Nearest Neighbor Speculative Decoding for LLM Generation and Attribution

Minghan Li[1*], **Xilun Chen**[2] , **Ari Holtzman**[3] , **Beidi Chen**[2,4]
Jimmy Lin[5] , **Wen-tau Yih**[2] , **Xi Victoria Lin**[2]
[1] Cohere [2] Meta FAIR [3] University of Chicago
[4] Carnegie Mellon University [5] University of Waterloo
minghan@cohere.com, aholtzman@uchicago.edu, beidic@andrew.cmu.edu
jimmylin@uwaterloo.ca, {xilun, scottyih, victorialin}@meta.com

Abstract

Large language models (LLMs) often hallucinate and lack the ability to provide attribution for their generations. Semi-parametric LMs, such as kNN-LM, approach these limitations by refining the output of an LM for a given prompt using its nearest neighbor matches in a non-parametric data store. However, these models often exhibit slow inference speeds and produce non-fluent texts. In this paper, we introduce **Ne**arest Neighbor **S**pecula**t**ive Decoding (NEST), a novel semi-parametric language modeling approach that is capable of incorporating real-world text spans of arbitrary length into the LM generations and providing attribution to their sources. NEST performs token-level retrieval at each inference step to compute a semi-parametric mixture distribution and identify promising span continuations in a corpus. It then uses an approximate speculative decoding procedure that accepts a prefix of the retrieved span or generates a new token. NEST significantly enhances the generation quality and attribution rate of the base LM across a variety of knowledge-intensive tasks, surpassing the conventional kNN-LM method and performing competitively with in-context retrieval augmentation. In addition, NEST substantially improves the generation speed, achieving a $1.8\times$ speedup in inference time when applied to Llama-2-Chat 70B. Code will be released at https://github.com/facebookresearch/NEST/tree/main.

1 Introduction

Large language models (LLMs) have demonstrated strong potential as multi-task solvers, excelling in a wide range of applications (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a; Anil et al., 2024). Despite their advanced capabilities, LLMs frequently encounter the problem of hallucination, particularly when dealing with long-tail knowledge that is less represented in their training data (Kandpal et al., 2023; Asai et al., 2023a). To address this limitation, retrieval augmentation incorporates information retrieval and nearest neighbour search from a non-parametric data store to enhance evidence-based and situated reasoning with LLMs. The resulting semi-parametric LMs exhibit a reduced tendency to generate unsupported content (Khandelwal et al., 2020; Borgeaud et al., 2022; Shi et al., 2024a,b; Asai et al., 2023a).

However, the effectiveness of retrieval-augmented language models (RALMs) in ensuring *accurate and reliable content generation* varies. The widely used in-context retrieval-augmentation (RA) regime (Ram et al., 2023; Shi et al., 2024a,b) softly biases the LM output distribution by prepending retrieved content to the input, which does not reliably guarantee faithful attribution of information. Approaches such as kNN-LM (Khandelwal et al., 2020) modify the LM output with a non-parametric

---

[*]Work done during internship at Meta.

token distribution derived from nearest-neighbor matches in a corpus, which provide more direct attribution but has also been shown to degrade the quality of text generation (Wang et al., 2023a). Additionally, retrieval augmentation can significantly increase the *generation latency* due to the time required for the retrieval processes to complete and the subsequent expansion of the LM's context.

In this work, we propose **Ne**arest Neighbor **S**pecula**t**ive Decoding (NEST). This new semi-parametric language modeling approach is capable of incorporating real-world text spans of arbitrary length into the generations of an off-the-shelf LM, leading to improved quality and latency. NEST extends the standard kNN-LM approach, which interpolates the output distribution of an LM using the distribution of possible next tokens retrieved from a corpus (Khandelwal et al., 2020). It conducts an additional passage retrieval step at the beginning to limit the need to store and search over all tokens in the corpus, offering a balanced trade-off between search accuracy and efficiency. At each inference step, NEST performs content generation with three sub-steps:

1) Confidence-based interpolation. We use a novel *Relative Retrieval Confidence* (RRC) score to measure the uncertainty of the token retriever and use it as the interpolation coefficient of the output probability mixture. This enables flexible adaptation of the LM's output to different downstream tasks through dynamic interpolation with the token retrieval results.

2) Dynamic span selection. Inspired by the Copy Generator (COG) (Lan et al., 2023), NEST selects not only the best token predicted by the mixture probability but also extends to the span continuing from that token in the corpus when the token retrieval confidence exceeds a predefined threshold.

3) Relaxed speculative decoding. If a span of more than one token is selected, it undergoes evaluation based on the mixture probability. Through a rejection procedure similar to that in speculative decoding (Leviathan et al., 2023), only a prefix deemed highly likely by the mixture probability is accepted.

Evaluated on various free-form generation tasks—including question answering, text completion, and factuality-aware generation—using Llama-2-Chat models (Touvron et al., 2023b) of different sizes, NEST demonstrates superior performance compared to both the base LM and the standard kNN-LM under a zero-shot setting. For example, combined with NEST, the Llama-2-Chat 70B model demonstrates 42.3% improvement of ROUGE-1 on WikiText-103 and 21.6% improvement of FACTSCORE on Biography. Furthermore, NEST performs competitively with respect to in-context retrieval-augmentation on MMLU, Pile-of-Law, and TruthfulQA. We further demonstrate that the two approaches can be combined to enhance generation quality and attribution. Additionally, by generating multiple tokens at each time step, NEST significantly improves the efficiency of long-form generation. For Llama-2-Chat 70B, it achieves a $1.8\times$ speedup in inference time without compromising attribution or fluency.

2 Background

2.1 Problem Definition

Given an input x, a mixture model $\mathcal{M}$ predicts the output y consisting of segments $\{y_1, y_2, ..., y_T\}$. In our setting, $\mathcal{M}$ may produce multiple tokens at a time step t, and therefore y_t indicates the t-th segment consisting of at most n tokens where $1 \leq |y_t| \leq n$. Let $\{w_t^{(1)}, w_t^{(2)}, ..., w_t^{(n)}\}$ be the tokens in segment y_t, we use $p_{\mathcal{M}}(w|x, y_{<t})$ to denote the distribution of the next token, and use $p_{\mathcal{M}}(w = w_t^{(1)}|x, y_{<t})$ to denote the probability of $w_t^{(1)}$ of the next segment y_t.

2.2 Nearest Neighbor Language Models (kNN-LM)

The mixture model $\mathcal{M}$ involves a pre-trained LM and key-value datastore $(\mathcal{K}, \mathcal{V})$ that enables approximate nearest neighbors search without further training or fine-tuning.

Key-value datastore. To create the datastore $(\mathcal{K}, \mathcal{V})$ using the LM for a corpus $\mathcal{D}$, let $f(\cdot)$ be the mapping from input sequence c to the hidden states h of the LM at some fixed layer. Let w be the next word of c in the corpus $\mathcal{D}$. For a sample (c_i, w_i) in $\mathcal{D}$ after segmentation, we define the i-th key-value pair (k_i, v_i) in $(\mathcal{K}, \mathcal{V})$ as (h_i, w_i), where $h_i = f(c_i)$. The whole datastore is thus defined
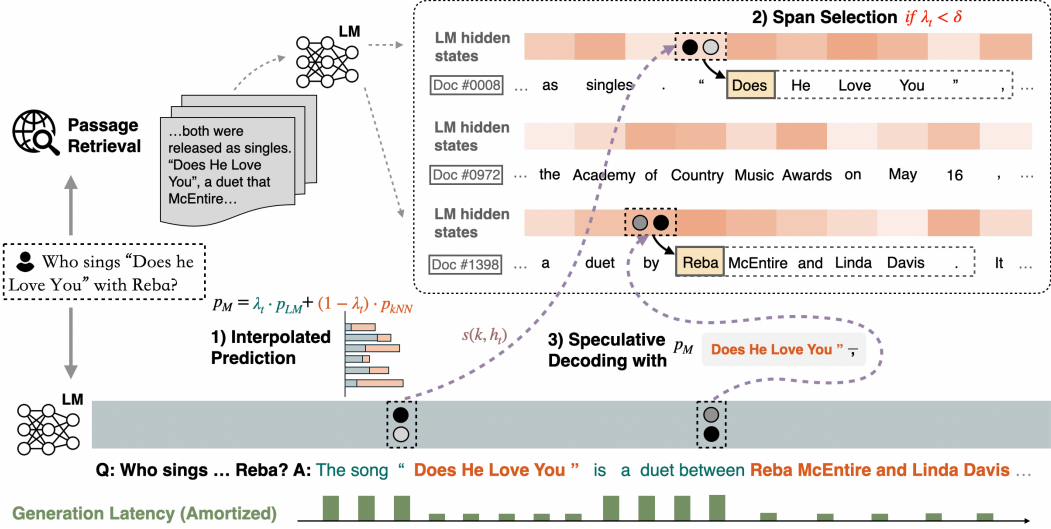
Figure 1: The NEST approach first locates the tokens in the corpus using the LM hidden states. The retrieval distribution $p_{k\text{-NN}}$ is dynamically interpolated with p_{LM} based on the retriever's uncertainty λ_t. The token and its n-gram continuation are then selected from the mixture distribution $p_\mathcal{M}$, while the final span length is determined by speculative decoding to remove undesired tokens. The spans incorporated in the final generation provide direct attribution and amortize the generation latency.

as the set of all possible key-value pairs in $\mathcal{D}$:

$$(\mathcal{K}, \mathcal{V}) = \{(h_i, w_i) | (c_i, w_i) \in \mathcal{D}\}. \tag{1}$$

The size of the datastore $(\mathcal{K}, \mathcal{V})$ is proportional to the total number of tokens in corpus $\mathcal{D}$. This brings difficulty in scaling the size of the corpus and the model, which may require massive storage space and computational resources.

Probability interpolation. During inference, the language model outputs the token distribution $p_{\text{LM}}(w|x, y_{<t})$, together with the hidden state q_t. The model uses q_t as a query to search the datastore $(\mathcal{K}, \mathcal{V})$ and retrieve the r-nearest neighbors π according to the similarity $s(q, k)$ between a query q and a key k. The final non-parametric distribution $p_{k\text{-NN}}(w|x, y_{<t})$ is computed using a softmax function over the similarity of all retrieved neighbors:

$$p_{k\text{-NN}}(w|x, y_{<t}) \propto \sum_{(k_i, v_i) \in \pi} \mathbb{I}_{w=v_i} \exp(\mu \cdot s(q_t, k_i)), \tag{2}$$

where μ is the inverse temperature. We use $1/\sqrt{\dim(q_t)}$ for μ in practice where $\dim(q_t)$ is the hidden state dimension. This is similar to computing attention in the Transformer model (Vaswani et al., 2017). For similarity $s(q, k)$, we follow Khandelwal et al. (2020) and use the negative squared ℓ_2 distance. Items not in π are assigned with 0 probability based on the indicator function $\mathbb{I}_{w=v_i}$.

Finally, the next token is sampled from the mixture distribution $p_\mathcal{M}$ of the non-parametric distribution $p_{k\text{-NN}}$ and the parametric distribution p_{LM} using a fixed hyper-parameter $\lambda \in [0, 1]$:

$$p_\mathcal{M}(w|x, y_{<t}) = \lambda \cdot p_{\text{LM}}(w|x, y_{<t}) + (1 - \lambda) \cdot p_{k\text{-NN}}(w|x, y_{<t}). \tag{3}$$

3 Nearest Neighbor Speculative Decoding

3.1 Two-Stage k-NN Search

As mentioned in Section 2.2, maintaining a token-level key-value store can be expensive in terms of both latency and storage. To provide a better trade-off between latency and accuracy, we adopt the two-stage design, which is widely applied in information retrieval and search engines.

3

First-stage passage retrieval Given the corpus $\mathcal{D}$, we segment the documents into separate passages of less than m tokens each. We then encode the corpus and use a hybrid retrieval system to select the relevant passages, as dense retrievers are good at handling semantics in queries (Karpukhin et al., 2020) and sparse retrievers are good at lexical matching (Sciavolino et al., 2021).

Second-stage k-NN token search After obtaining the top-b retrieved passages $\{d_1, d_2, ..., d_b\}$ at time step t, we use the encoder $f(\cdot)$ of LM to encode the prefixes of all tokens as keys as shown in Figure 1. The key-value datastore $(\mathcal{K}, \mathcal{V})$ therefore is created *on the fly*. Similarly, we use the negative squared ℓ_2 distance as the similarity function and q_t as the queries to search for the top-r nearest neighbors π in $(\mathcal{K}', \mathcal{V}')$.

The two-stage design provides a trade-off between search latency and accuracy and the passage-level index only takes a fraction of the token-level index in Section 2.2. In addition, the first-stage passage search also acts as a filter to remove deceptively similar tokens in non-relevant contexts.

3.2 Confidence-Based Output Interpolation

Similar to Equation (3), we linearly interpolate the language model's distribution p_{LM} and non-parametric distribution $p_{k\text{-NN}}$ using a coefficient λ_t for a time step t in generation. The difference is that we use the token retrieval score to compute λ_t:

$$\lambda_t = \sigma\left(\left(\frac{\min_i |s(q_t, k_i)|}{\max_i |s(q_t, k_i)|} - \alpha\right) / \tau\right), \tag{4}$$

where σ is the sigmoid function and the min-max ratio expresses the uncertainty of the k-NN component. We use the sigmoid activation to re-center and re-scale this uncertainty, where α is the offset and τ is the scale for the sigmoid function. We refer to this method as Relative Retrieval Confidence (RRC).

If the downstream task does not involve generation, such as perplexity evaluation and multi-choice tasks, our method will end at Equation (4). The mechanisms introduced in the following sections are only applied to generation, including token/span selection and post-hoc revision.

3.3 Dynamic Span Selection

Directly sampling tokens from the mixture distribution $p_{\mathcal{M}}$ might escalate the exposure bias since the seemingly coherent tokens might be retrieved from completely different sources. To maintain coherence, we extend the context of the current sampled token by using its n-gram continuation in the corpus. Given the current time step t, we first select the next token w_t from the mixture distribution $p_{\mathcal{M}}$. However, the sampled token w_t may correspond to multiple retrieved w_i (i.e., the value v_i), in the neighbors π which have different n-gram continuations. We use a simple max-pooling strategy[2] to select the starting token $w_t^{(1)}$ of the n-gram from π:

$$w_t^{(1)} = \underset{\{w_i | w_i = w_t, w_i \in \pi\}}{\arg\max} p_{k\text{-NN}}(w = w_i | x, y_{<t}) \tag{5}$$

The corresponding n-gram for time step t is $\{w_t^{(1)}, w_t^{(2)}, ..., w_t^{(n)}\}$ where n is fixed hyper-parameter. The final output is determined by the interpolation coefficient λ_t in Equation (4):

$$y_t = \begin{cases} w_t, & \text{if } \lambda_t > \delta; \\ \{w_t^{(1)}, w_t^{(2)}, ..., w_t^{(n)}\}, & \text{otherwise.} \end{cases} \tag{6}$$

where δ is a threshold and y_t is the segment output at time step t.

3.4 Relaxed Speculative Decoding

Despite the dynamic selection, the hyper-parameter n is hard to control over different tasks. To produce spans with adaptive length, we take inspiration from Leviathan et al. (2023), where we use $\mathcal{M}$ to revise the proposed n-gram. However, the proposal distribution $q(w|x, y_{<t})$ is unknown

---

[2]We used a slightly different implementation to ensure the sampled token is in π. Please see the code here: https://github.com/facebookresearch/NEST/blob/main/models/knn_transformers.py

besides the first token $w_t^{(1)}$. Therefore, we use a relaxed version of speculative decoding that upper bounds the acceptance probability. The probability of accepting the token $w_t^{(i)}$ in a span is:

$$P(\text{accept token } w_t^{(i)}) = \min\left(1, \frac{p_{\mathcal{M}}(w = w_t^{(i)} \mid x, y_{<t}, w_t^{(1)}, w_t^{(2)}, ..., w_t^{(i-1)})}{\gamma \cdot \max_w p_{\mathcal{M}}(w \mid x, y_{<t}, w_t^{(1)}, w_t^{(2)}, ..., w_t^{(i-1)})}\right), \quad (7)$$

where $\gamma \in (0, 1]$ is the relaxation factor, which is referred to as "leniency" by Leviathan et al. (2023). The smaller γ is, the less often $\mathcal{M}$ rejects the draft. If token $w_t^{(i)}$ is rejected, we will remove all the tokens from $w_t^{(i)}$ to $w_t^{(n)}$, and then re-select a token $w_t^{(i)}$ from the distribution $p_{\mathcal{M}}$ without going through the span selection. The computation for processing multiple tokens can be parallelized and NEST can thus maintain the latency or even accelerate the generation. Moreover, suppose all tokens in the draft are not rejected. In that case, we will directly fetch the n-gram's continuation in the corpus and use it for the next draft proposal until rejection, removing the reliance on the hyper-parameter n.

Once the n-gram is accepted, the corresponding parts are masked in the corpus and will never be used again in this generation. This is to prevent the k-NN component from repetitively retrieving the same segments in a small key-value store $(\mathcal{K}', \mathcal{V}')$. We provide the complete procedure in Algorithm 1.

4 Experiments

We evaluate NEST and other baselines on various tasks including text completion, question-answering, fact-verification, and multi-choice tasks, providing a comprehensive picture of factuality, fluency, and attribution of NEST in different domains. In all experiments, we focus on evaluating instruction-following models. We use Llama-2-chat under a zero-shot setting, where we remove the few-shot demonstrations from the instructions to simulate the realistic usage of these models.

4.1 Benchmark Datasets

Text completion. **WikiText-103** (Merity et al., 2017) is a standard benchmark for language modeling, extracted from the set of verified articles on Wikipedia. **Pile of Law** (Henderson et al., 2022) is a growing dataset of legal and administrative data. We use the datasets[3] from Huggingface and further split the test data into validation and test sets. For language modeling, we report the perplexity score. For free-form generation, we report ROUGE-1, 2, L (Lin, 2004) and MAUVE (Pillutla et al., 2021).

Question answering. We select four knowledge-intensive question-answering datasets, including Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), HotpotQA (HQA) (Yang et al., 2018), and MedMCQA (MQA) (Pal et al., 2022). Since the in-context demonstrations are removed for free-form generation, we use answer-level recall (i.e., Hit@1) (Karpukhin et al., 2020) which checks if the output contains any correct answers instead of exact match.

Fact verification. We evaluate a biography-generation task (Min et al., 2023) and Truthful-QA (Lin et al., 2022) which is a benchmark for testing false beliefs or misconceptions. We use FACTSCORE (Min et al., 2023) for biography. For TruthfulQA, we follow Lin et al. (2022) which uses the difference between the max similarity to a true reference answer and the max similarity to a false reference answer for BLEU and ROUGE-1.

Closed-set tasks. MMLU (Massive Multitask Language Understanding) (Hendrycks et al., 2021) benchmark covers 57 subjects across STEM, the humanities, the social sciences, and more. We report the macro accuracy for each domain.

4.2 Implementation

Knowledge Sources. Wikipedia (CC BY-SA 3.0): For tasks except text completion on Pile of Law, we use the Wikipedia 2021 dump released by Izacard et al. (2024) as the knowledge source and follow the same pre-processing procedures in RA-DIT (Lin et al., 2024), yielding ~33M passages with each

---

[3] https://huggingface.co/datasets/pile-of-law/pile-of-law/tree/main

less than 200 tokens. Pile of Law (CC BY-NC-SA 4.0): We use the training split from Huggingface and select only the English data. We then follow the same procedure applied in Wikipedia, yielding a corpus containing ∼15M passages after filtering. More details are provided in Appendix A.

Inference setting. kNN-LM and NEST share the same first-stage retriever. We use DRAGON+ (Lin et al., 2023) and BM25 (Robertson and Zaragoza, 2009) to encode the segments into dense and sparse vectors, respectively. Given the input, we query both the dense and sparse indexes at the same time and retrieve their corresponding top-$(b \cdot l)$ passages. We linearly interpolate the similarity scores between the two search results (also known as fusion) and sort them before selecting the top-b passages. The number of passage candidates b is set to be 40 and the scaling factor l is set to be 100. For RA, we use the top-3 passages in the prompt due to the context window limit. We further combine NEST and RA since they are independent methods. Greedy decoding is used during generation. More details about retrieval, decoding, and hyper-parameters are described in Appendix B.

Evaluation setting.   For text completion tasks and perplexity evaluation, we use 128 tokens as the prefix and the consecutive 256 tokens as the target. For the other tasks, we use 128 tokens as the max generation length for question answering and 512 for fact verification. For retrieval-based models, only the prefix will be used for retrieval. Hyper-parameters of all baselines and NEST are tuned on the dev set of WikiText-103, NQ, and Biography. Each baseline uses the same hyper-parameters for all tasks evaluated. We first tune the related hyper-parameters for perplexity and then tune the rest for generation metrics to reduce the search space. More details are provided in Appendix B.

4.3   Baselines

Base LMs.   We evaluate publicly available, instruction-tuned language models, Llama-2-chat series[4], with model sizes ranging from 7B, 13B to 70B.

Two-Stage kNN-LM.   We apply the two-stage strategy described in Section 3.1 to kNN-LM as well, where we retrieve the top-b passages and encode a key-value datastore $(\mathcal{K}', \mathcal{V}')$ on the fly.

In-Context Retrieval Augmentation (RA).   A common retrieval-augmentation method is adding the retrieved evidence into the prompt. We perform retrieval given the only input instead of retrieving new passages every k step due to the expense of refreshing the kv-cache.

4.4   Main Results

Table 1 shows the main results of NEST and other baselines. For **language modeling**, RA-NEST is able to achieve the lowest complexity on both WikiText-103 and Pile of Law. For **text completion**, RA has the best MAUVE scores and ROUGE scores in Wikitext-103 while RA-NEST works better for 7B and 13B models on Pile of Law. We observe that for legal documents, quoting the exact clauses from the source might be more favourable compared to Wikipedia.

For **question-answering**, RA-NEST tends to work better for smaller models (7B and 13B) in general. The gap between base LMs and other methods diminishes for 70B LMs, which is consistent with previous work where retrieval is found most useful for smaller models (Borgeaud et al., 2022).

For **fact-verification**, NEST is able to outperform the base LMs but underperform RA in terms of the FACTSCORE. RA-NEST is able to outperform RA for the 70B model. The degradation for RA-70B is caused by generating shorter claims which is punished by the FACTSCORE. On TruthfulQA, the semi-parametric LMs consistently outperform base LMs and RAs where in-context retrieval seems to have a negative effect on the scores. This is because TruthfulQA is an adversarial dataset containing difficult questions where in-context RA is more susceptible to the "evidence" in the prompt (e.g., astrology and myths). In contrast, NEST only interpolates the results at the output level and therefore performs better in this case. The combination RA-NEST is also affected by the in-context retrieval.

For **closed-set tasks**, NEST is comparable to RA and RA-NEST manages to achieve the best macro scores on average. Overall, NEST is able to outperform base LMs and kNN-LM's on most tasks while being on par with RA. The combination of RA and NEST further improves over the two methods

---

[4] https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

Models	Wikitext-103						Pile of Law					
	PPL(↓)	MAUVE	RG-1	RG-2	RG-L	Avg. Len	PPL(↓)	MAUVE	RG-1	RG-2	RG-L	Avg. Len
Llama-2-Chat$_{7B}$	14.6	58.8	15.8	3.7	14.4	175.4	10.1	80.7	19.1	5.5	17.1	211.4
+RA	7.2	74.6	**35.7**	**23.1**	**34.4**	204.5	7.1	84.7	23.1	8.9	21.1	222.0
+kNN-LM	9.8	**82.5**	23.7	7.7	21.7	**238.2**	8.8	81.1	19.4	5.7	17.4	214.3
+NEST	8.4	73.2	28.4	14.2	27.1	218.4	8.1	88.0	23.7	8.7	21.5	226.5
+RA-NEST	**6.4**	72.6	35.2	22.7	34.0	202.0	**6.7**	**90.0**	**24.4**	**9.0**	**22.2**	**232.1**
Llama-2-Chat$_{13B}$	12.0	75.9	19.9	4.9	18.0	218.4	8.2	72.8	17.5	5.3	15.7	181.7
+RA	6.5	**91.5**	**38.9**	**24.2**	**37.2**	**249.3**	5.9	86.6	23.6	9.1	21.5	228.7
+kNN-LM	8.6	76.3	23.7	8.2	21.9	238.5	7.4	71.5	17.7	5.3	15.9	183.7
+NEST	7.2	67.1	29.3	15.6	28.1	207.1	6.8	86.0	22.9	8.7	20.9	212.3
+RA-NEST	**5.8**	86.8	38.6	24.0	37.0	245.5	**5.7**	**90.1**	**24.7**	**9.2**	**22.4**	229.4
Llama-2-Chat$_{70B}$	9.9	88.6	22.9	6.2	20.8	239.6	6.9	93.4	23.0	7.1	20.7	250.1
+RA	5.3	91.6	**40.5**	**26.1**	**38.8**	235.9	4.9	95.5	**26.3**	**10.1**	**24.0**	253.2
+kNN-LM	7.1	83.6	26.1	9.6	24.1	**253.9**	6.3	94.4	23.1	7.2	20.8	251.3
+NEST	6.3	82.6	32.6	17.2	31.1	236.3	5.9	95.4	25.6	9.4	23.2	251.3
+RA-NEST	**4.8**	90.0	40.2	25.9	38.6	233.1	**4.7**	**97.6**	26.2	9.5	23.7	**253.6**

| Models	TQA	NQ	HQA	MQA	Avg.	TruthfulQA		Biography		MMLU				
			Answer-Level Recall			ΔBLEU	ΔRG-1	FS	# Facts	Human.	STEM	Social	Other	Avg.
Llama-2-Chat$_{7B}$	61.1	38.9	30.6	9.3	35.0	-0.02	0.42	27.2	**71.2**	37.8	32.6	38.9	39.6	37.2
+RA	**69.5**	48.4	44.1	12.8	43.7	-0.34	0.18	56.5	67.1	41.8	35.3	**42.2**	43.3	40.7
+kNN-LM	63.4	42.4	33.5	9.5	37.2	**0.13**	**0.66**	30.6	59.8	38.0	33.1	39.2	40.1	37.6
+NEST	61.5	43.2	33.5	10.2	37.1	0.03	0.45	38.9	58.2	**42.0**	**35.4**	42.0	**43.4**	**40.7**
+RA-NEST	69.0	**48.8**	**45.3**	**13.3**	**44.1**	-0.32	0.21	55.1	57.7	37.9	32.7	39.3	39.8	37.4
Llama-2-Chat$_{13B}$	63.5	42.3	32.6	10.2	37.2	0.13	0.81	28.8	49.9	41.5	35.0	40.2	43.8	40.1
+RA	70.9	51.6	44.6	14.0	45.3	-0.16	0.25	**59.1**	51.2	43.4	37.4	43.5	46.4	42.7
+kNN-LM	64.7	43.5	34.2	11.2	38.4	0.20	0.95	31.1	46.1	41.4	34.7	40.6	44.2	40.2
+NEST	64.2	44.2	34.3	10.9	38.4	**0.29**	**0.98**	35.7	47.2	41.3	34.9	40.2	43.7	40.0
+RA-NEST	**70.9**	**51.7**	**45.3**	**14.7**	**45.7**	-0.14	0.25	58.4	52.4	**43.5**	**37.7**	**43.5**	**46.7**	**42.8**
Llama-2-Chat$_{70B}$	74.0	50.1	39.5	12.8	44.1	0.14	0.70	34.2	**58.8**	43.5	37.9	44.4	47.0	43.2
+RA	**75.5**	**55.4**	**52.5**	16.0	**49.9**	-0.13	0.40	52.9	42.1	**45.9**	39.7	46.2	48.6	45.1
+kNN-LM	74.6	51.2	40.2	13.5	44.9	0.08	0.58	36.1	54.4	44.0	37.4	44.1	47.1	43.2
+NEST	74.2	51.6	41.4	13.8	45.2	**0.17**	**0.70**	41.6	56.2	43.8	38.0	44.4	47.8	43.5
+RA-NEST	75.4	55.2	52.4	**16.3**	49.8	-0.19	0.31	**59.2**	53.8	45.8	**39.7**	**46.2**	**48.9**	**45.1**

Table 1: Results on text completion (upper table) and other tasks (lower table). Bold numbers indicate the best performance. PPL: Perplexity. RG: ROUGE score. Avg. Len: Average generation length. ΔBLEU/ΔRG: The difference between the max score to correct references and the max score to incorrect references. FS: FACTSCORE with length penalty.

on some tasks. Despite the limited improvement, we will show that NEST is able to provide better attribution and latency in the following sections.

4.5 Latency Analysis

Latency breakdown. The combination of dynamic span selection and relaxed speculative decoding can improve the latency of the LLM generation by quick draft proposal and processing multiple tokens at a time step. Figure 2a shows the latency breakdown of a NEST-70B model ($\alpha = 0.3, \tau = 0.1, \delta = 0.5$) for different relaxation factors on the Biography validation data. The latency experiment is done on 8×A100 GPUs (for model parallelization) and 32 CPU threads (for search). The batch size is set to 1. We use internal, research-purpose implementation of the base Llama-2-chat model which did not optimize for latency. As we can see, the LM encoding time takes about half of the latency, while the sum of the others takes the rest. Noticeably, the cost of passage search and token index building stay relatively constant per query, while the others are related to the number of tokens processed per time step. Still, even with extra retrieval overheads, the slowest NEST model is faster than the base LM, showing the efficacy of span selection and speculative decoding.

Latency-accuracy trade-off. To understand why NEST can accelerate generation, we first show the latency-accuracy trade-off by tuning the relaxation factor in Figure 2b. The smaller γ is, the less often NEST rejects a segment retrieved from the corpus, which enables more tokens to be processed in parallel. The average proposed span length in Figure 2a can increase from 5 tokens to 35 tokens at each time step as the relaxation factor gets smaller. Combined with Figure 2a, we can reach the conclusion that fetching longer spans from the corpus results in lower generation latency per query. For the accuracy, the FACTSCORE on Biography validation data shows that there is a sweet spot around $\gamma = 5e - 2$ where both low latency and high accuracy can be achieved at the same time.
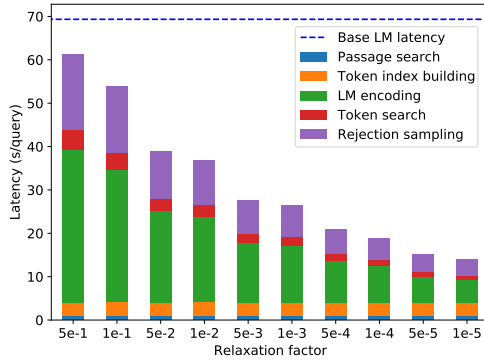
(a) Latency Breakdown.
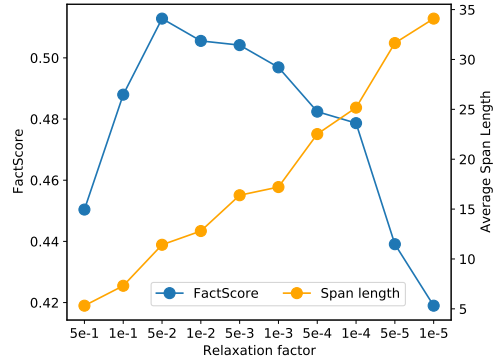
(b) Span length and FACTSCORE on Biography.

Figure 2: Latency-accuracy trade-off and breakdown on Biography using Llama-2-Chat 70B+NEST. As the relaxation factor γ decreases, NEST tends to accept longer spans from the corpus. We choose $\gamma = 5e-2$ in our main experiments, which accelerates the generation and improves the FACTSCORE.

Task	Models	Ratio (%)	Length	Example (Generated by NEST-70B)
NQ	NEST 7B	93.4	18.4	Question: who sings does he love me with reba?
	NEST 13B	52.4	4.4	Answer: The song "Does He Love You" [1] is a duet
	NEST 70B	58.8	4.3	between Reba McEntire and Linda Davis. It was
	RA-NEST 7B	77.5	3.1	released in August 1993 as the first single from Reba's
	RA-NEST 13B	33.2	3.0	compilation album Greatest Hits Volume 2. It is one of
	RA-NEST 70B	44.9	3.9	country music's several songs about a love triangle. [2]
Bio				Question: Tell me a bio of Taral Hicks.
				Answer: Taral Hicks (born September 21, 1974) is an American actress
	NEST 7B	95.5	27.9	and singer. Hicks is best known for her acting in such films as 1993's
	NEST 13B	53.9	10.6	American crime drama film A Bronx Tale and her singing in such work
	NEST 70B	58.6	7.0	as her 1997 [3] debut studio album This Time which peaked at No. 4 on
	RA-NEST 7B	50.3	5.1	Billboard Bubbling Under Hot 100 Singles. Raised in Teaneck,
	RA-NEST 13B	48.5	5.9	New Jersey, Hicks graduated from Teaneck High School in 1994 [3].
	RA-NEST 70B	80.7	11.0	She is the younger sister of actress and singer D'atra Hicks.

Table 2: Attribution analysis. (Attribution) Ratio: Proportion of tokens that are taken from the corpus. (Attribution) Length: Average length of consecutive spans in the generation that are taken from the same document. Green: Segments taken from the corpus. Gray: Reference.

4.6 Attribution and Qualitative Analysis

One of the most important features of NEST is providing attribution directly at a span level, where the reference for the corresponding statement is accurate since it is directly taken from the corpus. Table 2 shows the attribution ratio, average attributed span length, and two examples for analysis. For NQ and Biography tasks, depending on the model and hyper-parameters in Equation (4) and (7), the ratio of tokens that can be traced back to the corpus ranges from 33.2% to 95.5%. In addition, it is more desirable to have consecutive segments that come from the same source so that consistent attribution can be provided, and the average length of spans taken from the corpus ranges from 3.0 to 27.9 tokens. This feature provides span-level attribution for most claims in the LLM generation. To our knowledge, neither of the baselines can achieve the same granularity and preciseness for the attribution as NEST. We provide more analyses on sensitivity and ablation for NEST in Appendix C.

5 Related Work

5.1 Retrieval-Augmentation

Retrieval Augmentation involves external knowledge sources to improve the effectiveness of language models on knowledge-intensive tasks. Chen et al. (2017) propose DrQA which combines extractive models and independent retrievers for open-domain question-answering. Follow-up works on retrieval-augmentation such as REALM (Guu et al., 2020), RAG (Lewis et al., 2020), and Atlas (Izacard

et al., 2024) further combine the retrieval component in pre-training and fine-tuning for downstream knowledge-intensive tasks. Asai et al. (2024) further divide them into three categories:

Input augmentation. REPLUG (Shi et al., 2024a) and in-context RALM (Ram et al., 2023) propose to pre-pend the retrieved passages in the prompts for zero-shot factual support. Recently, Self-RAG (Asai et al., 2023b) leverages special tokens to perform adaptive retrieval and different critics to iterative refine the RALM's output. RA-DIT (Lin et al., 2024) retrofits LLMs with retrieval capabilities via instruction fine-tuning.

Intermediate fusion. RETRO (Borgeaud et al., 2022) employs a novel attention mechanism to incorporate multiple pre-processed text fragments in intermediate layers for more efficient integration of retrieved results. This approach has been successfully applied to larger decoder-only language models as demonstrated by RETRO++ (Wang et al., 2023b) and InstructRetro (Wang et al., 2024). FiD (Izacard and Grave, 2021) applies similar an encoder-decoder structure in a zero-shot manner and achieves better effectiveness at a document level.

Output integration. kNN-LM (Khandelwal et al., 2020) pioneers this direction and proposes to interpolate the retrieval distribution and LM's prediction. Follow-up works further propose adaptive interpolation methods which involve training (He et al., 2021; Bhardwaj et al., 2023) and excessive tuning (Drozdov et al., 2022). Another line of work proposes to joint train the phrase encoder and LM to expand the vocabulary dynamically using the retrieved phrases, such as Copy-Generator (Lan et al., 2023) and its follow-up work (Cao et al., 2024). Martins et al. (2022) proposes a chunk-based kNN machine translation model which retrieves chunks of tokens from the datastore.

5.2 Inference-Time Revision

Speculative decoding (Leviathan et al., 2023; Chen et al., 2023; Miao et al., 2023; Spector and Re, 2023) is an acceleration method that leverages a small model to generate drafts for a large model to evaluate. The latency is improved as the larger model can process multiple tokens in parallel at each time step. Recently, REST (He et al., 2024) proposes to draw multiple drafts from a datastore and leverages a prefix trie tree to compute the proposal distribution, which is the closest concurrent work. Yang et al. (2023) also utilizes prefix matching to select draft sentences from a datastore, and keep the continuation of the draft sentence as long as the token matches with the model generation.

In general, speculative decoding can be categorized as an unbiased self-revision method. In comparison, NEST changes the LM output distribution through interpolation with a non-parametric probability distribution. Previous work focusing on fact-checking follows a similar idea to generate factually consistent texts with a set of evidence via post-hoc editing, such as FRUIT (Iv et al., 2022) and PEER (Schick et al., 2022). Recently, RARR (Gao et al., 2023) leverages more complex planning with LLMs to verify the retrieved evidence and generate attribution reports.

6 Limitations

While being able to directly retrieve segments from the corpus and apply them in the generation, the output of NEST might still contain factual errors depending on the accuracy of the first-stage passage retrieval and the second-stage token retrieval. Moreover, as a plug-and-play method, our main goal is to provide a flexible solution that can combine different LLMs and data stores in zero- and few-shot manners. Without further fine-tuning, the integrated system might be sub-optimal and the results can be better if it is fine-tuned on appropriate tasks. Lastly, such semi-parametric LMs may not improve the ability of in-context learning, since the demonstrations in the prompts are unlikely to appear in any contexts that can be found in the database. An observation from preliminary experiments is that the current neural retrievers do not have the capability to process the in-context few-shot information, where techniques such as query reformulation might be needed for parsing the demonstrations.

7 Conclusion

This paper presents NEST, an inference-time revision method for LMs that improve their factuality and attribution through nearest neighbor speculative decoding. Leveraging two-stage k-NN search, relative retrieval confidence, dynamic span selection, and relaxed speculative decoding, NEST improves both validation perplexity and free-form generation quality on nine different tasks. Its

effectiveness can be further improved when combined with in-context retrieval augmentation. With these results, we demonstrate that NEST is capable of generating text grounded to real-world sources in low latency while maintaining fluency.

8 Broader Impact

The ability to copy real-world texts from existing data stores is useful for finding the source of the claim (credibility), preventing hallucination (factuality), as well as protecting copyright (risk management). It helps to resolve the dispute that often happens in AI tools by acknowledging the contents that are borrowed from existing human works (e.g., arts, books, and other creative content). Meanwhile, the information on the Internet is mixed and it is important to filter out false and sensitive information before directly injecting them into the generation.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

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

Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James

Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg,

James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-

David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srini-vasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham

Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2024.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46, Toronto, Canada, July 2023a. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations (ICLR)*, 2020.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*, 2022.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2024a.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2024b.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

Shufan Wang, Yixiao Song, Andrew Drozdov, Aparna Garimella, Varun Manjunatha, and Mohit Iyyer. *k*NN-LM does not improve open-ended text generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15023–15037, Singapore, December 2023a. Association for Computational Linguistics.

Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. Copy is all you need. In *The Eleventh International Conference on Learning Representations*, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR, 23–29 Jul 2023.

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. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Peter Henderson, Mark Simon Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel E. Ho. Pile of law: Learning responsible data filtering from the law and a 256GB open-source legal dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1), mar 2024. ISSN 1532-4435.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. RA-DIT: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore, December 2023. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and Wen tau Yih. Reliable, adaptable, and attributable language models with retrieval. *arXiv preprint arXiv:2403.03187*, 2024.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023b.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, Anima Anandkumar, and Bryan Catanzaro. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7763–7786, Singapore, December 2023b. Association for Computational Linguistics.

Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. Instructretro: Instruction tuning post retrieval-augmented pretraining. *arXiv preprint arXiv:2310.07713*, 2024.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online, April 2021. Association for Computational Linguistics.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

Rishabh Bhardwaj, George Polovets, and Monica Sunkara. Adaptation approaches for nearest neighbor language models. *arXiv preprint arXiv:2211.07828*, 2023.

Andrew Drozdov, Shufan Wang, Razieh Rahimi, Andrew McCallum, Hamed Zamani, and Mohit Iyyer. You can't pick your neighbors, or can you? when and how to rely on retrieval in the kNN-LM. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2997–3007, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Bowen Cao, Deng Cai, Leyang Cui, Xuxin Cheng, Wei Bi, Yuexian Zou, and Shuming Shi. Retrieval is accurate generation. *arXiv preprint arXiv:2402.17532*, 2024.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. Chunk-based nearest neighbor machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4228–4245, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. Specinfer: Accelerating generative llm serving with speculative inference and token tree verification. *arXiv preprint arXiv:2305.09781*, 2023.

Benjamin Spector and Chris Re. Accelerating llm inference with staged speculative decoding. *arXiv preprint arXiv:2308.04623*, 2023.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. Rest: Retrieval-based speculative decoding. *arXiv preprint arXiv:2311.08252*, 2024.

Nan Yang, Tao Ge, Liang Wang, Binxing Jiao, Daxin Jiang, Linjun Yang, Rangan Majumder, and Furu Wei. Inference with reference: Lossless acceleration of large language models, 2023. URL https://arxiv.org/abs/2304.04487.

Robert Iv, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. FRUIT: Faithfully reflecting updated information in text. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3670–3686, Seattle, United States, July 2022. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. Peer: A collaborative language model. *arXiv preprint arXiv:2208.11663*, 2022.

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

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, 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. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada, July 2023. Association for Computational Linguistics.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. ExpertQA: Expert-curated questions and attributed answers. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3025–3045, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

A   Additional Implementation Details

Two-stage k-NN search   For the first-stage passage search, we use a Faiss (Douze et al., 2024) dense index and Pyserini (Lin et al., 2021) BM25 index for efficient search. For the dense index, we first use DRAGON+ to encode each passage in the corpus into a single vector, and then use Faiss (index string "IVF65536,PQ256") to cluster the vectors into 65536 centroids and quantize them into 256 codes of 8 bits each. For the sparse index, we use the default hyper-parameters and the "optimize" option in Pyserini to reduce the index size. For the approximate nearest neighbor retrieval, we use nprobe$= 4096$. During passage search, we retrieve 4000 passages from each index and keep the similarity score for fusion. The fusion coefficient η is determined by the relative confidence of dense and sparse retrievers similar to Equation (4). We set the dense coefficient $\eta_{\text{dense}} = 1 - \text{top-100}(s_{\text{dense}}(q, d))/\max s_{\text{dense}}(q, d)$ and same for the sparse coefficient η_{sparse}. The final interpolation coef is $\eta = 0.7 * (1 - \eta_{\text{sparse}}) + 0.3 * \eta_{\text{dense}}$. The fusion score for each document $s(q, d) = \eta * s_{\text{dense}} + (1 - \eta) * s_{\text{sparse}}$. If a document is missing in either dense or sparse retrieval results, we set its score to the minimum similarity of the dense/sparse retrieval results. The first-stage search is done on RAM and CPUs with 32 threads. The final Wikipedia dense index size is about 8.96GB, and the sparse index size is about 3.48GB on disk.

For the second-stage token search, we use the LLM to encode the sequence and use the input to the final layer's feed-forward network after layer normalization as the key and query vectors following Khandelwal et al. (2020). We retrieve the top-1024 tokens using the squared ℓ_2 distance and compute the non-parametric probability according to Equation (2).

Rest of NEST   For relative retrieval confidence, we set $\alpha = 0.3, \tau = 0.1$ for all Wikipedia-based tasks and $\alpha = 0.2, \tau = 0.1$ for Pile of Law for all model sizes in Equation (4). For dynamic span selection, we set the n-gram length to be 64 and $\delta = 0.5$ for all model sizes and all tasks in Equation (6). For relaxed speculative decoding, we set $\gamma = 5e - 4$ for Pile of Law tasks for all model sizes in Equation (7). For Wikipedia-based tasks, we set $\gamma = 5e - 4$ for the 7B model, $\gamma = 5e - 3$ for the 13B model, and $\gamma = 5e - 2$ for the 70B model. For RA-NEST, all models use the same $\gamma = 5e - 1$ for all tasks except Pile of Law which still uses $\gamma = 5e - 4$ We observe that as the model gets stronger, using larger γ which leads to more rejection, is more beneficial to generation quality. The complete NEST procedure is provided in Algorithm 1.

B   Evaluation Details and Hyper-parameter Tuning

Datasets.   We sample subsets of WikiText-103, NQ, and Biography as dev sets for hyper-parameter tuning. We use the validation sets of WikiText-103 (CC BY-SA 3.0), NQ (Apache License 2.0), TriviaQA (Apache License 2.0), MedMCQA (MIT License), HotpotQA (Apache License 2.0), MMLU (MIT License), and Biography (MIT License) for validation. TruthfulQA (Apache License 2.0) only has the test set. We finally test all datasets shown in Table 1. For HotpotQA and MedMCQA, we do not have access to the test set and therefore the validation results are reported. For Biography, we use the labeled data that have human annotation as the validation and dev set, and the unlabeled data as the test set. In the original FACTSCORE paper, the authors use InstructGPT (Ouyang et al., 2022) to perform fact decomposition before verification. We hereby train our own decomposition model by further fine-tuning Llama-2 7B using publicly available datasets (Chen et al., 2022; Liu et al., 2023; Malaviya et al., 2024). For fact verification, we use the option `retrieval+llama+npm` to evaluate the decomposed atomic facts.

Inference and prompts.   For language modelling and text completion, we use a context length of 128 tokens and a max generation length of 256 tokens. For the other tasks, we use 128 tokens as max generation length for question answering and 512 for fact verification. We remove all the in-context demonstrations from the prompt to test the zero-shot effectiveness of our model. We use greedy decoding in our experiments as the randomness in sampling can undermine factuality.

Regarding the prompts we use for evaluation, for MMLU, we compare the perplexity of each option concatenated with the question and select the one with the minimum perplexity.

For text completion, we use the following prompt "`[INST]` Write an article.\n Article: `[/INST]` {prefix}" where the `[INST]` is a format tag for Llama-2-Chat.

Algorithm 1 NEST w/ Greedy Decoding

Inputs: Language model LM, hidden state encoder f, first-stage retriever R, corpus $\mathcal{C}$, input x.
▷ First-stage retrieval: Retrieve documents $d_1, d_2 \ldots, d_b$ from corpus $\mathcal{C}$
$d_1, d_2 \ldots, d_b \leftarrow R(x, \mathcal{C})$
▷ Second-stage retrieval: Construct token-level key-value memory
$(\mathcal{K}', \mathcal{V}') \leftarrow \varnothing$
for $i = 1$ **to** b **do**
    $w_1^{d_i}, w_3^{d_i}, ..., w_m^{d_i} \leftarrow d_i$
    $h_1^{d_i}, h_3^{d_i}, ..., h_m^{d_i} \leftarrow f(d_i)$
    for $i = 1$ **to** $m - 1$ **do**
        $(\mathcal{K}', \mathcal{V}').\text{add}(h_j^{d_i}, w_{j+1}^{d_i})$
    end for
end for
▷ Generation
$y_{<t} \leftarrow x$
for $t = 1$ **to** T **do**
    ▷ Compute query embedding
    $q_t \leftarrow f(y_{<t})[-1]$
    ▷ Token embeddings search, return top-r scores and values
    $\pi \leftarrow (\mathcal{K}', \mathcal{V}').\text{search}(q_t, r)$
    $(s_1, v_1), (s_2, v_2), ..., (s_r, v_r) \leftarrow \pi$
    ▷ Compute non-parametric distribution
    $p_{k\text{-NN}}(w | y_{<t}) \leftarrow 0, \forall w \in \text{vocabulary}$
    for $i = 1$ **to** r **do**
        $p_{k\text{-NN}}(w = v_i | y_{<t}) \leftarrow p_{k\text{-NN}}(w = v_i | y_{<t}) + \exp(\mu \cdot s_i) / \sum_{i=j}^{r} \exp(\mu \cdot s_j)$
    end for
    ▷ Confidence-based Interpolation
    $\lambda_t \leftarrow \text{sigmoid}((\frac{\min_i s_i}{\max_i s_i} - \alpha)/\tau)$
    $p_{\mathcal{M}}(w | y_{<t}) \leftarrow \lambda_t \cdot p_{\text{LM}}(w | y_{<t}) + (1 - \lambda_t) \cdot p_{k\text{-NN}}(w | y_{<t})$
    ▷ Dynamic span selection
    $w_t \leftarrow \underset{w}{\text{argmax}}\, p_{\mathcal{M}}(w | y_{<t})$
    $v_t \leftarrow \underset{v_i = w_t}{\text{argmax}}\, p_{k\text{-NN}}(w = v_i | y_{<t})$
    $v_{t:t+n} \leftarrow \mathcal{C}.\text{get-ngram}(v_t, n)$
    $y_t \leftarrow \begin{cases} w_t, & \text{if } \lambda_t > \delta; \\ v_{t:t+n}, & \text{otherwise.} \end{cases}$
    $n \leftarrow |y_t|$
    ▷ Relaxed Speculative Decoding
    for $i = 1$ **to** n **do**
        $p_{\text{accept}}(w_t^{(i)}) \leftarrow \frac{p_{\mathcal{M}}(w = w_t^{(i)} | x, y_{<t}, w_t^{(1)}, w_t^{(2)}, ..., w_t^{(i-1)})}{\gamma \cdot \max\limits_{w} p_{\mathcal{M}}(w | x, y_{<t}, w_t^{(1)}, w_t^{(2)}, ..., w_t^{(i-1)})}$
        Break if $p_{\text{accept}}(w_t^{(i)}) \leq 0.5$
    end for
    if $i < n$ and $n > 1$ **then**
        $w_t^{(i)} \leftarrow \underset{w}{\text{argmax}}\, p_{\mathcal{M}}(w | y_{<t}, w_t^{(1)}, w_t^{(2)}, ..., w_t^{(i-1)})$
    end if
    $y_{<t} \leftarrow \text{concatenate}(y_{<t}, w_t^{(1)}, w_t^{(2)}, ..., w_t^{(i)})$
end for
Return $y_{<t}$

(a) Passages and tokens. (b) Interpolation coefficient. (c) Threshold for span selection.
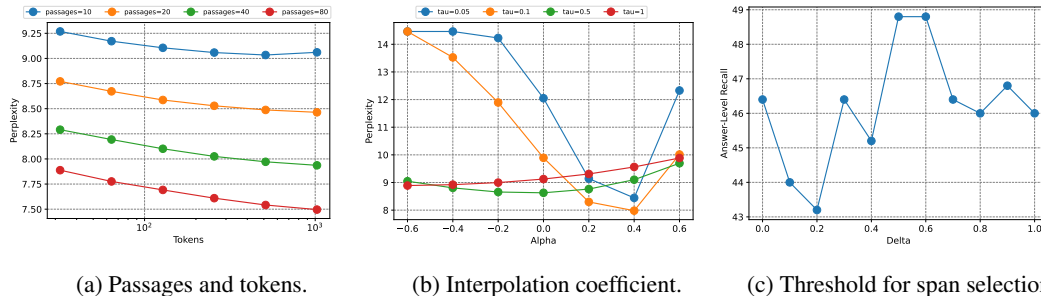
Figure 3: Sensitivity analysis on WikiText-103 and NQ dev set for the NEST-7B model with the above hyper-parameters in the sub-figures.

For question-answering and fact-verification tasks, we use the following template: "`[INST]` Question: {question} Answer: `[/INST]`" where we format the input question in the bracket.

For the RA models, we use the prompt "`[INST]` Write an article with the background context as reference. Background: {retrieved passages}\n Article: `[/INST]` {prefix}" for text completion. For retrieval-augmented question-answering and fact-verification tasks, we use "`[INST]` Answer the question with the background context as reference. Background: {retrieved passages}\n Question: {question} Answer: `[/INST]`".

Hyper-parameters and baselines.   For the base LM, we do not tune the hyper-parameters released with the original Llama-2-chat models. For the in-context retrieval augmented baseline, we select the top-3 retrieved passages. For kNN-LM, we follow Equation (3) and use an interpolation coefficient of 0.7 for Wikipedia-based tasks and 0.9 for Pile of Law. For NEST and RA-NEST, we first tune the hyper-parameters in Equation (4) on language modelling tasks using perplexity. We then fix those hyper-parameters and then tune the rest of the parameters in Equation (6) and (7) on generation tasks. All hyper-parameters in the above methods are tuned on the dev sets of WikiText-103, NQ, and Biography.

C   Analysis

The following analyses are performed on the validation set of WikiText-103, NQ, and Biography data with the Llama-7B-chat model.

C.1   Sensitivity

Number of retrieved passages and tokens   Khandelwal et al. (2020) show that increasing the size of the database and the number of tokens can improve the perplexity with proper hyper-parameter setting. We also verify whether our two-stage k-NN search and RRC approach follow the same trend. Figure 3a shows the validation perplexity on WikiText-103. For a fixed number of passages, the perplexity decreases as the number of tokens increases; for a fixed number of tokens, the perplexity decreases about $0.5 \sim 1.0$ as the number of passages doubles. However, as NEST needs to encode the retrieved passages on the fly, the latency also increases linearly w.r.t. the number of passages. Therefore, we set the passage number to be 40 and the token number to be 1024 in the main experiments.

Interpolation coefficient   Figure 3b shows the sensitivity of the hyper-parameters α (offset) and τ (temperature) in Equation (4) on WikiText-103. When τ is big, λ_t is close to a uniform distribution and therefore the offset α does not have a big impact on the perplexity. When τ is small, the impact of α is enlarged and the sweet spot is achieved around $\tau = 0.1$ and $\alpha = 0.4$.

Threshold for dynamic span selection   Figure 3c shows how threshold δ in Equation (6) affects the generation on NQ. A bigger δ means selecting the span instead of a token more often. We can see that the answer-level recall on NQ first increases and then decreases as we increase the value of δ, where the sweet spot is around $\delta = 0.5$.

Models (7B)	Wiki./ROUGE-1	NQ/ALR	Bio./FS
kNN-LM (two-stage)	20.1	40.8	34.8
+ Relative Retrieval Confidence	24.7	44.4	41.6
+ Dynamic Span selection	24.5	44.6	41.6
+ Relaxed speculative decoding	26.8	45.4	46.8

Table 3: Ablation study on the validation set of WikiText-103, NQ, and Biography. ROUGE-1 is reported for WikiText-103, ALR is reported for NQ, and FACTSCORE is reported for Biography.

C.2 Ablation Study

Table 3 shows a progressive ablation of NEST on WikiText-103, NQ, Biography. As mentioned in Section 3.1, it is extremely expensive to encode billion-token corpus with billion-parameter models. Therefore, we directly start with the two-stage implementation of kNN-LM and gradually add the methods applied in NEST. As we can see, adding the RRC component gives the first effectiveness boost. The second dynamic span selection method does not seem to increase the effectiveness, yet it is crucial to give consistent attribution for consecutive spans and tokens. The last relaxed speculative decoding method further improves the final generation quality.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that NEST can provide better generation (Section 4.4), attribution (Section 4.6), and latency (Section 4.5) compared to the base LM and kNN-LM models.

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6.

   Guidelines:
   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the complete algorithm of NEST in Algorithm 1 and implementation in Appendix A for reproduction.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code base is released at https://github.com/facebookresearch/NEST/tree/main. All the datasets we used are publicly available as discussed in Section 4.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide details about datasets, inference, evaluation, hyper-parameters, and baseline in Section 4 and Appendix B. Our method is a training-free method and therefore does not involve training-related hyper-parameters such as training steps and learning rate.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: This paper does not include significant tests considering the performance gap between the proposed approach and the baselines.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Computational resources and hardware information for inference are reported in Section 4.5 and Appendix A. Our method does not involve training.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our paper conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section 8.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our method is a general framework that can be applied to different language models and knowledge sources, and therefore does not require safeguards.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Except the internal code base, all the existing assets including code and data used in this paper are properly cited in Section 4.2 and Appendix B.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not introduce new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.