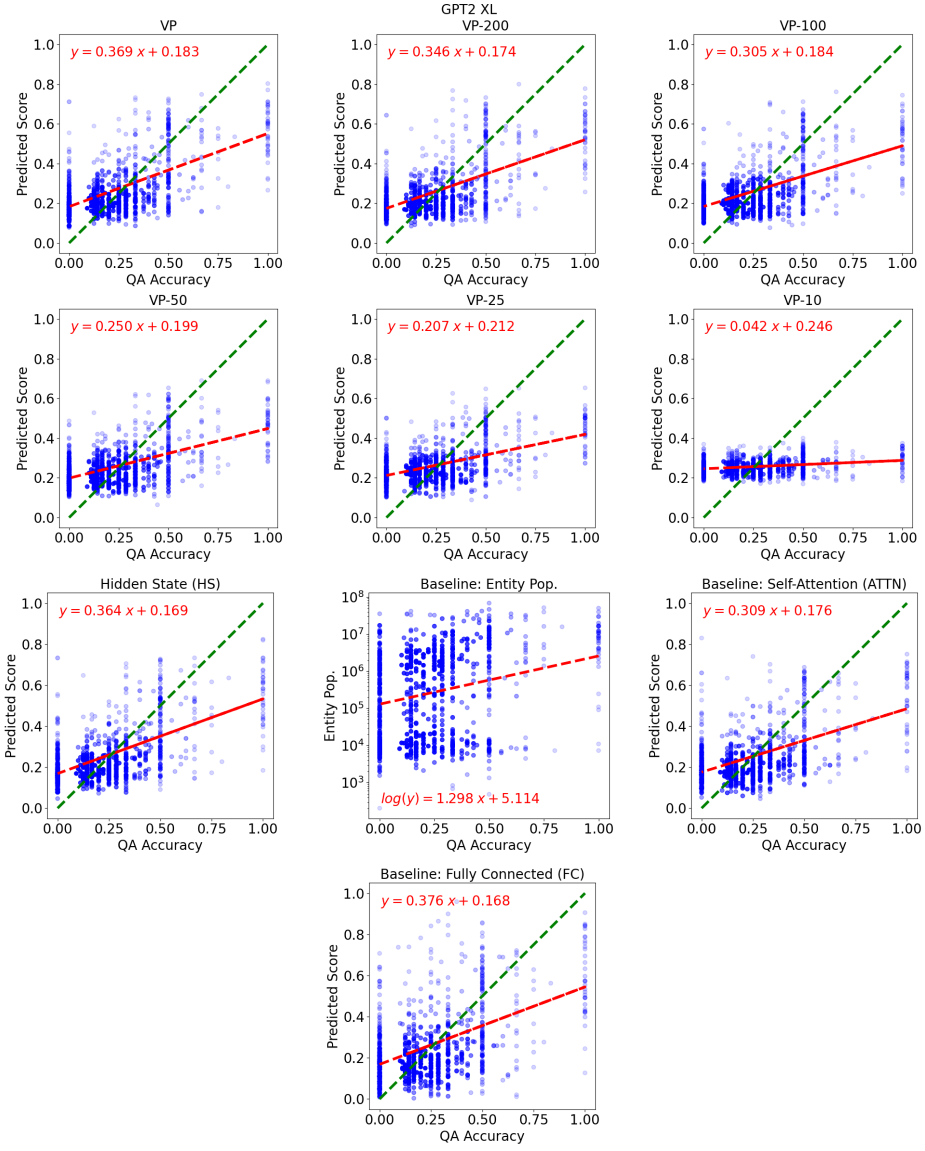

Figure 10: GPT2 XL: Predicted scores from the KEEN QA probe versus the golden QA accuracy scores.

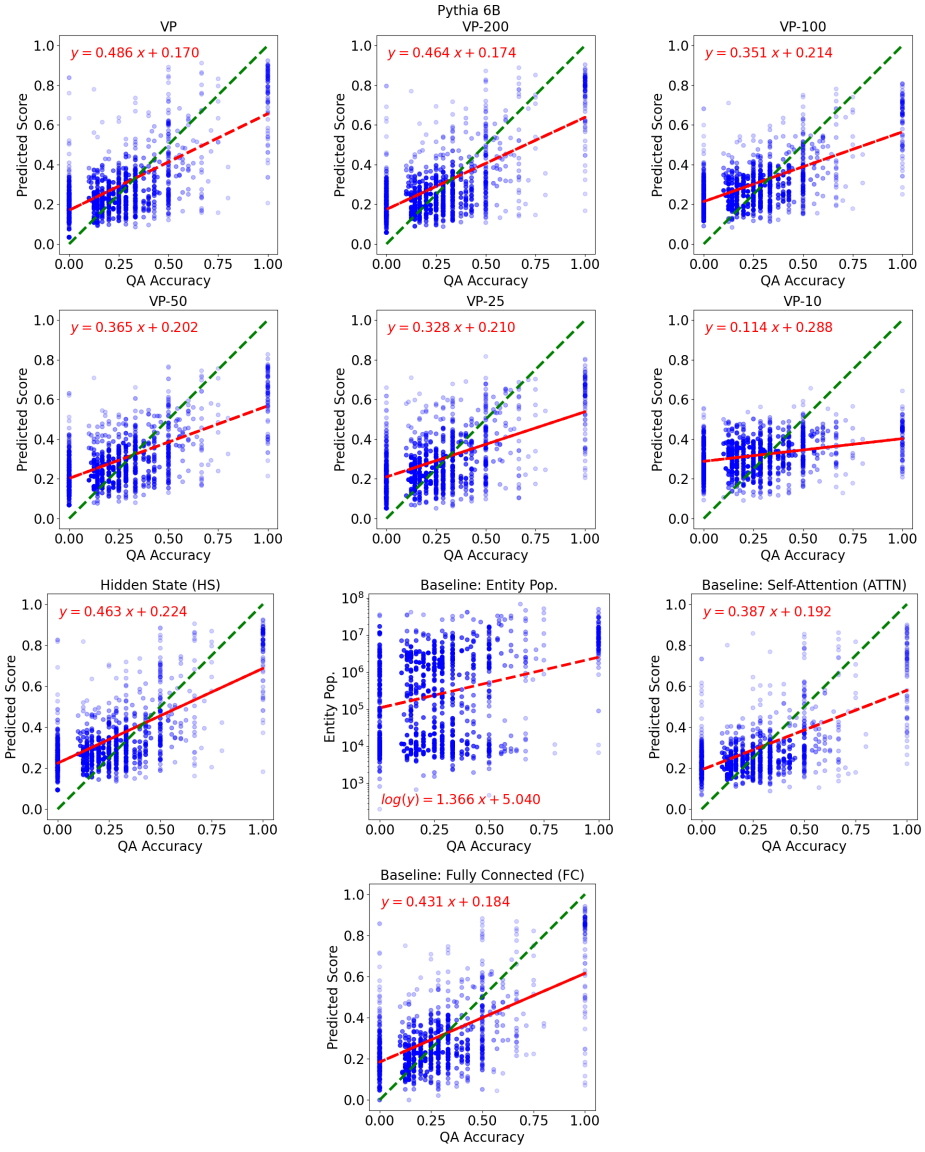

Figure 11: Pythia 6B: Predicted scores from the KEEN QA probe versus the golden QA accuracy scores.

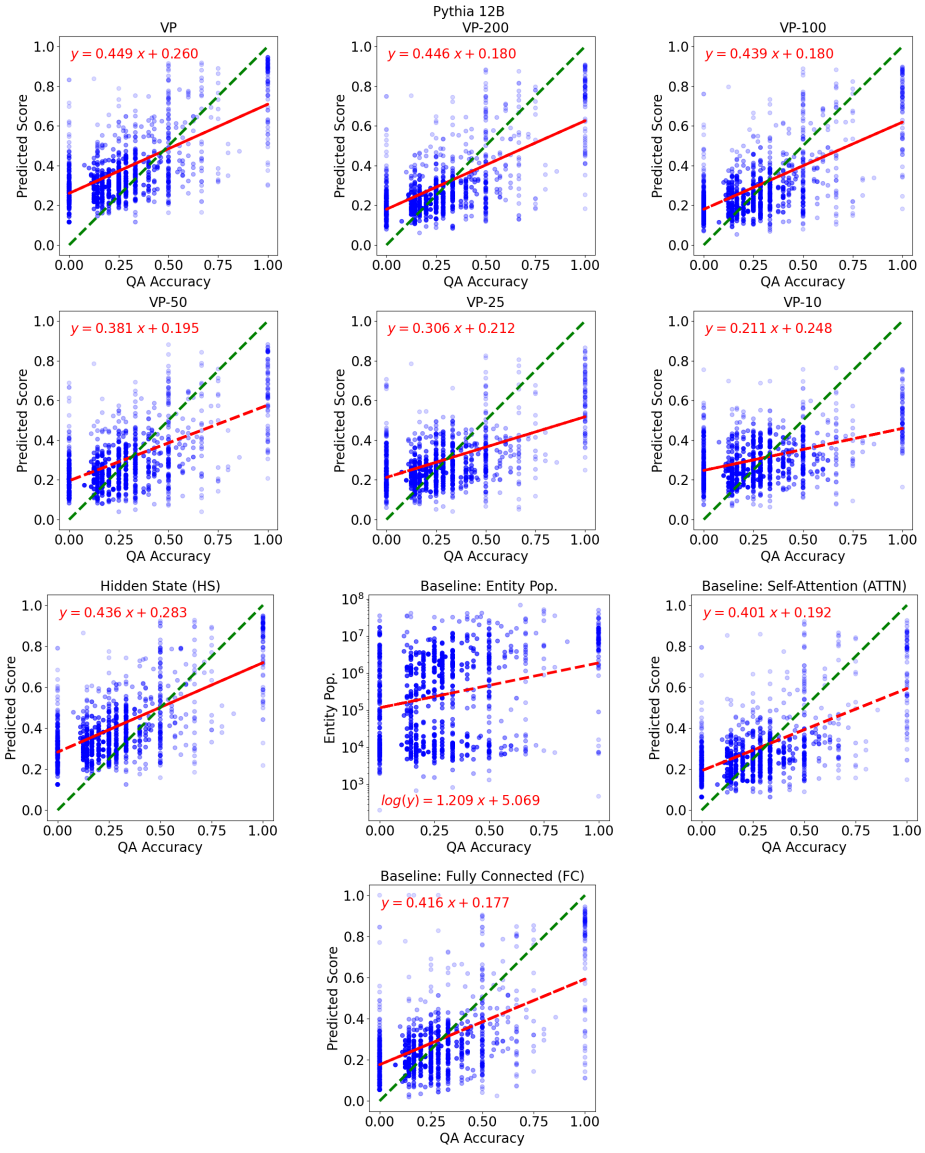

Figure 12: Pythia 12B: Predicted scores from the KEEN QA probe versus the golden QA accuracy scores.

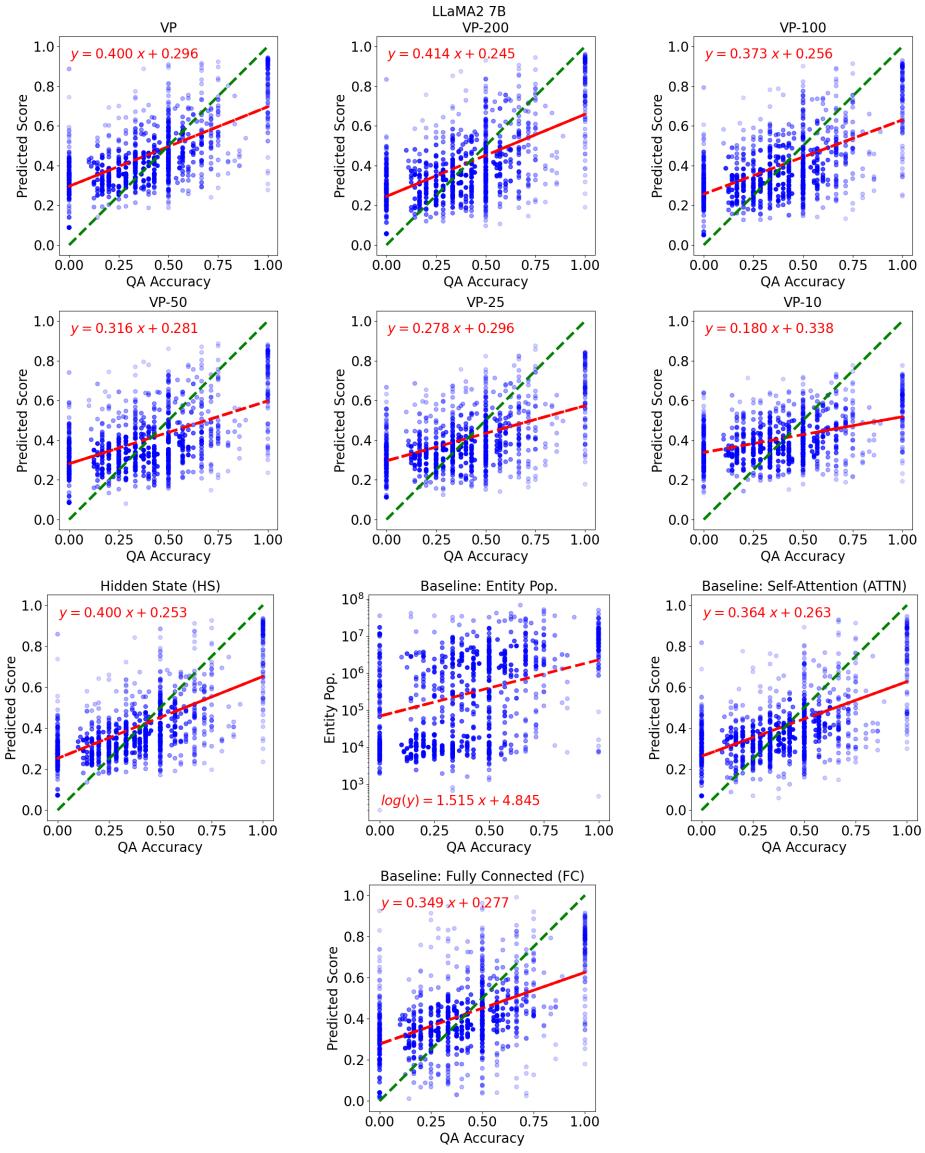

Figure 13: LLaMA 7B, Predicted scores from the KEEN QA probe versus the golden QA accuracy scores.

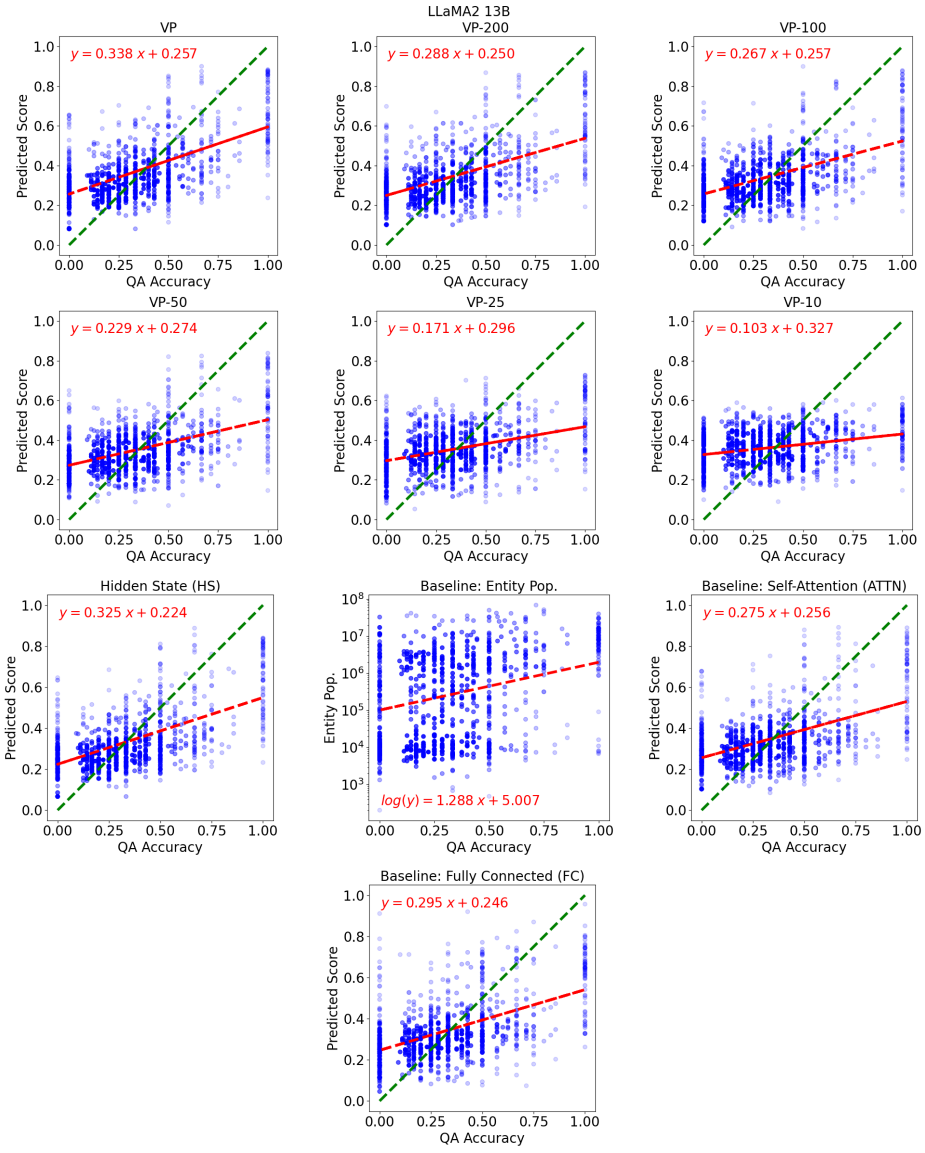

Figure 14: LLaMA 13B: Predicted scores from the KEEN QA probe versus the golden QA accuracy scores.

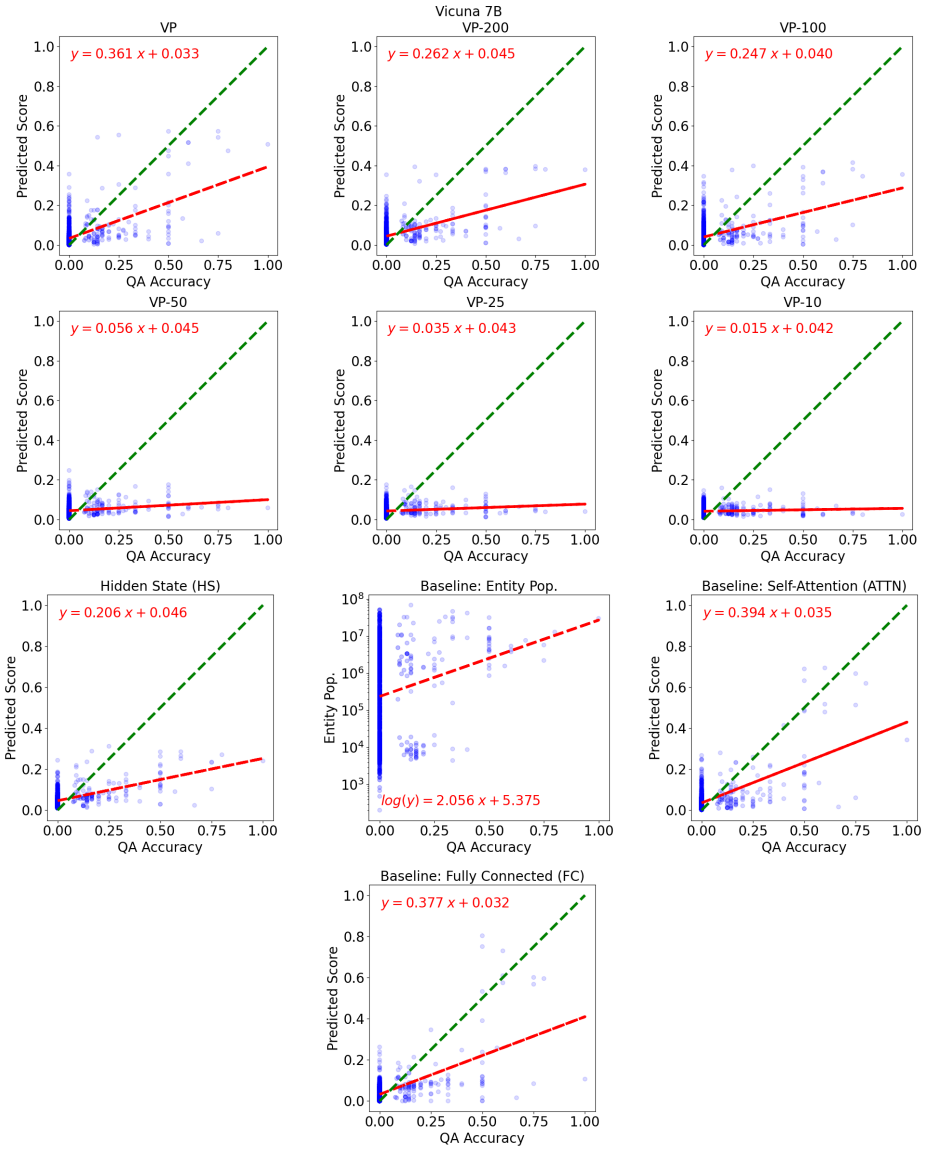

Figure 15: Vicuna 7B: Predicted scores from the KEEN QA probe versus the golden QA accuracy scores.

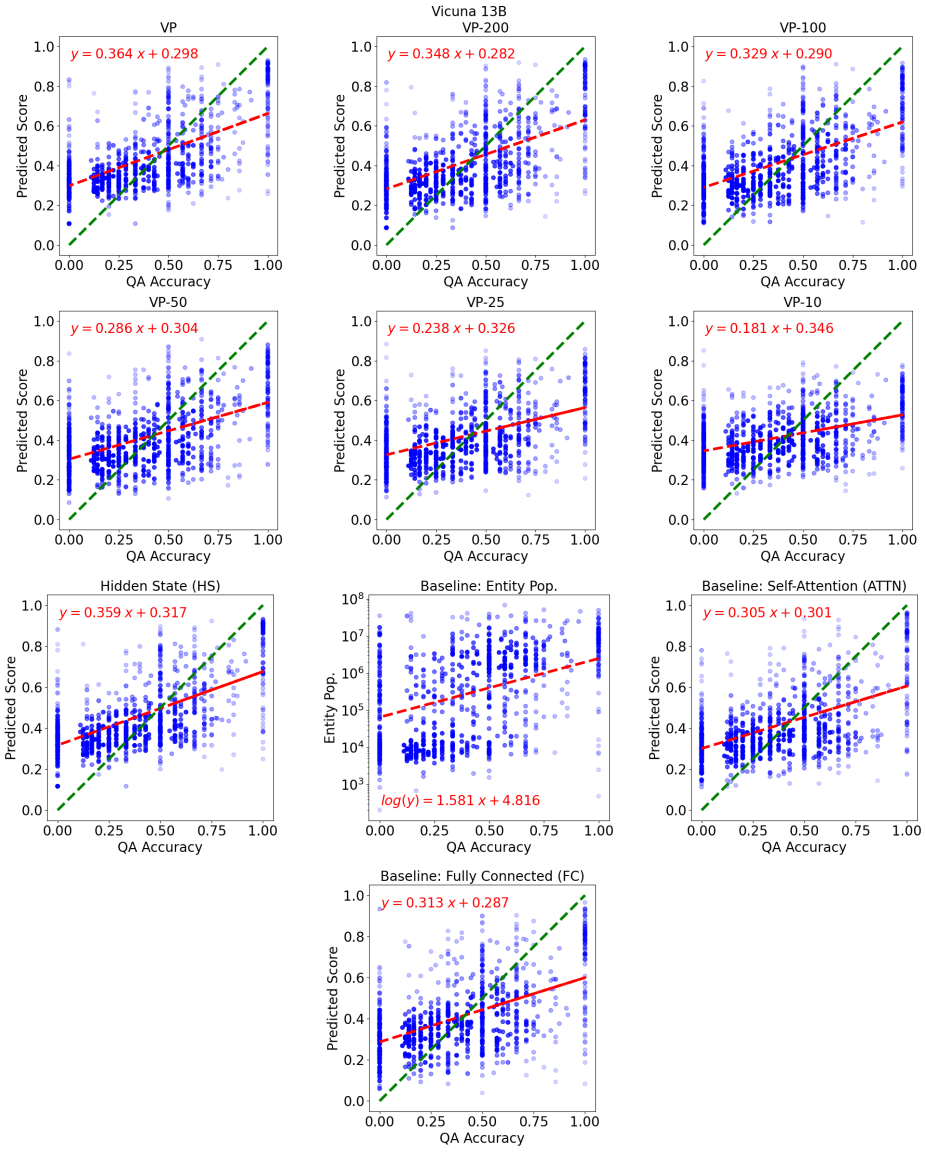

Figure 16: Vicuna 13B: Predicted scores from the KEEN QA probe versus the golden QA accuracy scores.

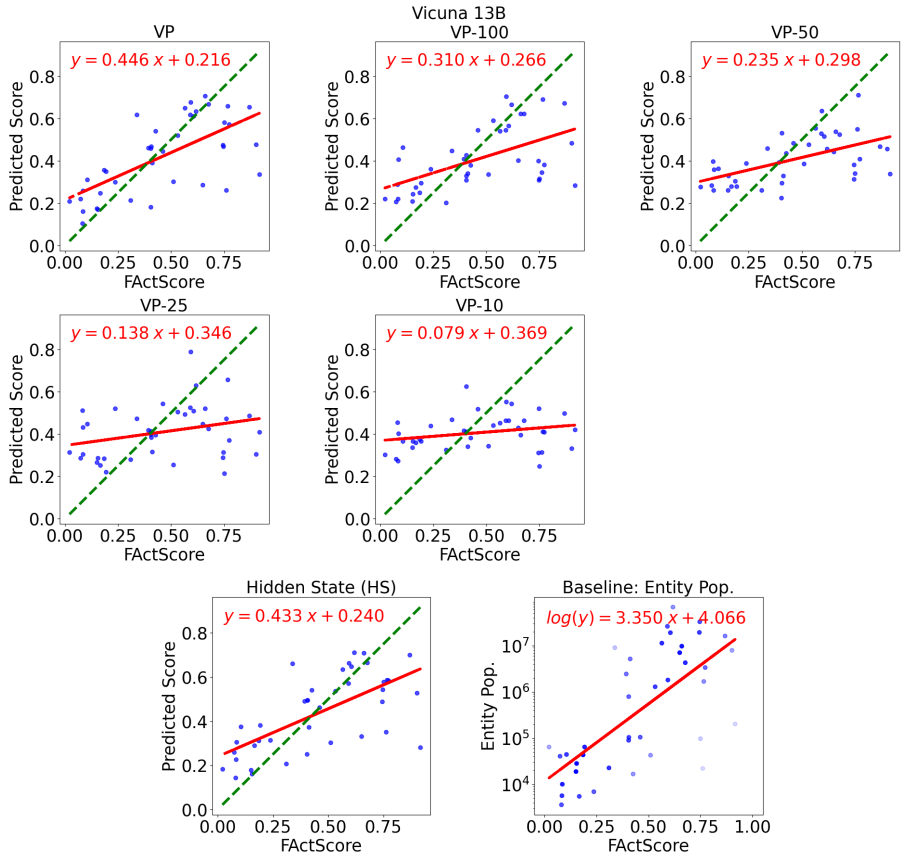

Figure 17: Vicuna 13B: Predicted scores from the KEEN OEG probe versus the golden FActScore scores.

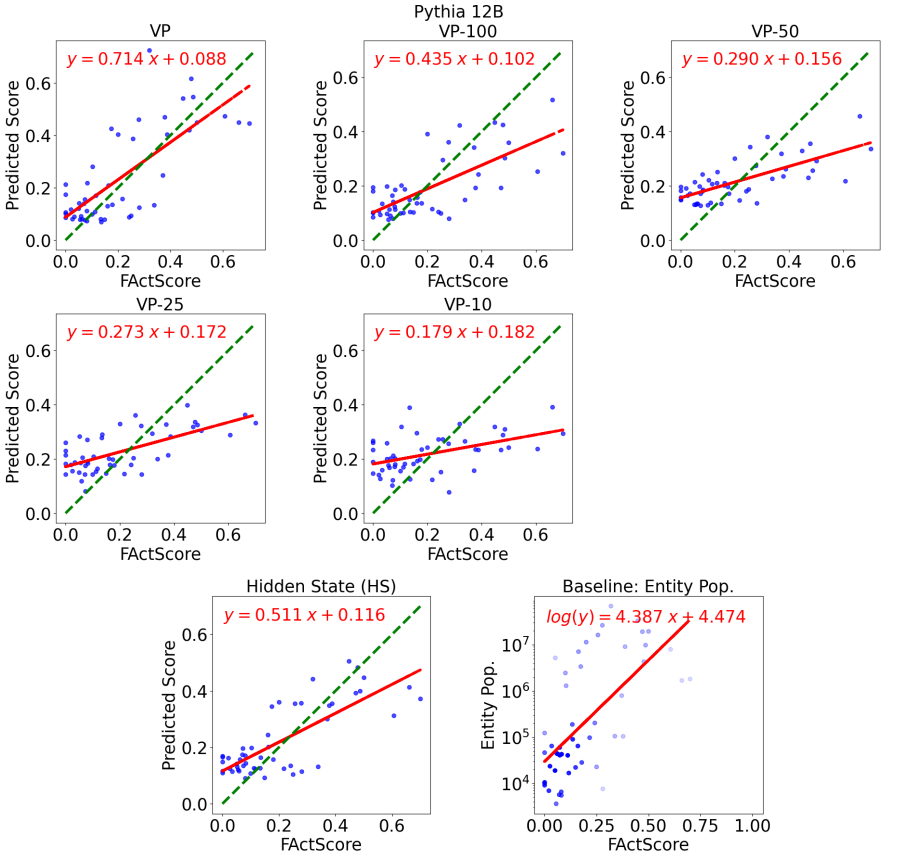

Figure 18: Pythia 12B: Predicted scores from the KEEN OEG probe versus the golden FActScore scores.