GistScore: Learning Better Representations for In-Context Example Selection with Gist Bottlenecks

Shivanshu Gupta[1*] **Clemens Rosenbaum**[2*] **Ethan R. Elenberg**[2]
[1]University of California Irvine [2]ASAPP
shivag5@uci.edu, cgbr@cs.umass.edu, eelenberg@asapp.com

Abstract

Large language models (LLMs) have the ability to perform in-context learning (ICL) of new tasks by conditioning on prompts comprising a few task examples. This work studies the problem of selecting the best examples given a candidate pool to improve ICL performance on given a test input. Existing approaches either require training with feedback from a much larger LLM or are computationally expensive. We propose a novel metric, *GistScore*, based on *Example Gisting*, a novel approach for training example retrievers for ICL using an attention bottleneck via Gisting, a recent technique for compressing task instructions. To trade-off performance with ease of use, we experiment with both fine-tuning gist models on each dataset and multi-task training a single model on a large collection of datasets. On 21 diverse datasets spanning 9 tasks, we show that our fine-tuned models get state-of-the-art ICL performance with 20% absolute average gain over off-the-shelf retrievers and 7% over the best prior methods. Our multi-task model generalizes well out-of-the-box to new task categories, datasets, and prompt templates with retrieval speeds that are consistently thousands of times faster than the best prior training-free method.

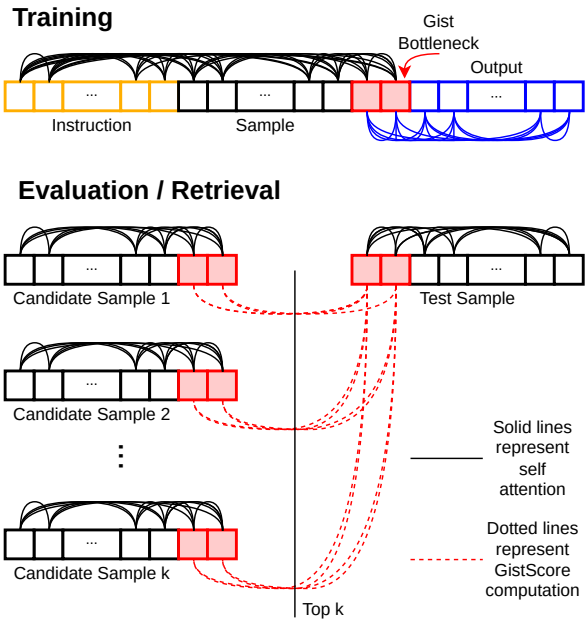

Figure 1: Top – Attention mask for the proposed method of Example Gisting. Gist bottleneck tokens (red) may attend to the input sample (white) and the task instruction (yellow, optional). However, the output (blue) may only attend to the gist tokens, creating a bottleneck that encourages concise, task-dependent representations of salient aspects of the test input. Bottom – Retrieval of the candidate examples with highest GistScore with the test input.

1 Introduction

In-context Learning (ICL) (Brown et al., 2020) is a few-shot inference paradigm that leverages increasingly powerful large language models (LLMs) for new tasks by conditioning them on a prompt comprising a few task demonstrations. In contrast to traditional supervised fine-tuning, the training-free approach allows a single model to instantly switch between an arbitrary number of tasks with improved generaliztion (Anil et al., 2022; Qiu et al., 2022; Drozdov et al., 2023; Wei et al., 2023) and reasoning skills (Wei et al., 2023). Unfortunately, its performance is highly sensitive to the choice of examples placed in the prompt (Zhao et al., 2021; Liu et al., 2022b; Lu et al., 2022; Rubin et al., 2022; Schick and Schütze, 2021).

Despite extensive prior work on better example selection methods (Rubin et al., 2022; Ye et al., 2023; Mualem et al., 2023; Gupta et al., 2023), the predominant approach in practice remains to use off-the-shelf retrievers like BM25 or cosine similarity between general purpose encoder representations (Reimers and Gurevych, 2019). This is because most prior work requires training on the target task and/or feedback from a much larger Inference LLM (Rubin et al., 2022; Ye et al., 2023; Hu et al., 2022), eliminating a key advan-

[*] Work done at ASAPP

tage of ICL methods. More recently, Gupta et al. (2023) proposed training-free approaches based on BERTScore (Zhang et al., 2020). However, BERTScore-Recall (BSR) is computationally expensive for tasks with long instances or large candidate pools. Moreover, contextualized token embeddings obtained from general-purpose off-the-shelf encoders may not be suitable for many tasks, leading to suboptimal selection.

This work proposes *Example Gisting*, a novel approach to training encoders for ICL example selection without feedback from a larger LLM. Based on Gisting, a recent technique by Mu et al. (2023) for compressing prompts, Example Gisting induces a bottleneck between example inputs and outputs via a structured attention mask. Supervised training with this bottleneck comprising a few *gist tokens* forces the model to store salient information about the inputs into those tokens' activations. Subsequently, the learned model efficiently maps both candidate examples and new test inputs into sequences of gist token embeddings. These embeddings are the basis for *GistScore*, a novel metric to rank candidates for ICL. As the gists comprise only a few tokens, GistScore-based selection can be significantly faster than BERTScore. Finally, we explore 2 variations of Example Gisting models: fine-tuning on a particular dataset to get the best performance on that dataset and, inspired by instruction tuning, multi-task training on a large collection of datasets and tasks for a training-free ICL example selection method for new tasks, datasets, and prompt templates.

Evaluating our proposed approach on 21 diverse datasets spanning 9 task categories and three LLMs of varying sizes, we show that example selection using our gisting models dramatically improves ICL performance. Our task fine-tuned gist models consistently outperform all prior selection methods, including those that leverage training. Further, our multi-task pre-trained gist model also outperforms all prior methods on held-in datasets while also performing better than prior training-free methods on held-out datasets. Additionally, as gists comprise only a few tokens, selection using our approach is over 10,000 times faster than BERTScore. Overall, our multi-task pretrained gist model presents the best tradeoff of performance, ease of use, and selection speed and can potentially replace the standard approach of using off-the-shelf retrievers. Finally, analysis of the gist tokens embeddings reveals that they do indeed capture abstract, task-specific salient aspects.

2 Related Work

2.1 In-Context Learning

In-context learning is an appealing approach for leveraging LLMs due to its training-free nature coupled with improved generalization (Anil et al., 2022; Qiu et al., 2022; Drozdov et al., 2023) and reasoning skills (Wei et al., 2023). However, its performance is highly sensitive to the choice of in-context examples. Prior work has tried various approaches to select examples that make it more likely for the LLM to produce the correct answer: (1) selecting diverse examples to reduce redundancy among them (Su et al., 2022; Levy et al., 2022; Agrawal et al., 2022; Ye et al., 2022), (2) selecting examples that minimize the entropy of the LLM's output distribution for the test input (Lu et al., 2022; Wu et al., 2023), (3) Bayesian inference (Wang et al., 2023b), and (4) selecting examples as a set (Gupta et al., 2023; Ye et al., 2023; Mualem et al., 2023).

Perhaps the most relevant to our work are Rubin et al. (2022) and Wang et al. (2023a) which propose different ways to train retrievers based on LLM feedback for ranking and selecting the most relevant examples. Another related approach is using different metrics for scoring; Gupta et al. (2023) suggest using BERTScore (Zhang et al., 2020) as a training-free metric to rank and select candidates that are informative with respect to the test input. However, while Rubin et al. (2022) and Wang et al. (2023a)'s approaches are limited in their ease of use due to need for task and LLM-specific training, Gupta et al. (2023)'s approach is limited by computational cost.

2.2 Instruction Gisting and Sparse Attention

Since the transformer architecture's self-attention mechanism has quadratic runtime complexity as a function of the input length, methods for sparsifying the attention connections to reduce the runtime complexity are nearly as old as the transformer architecture itself. Early approaches include different forms of dense local attention with additional mechanisms to model global dependencies. Dai et al. (2019) use block-wise dense local attention combined with recursive attention to the previous attention block. Child et al. (2019) and Beltagy et al. (2020) use different forms of sliding (and

strided) attention; they model long dependencies with either overlapping windows or specific tokens with overlapping attention. More recently, architectures like the ones proposed by Guo et al. (2022) and Xiao et al. (2023) are designed around the fact that the global attention elements modeling long sequence dependencies can be modeled with global tokens that are designed to act like a shared global memory, and not like passthrough tokens. This intuition is shared by Gisting (Mu et al., 2023): since the shared memory consists of much fewer tokens than the whole sequence length, it appears that token representations can act as *information bottlenecks* that can contain a lot of information pertinent to the task at hand.

3 Preliminaries

In-context Learning (ICL) Many LLMs have the ability solve test inputs of a task by prompting them with a few examples of that task. Formally, given a set of (input, output) pairs $\{(x_i, y_i)\}_{i=1}^{k}$, prompt template $\mathcal{T}$, and the test input $\mathbf{x}_{\text{test}}$, ICL using an *Inference LLM* involves prompting it to conditionally generate the following test output:

$$\mathbf{y}_{\text{test}} \sim \mathcal{P}_{LM}\left(\cdot \mid \mathcal{T}\left(\mathbf{x}_1, \mathbf{y}_1, \ldots, \mathbf{x}_k, \mathbf{y}_k, \mathbf{x}_{\text{test}}\right)\right). \tag{1}$$

Example Selection In this work, we focus on the setting where the k in-context examples need to be selected from a large pool of $N \gg k$ labeled candidates. This is a common setting due to limited context windows of LLMs. Moreover, even if the context window of the inference LLM could fit the entire pool, previous work has shown that LLMs are sensitive to both order (Liu et al., 2022b) and position of in-context examples (Liu et al., 2023). Thus, we seek to improve both computational efficiency and accuracy by choosing a few relevant candidates. More formally, the goal is to select a subset $\mathcal{S} \subset \{(x_i, y_i)\}_{i=1}^{N}$ of size k that maximizes the probability of generating the desired $\mathbf{y}_{\text{test}}$ when the Inference LLM is conditioned on $\mathbf{x}_{\text{test}}$ and $\mathcal{S}$.

Since the test output is unavailable, the standard approach for this problem is to score and retrieve the top-k examples from the candidate pool using either the BM25 algorithm or cosine similarity between the test input and the candidates using an off-the-shelf encoder. However, such general-purpose retrievers are trained for language modeling rather than selecting examples for in-context learning. Moreover, standard approaches for training retrievers (Karpukhin et al., 2020) are not applicable as the gold retrieval is unknown. Recent works attempt to mitigate this problem by training retrievers using feedback from a much larger Inference LLM (Rubin et al., 2022; Wang et al., 2023a). However, the need to train for every task goes against the training-free nature of in-context learning, and training using an Inference LLM can limit the effectiveness of these approaches to larger Inference LLMs as shown in Gupta et al. (2023).

Different from these, Gupta et al. (2023) showed that simply using BERTScore for scoring examples yields a training-free method that selects in-context examples that are informative about how to solve the test input, significantly improving ICL performance. However, as BERTScore matches every pair of token embeddings in the candidate and the test input, it is computationally expensive and does not scale well as N increases. Moreover, general-purpose encoders used with BERTScore may not capture informativeness for every task.

Instruction Gisting Recently Mu et al. (2023) proposed a technique called Gisting[1] to compress instruction-following prompts into shorter *gist tokens* for efficient LLM inference. To perform this mapping, they use a gisting model, GM, that is simultaneously trained to compress prompts comprising task instructions into gist tokens and to follow instructions encoded in those gist tokens. This is achieved by masking attention such that any attention to/from the task instruction goes through the gist tokens.

Specifically, given an initial model LM and an instruction tuning dataset $\mathcal{D}_{\text{instr}} = \{(t_i, x_i, y_i)\}$ of instruction, (optional) input, and target tuples, the model is trained to predict y from the sequence $[t, G, x]$, where G is the sequence of special gist tokens added to the model vocabulary. The model must store the information in t into the activations above G. Moreover, since the tokens following G do not directly attend to t, the distillation training loss may be written as

$$\mathcal{L}_G\left(p_G, \mathcal{D}_{\text{instr}}\right) = \mathop{\mathbb{E}}_{t,x,y\sim\mathcal{D}_{\text{instr}}}\left[\mathrm{KL}\left(p_{\text{LM}}(y \mid t, x) \parallel p_{GM}(y \mid G(t), x)\right)\right]. \tag{2}$$

[1]We refer to this method as Instruction Gisting to distinguish between prior work and our proposed methods.

During inference, the model handles new instructions by feeding it the sequence $[t, G]$, precomputing the activations above G, and then prompting GM with those activations instead of t.

4 Method

Next, we describe the motivation and details of our proposed approach for ICL example selection.

4.1 Motivation

The intutition behind our proposed approach relies on two core insights. The first was arguably made most explicit by Xiao et al. (2023). They note that individual tokens can act as *attention sinks*, *i.e.*, an information bottleneck that acts like a global memory capturing essential information of (possibly) infinitely long input sequences. This suggests that attention based architectures are able to store essential information in otherwise meaningless token activations.

The second insight is possibly stated best by Gupta et al. (2023), which notes that example selection for in-context learning requires a similarity metric that is able to capture the potentially many task-specific facets under which two samples can be similar. These *salient aspects* can capture reasoning patterns, rules, or similar properties of samples that make one a good example for solving another.

Combined, these two insights raise an interesting question: can we train attention-based models that can capture these salient aspects in memory-like bottlenecks? Going beyond global information, we hypothesize that training a model to perform a task with a bottleneck between the inputs and output would enable these bottlenecks to store the task-relevant aspects of the inputs, which are most helpful for distinguishing between candidates during example selection.

4.2 Example Gisting

We now describe Example Gisting, our approach to train example encoders for ICL example selection. Consider an initial LM and a labeled dataset for target task t: $\mathcal{D}_t = \{(x_i, y_i)\}$. Analogous to Instruction Gisting, we train a model GM to predict y_i given the inputs $[x_i, G]$, where G is the attention bottleneck comprising l gist tokens. As in Eq. (2), this is akin to minimizing the following distillation objective:

$$\mathcal{L}_G\left(p_G, \mathcal{D}_t\right) = \mathop{\mathbb{E}}_{x,y\sim\mathcal{D}_t}\left[\mathrm{KL}\left(p_{\mathrm{LM}}(y \mid x) \| p_{GM}(y \mid G(x))\right)\right]. \quad (3)$$

As motivated in § 4.1, Example Gisting training as described above forces the model to encode task-specific salient information of the inputs in the activations of the gist tokens. Note that, unlike Instruction Gisting, we only use the gisting model's activations to select examples. The subsequent Inference LLM receives the full text of these examples, which means that Example Gisting is completely agnostic to which model is used for ICL inference on the target task.

4.3 Example Selection

Having trained an Example Gisting model GM, we use it to map the candidates and the test input to sequences of one or more *gist embeddings* that can be used for scoring the candidates. Specifically, given the gist tokens $G(x_{\text{test}})$ of the test input, and $G(z)$ for each candidate z, we use the final layer output of the gisting model as gist embeddings, *i.e.* $\mathbf{z} = \mathbf{z}_1, \ldots \mathbf{z}_l = GM(z)[-1]$ and $\mathbf{x} = \mathbf{x}_1, \ldots \mathbf{x}_l = GM(x_{\text{test}})[-1]$. Then we use the following metric, which we call GistScore, to measure the relevance of each candidate with respect to the test input:

$$\mathrm{GS}(\mathbf{x}, \mathbf{z}) = \frac{1}{l}\sum_{i=1}^{l} \max_{j=1,\ldots l} \frac{\mathbf{x}_i^T \mathbf{z}_j}{\|\mathbf{x}_i\|\|\mathbf{z}_j\|} \quad (4)$$

Finally, the candidate examples are then ranked according to their GistScore, and the top-k examples are selected. Note that for $l = 1$, GistScore reduces to cosine similarity. Using $l > 1$ may be useful when a single embedding is insufficient to capture all salient information, for example, when the examples long or complex.

4.4 Multi-Task Training

The approach described in § 4.2 requires having enough training data for task t to fine-tune the gisting model. Even with an abundance of data, the additional training step limits the method's ease of use. In this section we describe a multi-task pre-training method which enables the gisting model to be used out-of-the-box on new tasks without any additional training. This preserves a key advantage of ICL: the entire pipeline may be used with new

Task Category	Dataset	Train Size	Test Size
Commonsense Reasoning	COPA (Roemmele et al., 2011)	400	100
	CMSQA (Talmor et al., 2019)	9741	1221
Chain-of-Thought Reasoning	GSM8K (Wei et al., 2023)	7473	1319
Grammar	COLA (Warstadt et al., 2019)	8551	1043
Natural Language Inference	QNLI (Wang et al., 2018)	104738	5463
	MNLI (Williams et al., 2018)	392699	9815
	RTE (Bentivogli et al., 2009)	2490	277
	WANLI (Liu et al., 2022a)	102885	5000
	XNLI (Conneau et al., 2018)	392684	2490
	MEDNLI (Herlihy and Rudinger, 2021)	11232	1395
Paraphrase Detection	MRPC (Dolan and Brockett, 2005)	3668	408
	QQP (Wang et al., 2018)	363846	40430
	PAWS (Zhang et al., 2019)	49401	8000
Reading Comprehension	DROP (Dua et al., 2019)	66610	6100
	BOOLQ (Clark et al., 2019)	9395	3252
Semantic Parsing	SMCALFLOW (Andreas et al., 2020)	25396	662
	MTOP (Li et al., 2021)	15667	2235
	COGS (Kim and Linzen, 2020)	24155	3000
Sentiment Analysis	SST2 (Socher et al., 2013)	67349	872
	SST5 (Socher et al., 2013)	8544	1101
Summarization	AGNEWS (Zhang et al., 2015)	120000	7600

Table 1: Datasets used in this work. The train and test set sizes are for the IID splits. For SMCALFLOW and COGS, we additionally evaluate on a compositional generalization test set described in App. A.

tasks, task categories, and prompt templates without any training. The main idea is to learn gist tokens that encode both the task instruction and the input. Formally, given an initial LM and a collection of datasets $\mathcal{D}_{\text{multi}} = \bigcup_{t \in T}\{(t, x, y) : (x, y) \in \mathcal{D}_t\}$ spanning tasks T, we train the model to predict y given the inputs $[t, x, G]$ where G is the attention bottleneck as before. This is equivalent to minimizing the following distillation objective:

$$\mathcal{L}_G\left(p_G, \mathcal{D}_{\text{multi}}\right) = \mathop{\mathbb{E}}_{t,x,y \sim \mathcal{D}_{\text{multi}}} \left[\text{KL}\left(p_{\text{LM}}(y \mid t, x) \| p_{GM}(y \mid G(t, x))\right)\right]. \quad (5)$$

5 Experimental Setup

5.1 Datasets

We experiment with 21 datasets spanning 9 task categories. Table 1 summarizes the datasets used in this work. For all datasets, we use the standard IID splits, except for SMCALFLOW and COGS, for which we additionally evaluate on a compositional generalization (CG) split. We refer the reader to App. A for more details about the datasets, splits, sample instances, and the prompt templates used for each dataset.

ICL Evaluation Following prior work (Gupta et al., 2023; Rubin et al., 2022; Ye et al., 2023), for each split, we use up to 44,000 random instances from the train set as the pool to select demonstrations from and evaluate using a random subsample of 1000 instances of the validation set if available, and the test set otherwise. We report Exact-Match Accuracy for all the Semantic Parsing datasets and Accuracy for the remaining datasets.

5.2 Models

We experiment with **GPT-Neo-2.7B** (Black et al., 2021): A 2.7B-parameter LM trained on The Pile (Gao et al., 2020), an 825 GB text corpus and **LLaMA** (Touvron et al., 2023): A collection of LMs ranging from 7B to 65B parameters pretrained on web datasets such as CommonCrawl, GitHub, and arXiv. We experiment with LLaMA-7B and LLaMA-13B base models. All three LLMs have a context window length of 2048.

5.3 Methods

5.3.1 Gist LM

We use encoder-decoder models for both task fine-tuned and multi-task pretrained gist models. This means that after training, we can drop the decoder and only keep the encoder for computing and ranking example gists. We experiment with the following different variants of Gist LM-based retrievers:

LLM	Selector	Sem.	NLI	Sent.	Para.	Comm.	Summ.	CoT	RC	LA	AVG	Held-in
GPT-Neo-2.7B	RANDOM	3.3	43.9	50	55	38	76.6	1.7	23.5	60.3	35.0	45.1
	SBERT	23.4	48.3	59.9	60.0	38.0	89.3	2.0	29.9	64.9	43.4	49.8
	BM25	28.4	47.3	60.7	61.4	37.2	89.4	4.0	31.1	64.4	44.6	50.0
	BSR	40.4	67.8	68.0	77.2	38.5	89.9	2.4	30.5	69.7	55.5	59.7
	GIST[FINETUNE]	**49.6**	**78.7**	**72.0**	**88.9**	**59.0**	**92.1**	3.1	**47.5**	**80.0**	**66.5**	**70.2**
	GIST[MULTI]	31.4	72.8	70.2	86.3	55.1	91.4	**3.4**	41.9	76.4	59.3	66.8
LLaMA-7B	RANDOM	6.8	46.7	66.3	53	61.5	85.7	11	47.8	60.1	42.7	56.2
	SBERT	31.2	49.0	67.6	59.8	58.5	86.8	12.3	50.5	70.3	50.1	59.0
	BM25	36.2	48.5	69.2	59.6	56.1	88.2	12.4	52.0	67.0	51.0	58.8
	BSR	49.5	61.0	68.2	71.4	58.1	88.9	14.3	52.5	70.3	59.1	64.5
	GIST[FINETUNE]	**56.7**	**74.2**	**70.7**	**85.6**	74.3	**90.7**	12.6	**59.6**	**77.4**	**68.5**	**73.3**
	GIST[MULTI]	39.7	67.2	70.5	82.1	**75.0**	90.4	**15.6**	57.2	74.4	62.3	71.8

Table 2: Comparing average 8-shot ICL performances with training-free methods on Semantic Parsing (Sem.), NLI, Sentiment Analysis (Sent.), Paraphrase Detection (Para.), Commonsense Reasoning (Comm.), Summarization (Summ.), Chain-of-thought (CoT), Reading Comprehension (RC), and Linguistic Acceptability (LA) datasets. While GIST[FINETUNE] clearly outperforms all prior training-free methods, GIST[MULTI] also matches or outperforms BSR. Note that Semantic Parsing is a held-out task and NLI includes held-out datasets for GIST[MULTI]. Additionally, the prompt templates differ from those in its seen during multi-task training. See App. B for complete results for each dataset and LLM.

Finetuned Gisting models (GIST[FINETUNE]) In this setting, we fine-tune Flan-T5-base Chung et al. (2022) models to produce gists of varying lengths on each individual dataset using the procedure described in § 4.2. For each dataset, we use the entire train set with instances longer than 500 tokens filtered out for computational efficiency. For early stopping, we compute Rouge-L (Lin, 2004) for DROP and GSM8K and Exact-Match Accuracy for the remaining datasets on up to 1000 random instances from the validation set. All training was done with batch size 36 for up to 40000 steps with early stopping with the Adafactor optimizer (Shazeer and Stern, 2018) and a constant learning rate of 5e-5.

Multi-task Pre-trained Gist Model (GIST[MULTI]) For this setting, we train multiple Flan-T5-large models for gisting to $l = \{1, 3, 6, 15\}$ tokens on a large dataset of prompts subsampled from the FLAN 2022 collection (Longpre et al., 2023) of 15M prompts from over 473 datasets and 146 task categories. Specifically, we take zero-shot prompts at most 256 tokens long and further subsample at most 10,000 prompts for every task category. We use 95% of this sub-collection for training and 1000 random instances from the remaining 5% for early stopping with Rouge-L (Lin, 2004) as the metric. Each model was trained using the Adafactor optimizer (Shazeer and Stern, 2018) on an NVIDIA A10G GPU with a batch size of 4 and 64 gradient accumulation steps for an effective batch size of 256. The learning rate was kept constant at 5e-4.

For both the task fine-tuned and multi-task pre-trained models, we discuss results with 1-token gist models unless specified otherwise.

	Out-of-the-box	Selection
SBERT	✔	$O(n)^\diamond$
BSR	✔	$O(nL^2)$
EPR	✖†	$O(n)^\diamond$
CEIL	✖†	$O(n)$
LLM-R	✔†	$O(n)\diamond$
GIST[FINETUNE]	✖	$O(n)\diamond$
GIST[MULTI]	✔	$O(n)\diamond$

Table 3: Comparison of training requirements and ICL example selection speed. L is the length of instances. Methods marked with † require training with an inference LLM. Note that by using FAISS[2] (Johnson et al., 2019), methods marked with $\diamond$ can be further reduced to $O(\log^2 n)$.

5.3.2 Baselines

In addition to RANDOM selection of examples, we compare with the following training-free baselines:

Cosine similarity (SBERT) Following (Gupta et al., 2023), we use the SentenceBert library (Reimers and Gurevych, 2019) with the `all-mpnet-base-v2` model.

BM25 (BM25) We use the Okapi variant (Robertson et al., 1993; Jones et al., 2000) of BM25 from the `rank_bm25`[3] library with unigram terms.

[3] https://github.com/dorianbrown/rank_bm25

BERTScore-Recall Following Gupta et al. (2023), we use the `bert_score`[4] library (Zhang et al., 2020) with `deberta-large-mnli` encoder which are DeBERTa models (He et al., 2021) finetuned on the MNLI dataset (Williams et al., 2018).

Additionally, we compare with the following methods that require training for the specific task and/or feedback from an Inference LLM:

EPR (Rubin et al., 2022) uses LLM perplexity to train a dense retriever for each dataset.

CEIL (Ye et al., 2023) uses EPR and feedback from an LLM to train a Determinantal Point Process (Kulesza, 2012) for each dataset and then uses it to select examples. For both EPR and CEIL we consider GPT-Neo-2.7B and compare with 8-shot results reported in Gupta et al. (2023) if available, defaulting to the 50-shot results reported in Ye et al. (2023) otherwise.

LLM-R (Wang et al., 2023a) iteratively trains a reward model based on feedback from LLaMA-7B to evaluate the quality of candidate examples followed by distillation into a dense retriever. We compare with their 8-shot ICL results using LLaMA-7B. Due to multi-task training, LLM-R can also be applied to new tasks out-of-the-box; however, as their held-out tasks are included in FLAN 2022, our multi-task collection, we only compare with LLM-R on its held-in datasets.

Table 3 compares these baselines to GIST[FINETUNE] and GIST[MULTI] in terms of selection time complexity (assuming a constant number of gist tokens) and whether they generalize out-of-the-box to new tasks. We see that GIST[MULTI] is best with respect to both criteria.

5.4 Prompt Construction

For k-shot (we use $k = 8$ unless specified otherwise) ICL with any given dataset (§ 5.1), demonstration selection method (§ 5.3) and LLM (§ 5.2), we construct the prompt as follows: (1) select up to k demonstrations depending on the context window of the LLM; (2) order the demonstrations in increasing order of relevance so that the most relevant demonstrations appear closest to the test input; and (3) linearize the ordered demonstrations and the test input using the dataset's example template in Tables 8, 9, and 10 and concatenate to form the prompt.

6 Results

[4] https://github.com/Tiiiger/bert_score

	SMCCS		MTOP	COGS		AVG
	CG	IID		CG	IID	
SBERT	3	38.1	49.8	29.3	35.8	31.2
BM25	9.4	46.4	55.1	31.4	39	36.3
BSR	10.6	55.1	59.7	55.2	66.8	49.5
GIST[MT, 1TOK]	4.5	43.2	55.2	41.3	54.3	39.7
GIST[MT, 6TOK]	9.8	48.2	60	48.9	59	45.2

Table 4: Comparison on Semantic Parsing, a held-out task for our multi-task (MT) gisting models with LLaMA-7B LLM (see App. B for other LLMs) ICL example selection using our multi-task gisting model outperforms prior training-free methods, except the much slower BSR, particularly when using 6 gist tokens.

	WANLI	XNLI	MEDNLI	AVG
SBERT	40.3	33.3	37.8	37.1
BM25	40.3	33.1	35.9	36.4
BSR	51.4	42.5	49.8	47.9
GIST[MULTI]	49.8	52.3	56.8	53.0

Table 5: Comparison on held-out NLI datasets with LLaMA-7B LLM (see App. B for other LLMs). GIST[MULTI] outperforms all prior training-free methods including BSR.

Example Gisting is an effective approach for training ICL example retrievers Tables 2, 6 and 7 compare the performance of ICL example selection using our gist models with prior training-free and trained approaches. Additional results on every dataset for all LLMs are provided in App. B. It is evident that example selection using our fine-tuned gisting models (GIST[FINETUNE]) consistently and dramatically outperforms all prior methods on every task category with 20 points absolute improvement compared to SBERT, 10 points compared to BSR 7 points compared to CEIL, and 8 points compared to LLM-R. Analyzing individual dataset results in App. B, we find gains as high as over 40% compared to both SBERT and BSR.

Surprisingly, on held-in datasets, even our multi-task model GIST[MULTI] outperforms all prior methods, including EPR, CEIL, and LLM-R, which receive some form of task or LLM specific training and often even GIST[FINETUNE]. This is even though the example templates we use for our ICL evaluation differ from those used by FLAN.

GIST[MULTI] gets the best out-of-the-box trade-off of performance and selection speed Tables 4 and 5 compare the performance of GIST[MULTI] on a held-out task (Semantic Parsing) and held-out datasets of a held-in task (NLI), respectively. It is clear that for semantic parsing, GIST[MULTI] is only outperformed by BSR. For held-in tasks, it

Selector	SMC-CG	SMC-IID	MTOP	QNLI	MNLI	SST5	MRPC	CMSQA	AVG
SBERT	3.9	39.7	53.9	82.6	76.7	45.1	70.1	20.1	49
EPR	3.6	54.5	62.2	74.9	66.1	42.8	76	36.8	52.1
CEIL	3.8	**59.1**	**60.5**	84.2	71.7	47	80.2	37.2	55.5
GIST[FINETUNE]	**4.4**	49.7	60.1	**91.4**	**82**	**50**	**87.3**	**59.9**	**62.5**
GIST[MULTI]	0.9	27	51.3	86.8	78.1	48.4	83.1	54.3	53.7

Table 6: Comparison with EPR and CEIL on GPT-Neo-2.7B. All numbers are 8-shot except EPR and CEIL on MNLI, SST5, MRPC, and CMSQA which are with 50-shots. GIST[FINETUNE] consistently outperforms EPR and CEIL. GIST[MULTI] also outperforms it on the non-semantic parsing datasets.

Selector	MNLI	RTE	SST2	MRPC	PAWS	QQP	COPA	AGNEWS	BOOLQ	AVG
BSR	76.3	70.8	95.8	59.8	74	80.4	86	88.9	77.6	78.8
LLM-R	69.8	70.4	93.1	78.2	57	83.3	84	**93.5**	74.1	78.2
GIST[FINETUNE]	**80.8**	84.5	94.6	**82.4**	**90.7**	**83.7**	85	**90.7**	**82.8**	86.1
GIST[MULTI]	78.5	**85.6**	**95.2**	77.9	86.3	82	**90**	90.4	81.8	85.3

Table 7: 8-shot ICL performance with LLaMA-7B on held-in datasets for LLMR. Both GIST[FINETUNE] and GIST[MULTI] consistently outperform LLM-R.

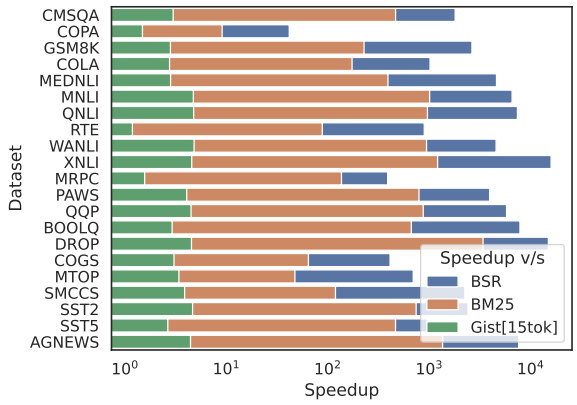

Figure 2: Relative speedup with gisting-based retrieval compared to BSR, BM25 on 1000 test inputs. We see that gisting is consistently over 1000 faster than BSR and 100 times faster than single-threaded BM25. Also, note that gisting-based retrieval scales well with the number of gist tokens. Also, see Table 14 for actual retrieval times for each method.

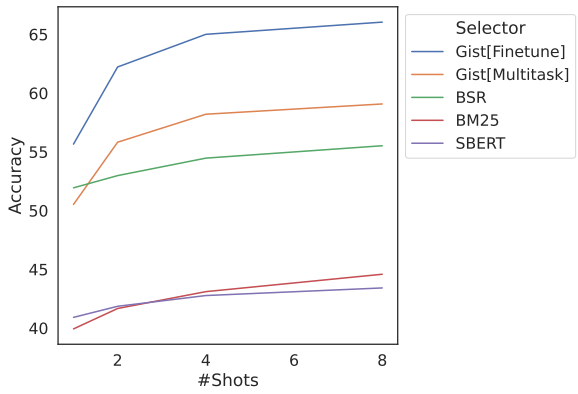

Figure 3: Average ICL performance across all datasets with GPT-Neo-2.7B for varying number of shots (in-context examples). Selection using both GIST[FINETUNE] and GIST[MULTI] consistently outperforms other methods.

matches or outperforms all training-free methods, including BSR. However, as shown in Figure 2, selection using GIST[MULTI] is thousands of times faster than BSR, which, due to its quadratic time complexity took over 20 seconds per test input for some tasks (see Table 14). Surprisingly, by taking advantage of GPU acceleration, we find selection using GIST[MULTI] to be significantly faster than even BM25.

Scaling the Number of In-Context Examples Figure 3 shows that while every example selection method improves ICL accuracy with increasing number of gist tokens, selection using our gist models (both fine-tuned and multitask) consistently outperforms other methods.

Effect of number of gist tokens As shown in Tables 11, 12, and 13 in App. B, for most tasks and datasets, we find no significant improvement from using more than one gist token, suggesting that a single token is enough to capture their salient aspects. This is similar to (Wang et al., 2023a), who did not find significant improvement in Instruction Gisting performance beyond 2 gist tokens. An exception to this is the semantic parsing datasets, which, as shown in Figure 5, benefit from using more than one gist token. This is likely because of the complex compositional nature of semantic parsing datasets, which requires more gist tokens to capture all salient aspects. Moreover, we find that while the performance of the multi-task model improves up to 6 tokens, for the fine-tuned model, the 3 tokens seem to suffice. This is likely because, having been trained for the task, fine-tuned models

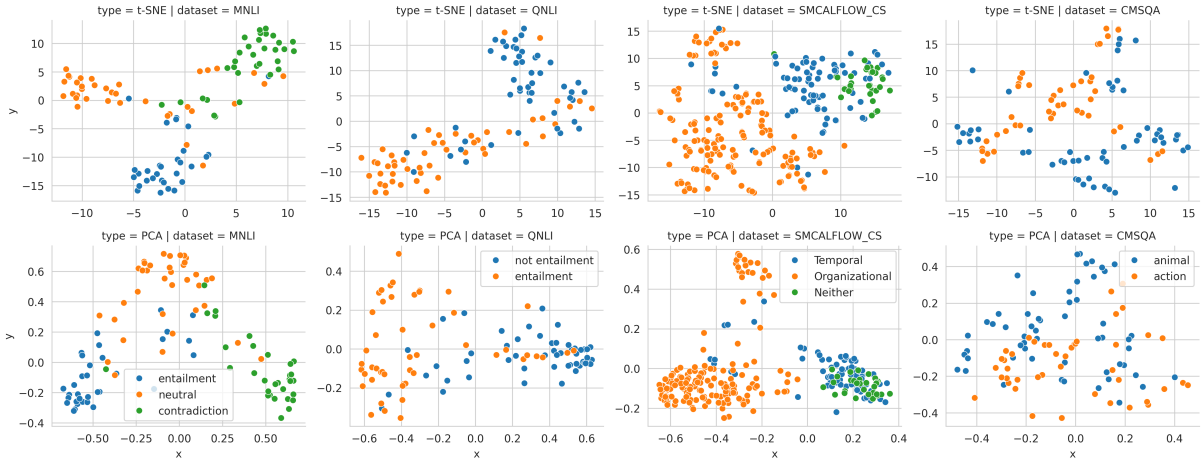

Figure 4: t-SNE and PCA Visualizations of gist activations for class labels (MNLI, QNLI) and more abstract salient aspects (SMCALFLOW, CMSQA). Gist embeddings encode task-specific salient information that helps in retrieving better in-context examples.

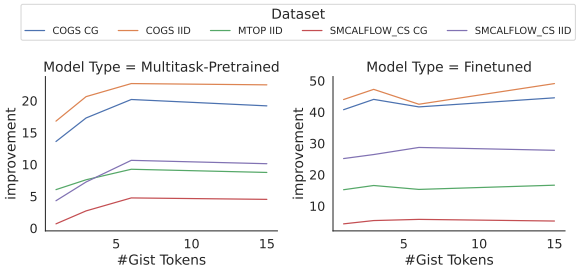

Figure 5: GPT-Neo-2.7B ICL performance on semantic parsing datasets using multitask pretrained and fine-tuned gisting LMs with a varying number of gist tokens.

can better utilize the gist tokens than the multi-task model for which semantic parsing is a held-out task.

7 Analysis

Next, we qualitatively analyze the gist token embeddings obtained from the GIST[MT, 1TOK] Flan-T5-large model across 4 diverse datasets. Additional qualitative results can be found in App. B.1

To more closely analyze the information encoded encoded by gist token embeddings, we present two dimensional visualizations of gist tokens across for various datasets in Figure 4. For classification tasks (MNLI and QNLI) we see a clear separation between different class labels suggesting the gists contain information of the correct label.

Further, for CMSQA which has different choices for every question, we see that the gist encoding contain information regarding the relevant concepts in the question, *i.e.* whether the question pertains to an animal ("cat", "bald eagle", "small dog", "snake", "fox", "weasel", "lizard", "horse", "shark", "monkey") or an action ("chatting with friends", "driving car", "competing", "doing housework", "killing people", "getting drunk").

Finally, for SMCALFLOW, we see that the gist token embeddings contain information about whether the input pertains to organizational hierarchy (*e.g.* "Who is Bill's manager?"), contains temporal information (e.g., "Book me a dentist appointment before 3pm today"), or neither (*e.g.* "I need a meeting with Steve"). These observations show that gist tokens encode not just class labels but also more abstract task-specific salient aspects (Gupta et al., 2023), further demonstrating the effectiveness of our gist-bottleneck approach.

8 Conclusion

This work presents Example Gisting, a novel approach for training retrievers for in-context learning through supervised fine-tuning of encoder-decoder models with a bottleneck that forces encoding the salient information in inputs into a few tokens. We additionally propose GistScore, a novel metric to compare the gist encodings of candidates with the test input. Evaluation on a wide range of tasks and LLMs validates the effectiveness of our approach by demonstrating superior performance of our fine-tuned encoders. Moreover we hope the out-of-the-box generalization of our multi-task pretrained models will help establish them as a new standard approach for ICL example selection. Future work could study the efficacy of gisting in other settings that require retrieval, such as retrieval augmented generation.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022. In-context examples selection for machine translation.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Cem Anil, Yuhuai Wu, Anders Johan Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Venkatesh Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The Long-Document transformer.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large scale autoregressive language modeling with meshtensorflow.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2023. Compositional semantic parsing with large language models. In *The Eleventh International Conference on Learning Representations*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 2023. Coverage-based example selection for in-context learning.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Christine Herlihy and Rachel Rudinger. 2021. MedNLI is not immune: Natural language inference artifacts in the clinical domain. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1020–1027, Online. Association for Computational Linguistics.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2627–2643, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Karen Spärck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Inf. Process. Manag.*, 36:809–840.

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

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Alex Kulesza. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286.

Itay Levy, Ben Bogin, and Jonathan Berant. 2022. Diverse demonstrations improve in-context compositional generalization.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

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

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Joram Meron. 2022. Simplifying semantic annotations of SMCalFlow. In *Proceedings of the 18th Joint ACL - ISO Workshop on Interoperable Semantic Annotation within LREC2022*, pages 81–85, Marseille, France. European Language Resources Association.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens.

Loay Mualem, Ethan R. Elenberg, Moran Feldman, and Amin Karbasi. 2023. Submodular minimax optimization: Finding effective sets.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022. Evaluating the impact of model scale for compositional generalization in semantic parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9157–9179, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1993. Okapi at trec. In *Text Retrieval Conference*, 500207, pages 109–123. National Institute of Standards and Technology.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4603–4611. PMLR.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2022. Selective annotation makes language models better few-shot learners.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Learning to retrieve in-context examples for large language models.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023b. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2022. Complementary explanations for effective in-context learning.

Pengcheng Yin, Hao Fang, Graham Neubig, Adam Pauls, Emmanouil Antonios Platanios, Yu Su, Sam Thomson, and Jacob Andreas. 2021. Compositional generalization for neural semantic parsing via span-level supervised attention. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2810–2823, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Datasets

We experiment with 21 datasets spanning 9 task categories. See Table 1 for a summary of the datasets used in this work. For all datasets other than XNLI, COGS, and SMCALFLOW, we use the standard IID splits. For XNLI which is a multilingual NLI dataset, we use the French split. For COGS, in addition to the standard IID, we also evaluate on the standard compositional generalization evaluation set. For SMCALFLOW we evaluate on the IID and compositional generalization splits from Yin et al. (2021) as described below.

SMCALFLOW (Andreas et al., 2020) is a dataset of task-oriented natural language dialogs about calendars, weather, places, and people paired with executable dataflow programs. SMCALFLOW-CS (Yin et al., 2021) is a subset of SMCALFLOW containing single-turn dialogs involving two domains (organization structure and calendar event creation), each having its own set of program symbols with two types of test sets: a cross-domain (C) test set containing only instances where both domains appear and meant to test for compositional generalization, and a single-domain (S) test set contains instances with only single-domain for in-distribution evaluation. For compositional evaluation, we use the 32-C split which is a few-shot cross-domain split where the training set includes 32 cross-domain examples. For our IID evaluation, following Levy et al. (2022), we use the 8-S split. Additionally, we use the programs with the simplified syntax provided by (Meron, 2022).

Templates Tables 8, 9, and 10 contain the textual templates we use to linearize the instances for example selection and ICL. The ICL prompt is constructed by concatenating the templatized demonstrations and the test instance using `\n\n` as the separator.

B Additional Results

Tables 11, 12 and 13 show the complete 8-shot ICL results with GPT-Neo-2.7B, LLaMA-7B, and LLaMA-13B, respectively, for all the datasets we experiment we use in this work.

Additionally, Table 14 provides the time taken to select 8 ICL examples using every selection method.

B.1 Additional Qualitative Results

We qualitatively compare the gist token embeddings with ordinary token embeddings in Figure 6 which considers 3 types of pairwise distance distributions: NLP x NLP, Gist x Gist, and NLP x Gist. Clearly gist token activations are embedded into a different geometry when compared to ordinary language tokens.

Dataset	Selector Example Template	ICL Example Template
SMCALFLOW	`Great , thanks ! I am going to need a meeting with Karen , Jim , and Pam tomorrow before noon .`	`Great , thanks ! I am going to need a meeting with Karen , Jim , and Pam tomorrow before noon . CreateEvent( AND(with_attendee(" Pam "), with_attendee(" Karen "), with_attendee(" Jim "),starts_at( OnDateBeforeTime(date=Tomorrow(), time=Noon()))))`
MTOP	`call Nicholas and Natasha`	`call Nicholas and Natasha [IN: CREATE_CALL [SL:CONTACT Nicholas ] [ SL:CONTACT Natasha ] ]`
COGS	`Liam hoped that a box was burned by a girl .`	`Liam hoped that a box was burned by a girl . hope ( agent = Liam , ccomp = burn ( theme = box , agent = girl ) )`

COPA	`The man turned on the faucet.`
	`What is the most likely effect in the above sentence?`
	`Option A: The toilet filled with water.`
	`Option B: Water flowed from the spout.`
	`Answer:`
		`The man turned on the faucet.`
		`What is the most likely effect in the above sentence?`
		`Answer: Water flowed from the spout.`

CMSQA	`Select one of the choices that best answers the following question:`
	`Question: A revolving door is convenient for two direction travel, but it also serves as a security measure at a what?`
	`Option A: bank`
	`Option B: library`
	`Option C: department store`
	`Option D: mall`
	`Option E: new york`
	`Answer:`
		`Question: A revolving door is convenient for two direction travel, but it also serves as a security measure at a what?`
		`Option A: bank`
		`Option B: library`
		`Option C: department store`
		`Option D: mall`
		`Option E: new york`
		`Answer: A`

GSM8K	`Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?`	`Question: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?`
		`Solution: Janet sells 16 - 3 - 4 = <<16-3-4=9>>9 duck eggs a day.`
		`She makes 9 * 2 = $<<9*2=18>>18 every day at the farmer's market.`
		`#### 18`

AGNEWS	`Classify the following news article into one of these categories: World, Sports, Business, Technology.`
	`Fears for T N pension after talks Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.`
	`Category:`
		`Article: Fears for T N pension after talks Unions representing workers at Turner Newall say they are ' disappointed' after talks with stricken parent firm Federal Mogul.`
		`Category: Business`

COLA	`Is the following sentence grammatical ( Yes or No)?`
	`The sailors rode the breeze clear of the rocks.`
	`Answer:`
		`Sentence: The sailors rode the breeze clear of the rocks.`
		`Answer: Yes`

Table 8: The example templates we use for example selection and in-context learning for the various datasets other than NLI, Paraphrase Detection, Sentiment Analysis, and Reading Comprehension which are in Tables 9 and 10.

Dataset	Selector Example Template	ICL Example Template
QNLI	As of that day, the new constitution heralding the Second Republic came into force. Can we know "What came into force after the new constitution was herald?" given the above sentence (Yes or No)?	Question: What came into force after the new constitution was herald? Sentence: As of that day, the new constitution heralding the Second Republic came into force. Answer: Yes
MNLI	Premise: The new rights are nice enough Does the above premise entail the hypothesis that " Everyone really likes the newest benefits " (Yes, Maybe, or No)? Answer:	Premise: The new rights are nice enough Hypothesis: Everyone really likes the newest benefits Answer: Maybe
RTE	Dana Reeve, the widow of the actor Christopher Reeve , has died of lung cancer at age 44, according to the Christopher Reeve Foundation. Based on the above paragraph can we conclude that " Christopher Reeve had an accident." (Yes or No) ?	Premise: Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according to the Christopher Reeve Foundation. Hypothesis: Christopher Reeve had an accident.? Answer: No
WANLI	Premise: In the past, I have found that there is no point in making a speech unless you have prepared it. Is the hypothesis that "You should prepare a speech ." an entailment, contradiction or neutral with respect to the above premise? Answer:	Premise: In the past, I have found that there is no point in making a speech unless you have prepared it. Hypothesis: You should prepare a speech. Answer: entailment
XNLI	Premise: Et il a dit, maman, je suis à la maison. Is the hypothesis that "Il a appelé sa mère dès que le bus scolaire l'a déposé." an entailment, contradiction or neutral with respect to the above premise? Answer:	Premise: Et il a dit, maman, je suis à la maison. Hypothesis: Il a appelé sa mère dès que le bus scolaire l'a déposé. Answer: neutral
MEDNLI	Premise: No history of blood clots or DVTs, has never had chest pain prior to one week ago. Is the hypothesis that "Patient has angina." an entailment, contradiction or neutral with respect to the above premise? Answer:	Premise: No history of blood clots or DVTs, has never had chest pain prior to one week ago. Hypothesis: Patient has angina. Answer: entailment
MRPC	Sentence 1: He said the foodservice pie business doesn 't fit the company 's long-term growth strategy . Sentence 2: " The foodservice pie business does not fit our long-term growth strategy . Do the above sentences convey the same meaning? Yes or No. Answer:	Sentence 1: He said the foodservice pie business doesn 't fit the company 's long-term growth strategy . Sentence 2: " The foodservice pie business does not fit our long-term growth strategy . Answer: Yes
QQP	Question 1: Why are African-Americans so beautiful? Question 2: Why are hispanics so beautiful? Are Questions 1 and 2 asking the same thing? Yes or No. Answer:	Question 1: Why are African-Americans so beautiful? Question 2: Why are hispanics so beautiful? Answer: No
PAWS	Sentence 1: Bradd Crellin represented BARLA Cumbria on a tour of Australia with 6 other players representing Britain , also on a tour of Australia . Sentence 2: Bradd Crellin also represented BARLA Great Britain on a tour through Australia on a tour through Australia with 6 other players representing Cumbria . Are these sentences paraphrases of each other? Yes or No. Answer:	Sentence 1: Bradd Crellin represented BARLA Cumbria on a tour of Australia with 6 other players representing Britain , also on a tour of Australia . Sentence 2: Bradd Crellin also represented BARLA Great Britain on a tour through Australia on a tour through Australia with 6 other players representing Cumbria . Answer: No
SST2	Review: it 's a charming and often affecting journey . Is the sentiment of the above review Negative or Positive? Answer:	Review: it 's a charming and often affecting journey . Sentiment: Positive
SST5	Review: in his first stab at the form , jacquot takes a slightly anarchic approach that works only sporadically . Does the review above see the movie as terrible, bad , OK, good, or great? Answer:	Review: in his first stab at the form , jacquot takes a slightly anarchic approach that works only sporadically . Sentiment: OK

Table 9: The example templates we use for example selection and in-context learning for the various NLI, Paraphrase Detection, and Sentiment Analysis datasets.

Dataset	Selector Example Template	ICL Example Template
DROP	`Hoping to rebound from their loss to the Patriots, the Raiders stayed at home for a Week 16 duel with the Houston Texans. Oakland would get the early lead in the first quarter as quarterback JaMarcus Russell completed a 20-yard touchdown pass to rookie wide receiver Chaz Schilens. The Texans would respond with fullback Vonta Leach getting a 1-yard touchdown run, yet the Raiders would answer with kicker Sebastian Janikowski getting a 33-yard and a 30-yard field goal. Houston would tie the game in the second quarter with kicker Kris Brown getting a 53-yard and a 24-yard field goal. Oakland would take the lead in the third quarter with wide receiver Johnnie Lee Higgins catching a 29-yard touchdown pass from Russell, followed up by an 80-yard punch return for a touchdown. The Texans tried to rally in the fourth quarter as Brown nailed a 40-yard field goal, yet the Raiders' defense would shut down any possible attempt.` `How many field goals did both teams kick in the first half?` `Answer:`	`Passage: Hoping to rebound from their loss to the Patriots, the Raiders stayed at home for a Week 16 duel with the Houston Texans. Oakland would get the early lead in the first quarter as quarterback JaMarcus Russell completed a 20-yard touchdown pass to rookie wide receiver Chaz Schilens. The Texans would respond with fullback Vonta Leach getting a 1-yard touchdown run, yet the Raiders would answer with kicker Sebastian Janikowski getting a 33-yard and a 30-yard field goal. Houston would tie the game in the second quarter with kicker Kris Brown getting a 53-yard and a 24-yard field goal. Oakland would take the lead in the third quarter with wide receiver Johnnie Lee Higgins catching a 29-yard touchdown pass from Russell, followed up by an 80-yard punch return for a touchdown. The Texans tried to rally in the fourth quarter as Brown nailed a 40-yard field goal, yet the Raiders' defense would shut down any possible attempt.` `Question: How many field goals did both teams kick in the first half?` `Answer: 2`
BOOLQ	`Ethanol fuel -- All biomass goes through at least some of these steps: it needs to be grown, collected, dried, fermented, distilled, and burned. All of these steps require resources and an infrastructure. The total amount of energy input into the process compared to the energy released by burning the resulting ethanol fuel is known as the energy balance (or ``energy returned on energy invested''). Figures compiled in a 2007 report by National Geographic Magazine point to modest results for corn ethanol produced in the US: one unit of fossil-fuel energy is required to create 1.3 energy units from the resulting ethanol. The energy balance for sugarcane ethanol produced in Brazil is more favorable, with one unit of fossil-fuel energy required to create 8 from the ethanol. Energy balance estimates are not easily produced, thus numerous such reports have been generated that are contradictory. For instance, a separate survey reports that production of ethanol from sugarcane, which requires a tropical climate to grow productively, returns from 8 to 9 units of energy for each unit expended, as compared to corn, which only returns about 1.34 units of fuel energy for each unit of energy expended. A 2006 University of California Berkeley study, after analyzing six separate studies, concluded that producing ethanol from corn uses much less petroleum than producing gasoline.` `does ethanol take more energy make that produces (yes or no)` `Answer:`	`Passage: Ethanol fuel -- All biomass goes through at least some of these steps: it needs to be grown, collected, dried, fermented, distilled, and burned. All of these steps require resources and an infrastructure. The total amount of energy input into the process compared to the energy released by burning the resulting ethanol fuel is known as the energy balance (or ``energy returned on energy invested''). Figures compiled in a 2007 report by National Geographic Magazine point to modest results for corn ethanol produced in the US: one unit of fossil-fuel energy is required to create 1.3 energy units from the resulting ethanol. The energy balance for sugarcane ethanol produced in Brazil is more favorable, with one unit of fossil-fuel energy required to create 8 from the ethanol. Energy balance estimates are not easily produced, thus numerous such reports have been generated that are contradictory. For instance, a separate survey reports that production of ethanol from sugarcane, which requires a tropical climate to grow productively, returns from 8 to 9 units of energy for each unit expended, as compared to corn, which only returns about 1.34 units of fuel energy for each unit of energy expended. A 2006 University of California Berkeley study, after analyzing six separate studies, concluded that producing ethanol from corn uses much less petroleum than producing gasoline.` `Question: does ethanol take more energy make that produces` `Answer: no`

Table 10: The example templates we use for example selection and in-context learning for Reading Comprehension datasets.

Dataset	RANDOM	SBERT	BM25	BSR	GIST[FINETUNE]		GIST[MULTI]				EPR	CEIL
					$l=1$	$l=3$	$l=1$	$l=3$	$l=6$	$l=15$		
SMCALFLOW (CG)	0	1.1	3.3	3.9	4.4	5.6	0.9	2.4	3.8	3.8	3.6	3.8
SMCALFLOW (IID)	3.3	23.6	31.1	39.7	49.7	51.2	27	30.7	34.7	35.5	54.5	59.1
MTOP	1.3	44.6	48.3	53.9	60.1	61.7	51.3	52.4	52.9	53.5	62.2	60.5
COGS (CG)	3.8	21.1	26.7	49.6	62.7	67	36.3	38.8	41.9	41.1		
COGS (IID)	8.1	26.6	32.5	55	71.3	73.7	41.7	45.8	48.8	47.9		
QNLI	54.8	56.3	56.8	82.6	91.4	93	86.8	85.9	85.5	85.8	74.9	84.2
MNLI	41.9	44	42.2	76.7	82	81.4	78.1	76.6	78.5	74.6	66.1	71.7
RTE	53.4	54.2	50.9	67.9	81.6	81.2	83	77.6	81.2	73.3		
WANLI	38.8	42.6	44.4	60	66.2	65.4	58.2	53.8	53	54.8		
XNLI	32.9	35.8	35	49.3	68.1	70.2	61.5	58	61.8	58.4		
MEDNLI	41.4	56.9	54.2	70.6	82.9	83	69.4	71.1	69.5	70.4		
SST2	86.9	81.9	82.6	90.9	93.9	94.3	92.1	92.4	92.5	89.6		
SST5	13	37.9	38.9	45.1	50	52.6	48.4	49.3	45.9	45.3	42.8	47
MRPC	51	52.5	57.6	70.1	87.3	85.3	83.1	88	84.1	75	76	80.2
QQP	65.9	75	71.3	86.4	86.7	88.6	85.6	85.2	85.7	84.8		
PAWS	48	52.5	55.2	75	92.7	91.6	90.1	90.2	88.1	84.7		
COPA	57	58	57	57	58	56	56	58	55	57		
CMSQA	19	18.1	17.5	20.1	59.9	57.2	54.3	55.6	55	44.5	36.8	37.2
AGNEWS	76.6	89.3	89.4	89.9	92.1	92.5	91.4	90.4	90.5	90.7		
GSM8K	1.7	2	4	2.4	3.1	3.5	3.4	1.8	3.5	3.6		
DROP	7.7	12.6	12.5	10.7	25.4	28.7	18.5	18.8	19.7	18		
BOOLQ	39.3	47.3	49.6	50.4	69.5	66.3	65.2	65	66.3	59.7		
COLA	60.3	64.9	64.4	69.7	80	80.3	76.4	75.9	74.4	70.4		
AVG	35.0	43.4	44.6	55.5	66.0	66.5	59.1	59.3	59.7	57.5		
HELD-IN AVG	45.1	49.8	50.0	59.7	70.2	70.2	67.5	67.4	67.1	63.8		

Table 11: 8-shot ICL with GPT-Neo-2.7B. l is the number of gist tokens. Red highlights datasets or tasks that are held-out from our multi-task pre-training collection. HELD-IN AVG is the average performance excluding these dataests.

Dataset	RANDOM	SBERT	BM25	BSR	GIST[FINETUNE]		GIST[MULTI]				LLM-R
					$l=1$	$l=3$	$l=1$	$l=3$	$l=6$	$l=15$	
SMCALFLOW (CG)	0	3	9.4	10.6	8.3	9.2	4.5	7.1	9.8	9.4	
SMCALFLOW (IID)	6.8	38.1	46.4	55.1	62.2	63.3	43.2	45.5	48.2	46.4	
MTOP	3.4	49.8	55.1	59.7	64.7	65.8	55.2	57.2	60	58.4	
COGS (CG)	13.2	29.3	31.4	55.2	69.2	71.5	41.3	46.2	48.9	47.7	
COGS (IID)	10.6	35.8	39	66.8	79.1	83.2	54.3	57.9	59	59.5	
QNLI	51.5	56.8	57.4	75.3	87.7	90.2	80.1	82.2	81.7	79.1	69.4
MNLI	54.3	58	56.1	76.3	80.8	80.1	78.5	76	77.4	76.2	69.8
RTE	70	67.9	68.2	70.8	84.5	84.8	85.6	80.1	81.6	78.7	70.4
WANLI	38.5	40.3	40.3	51.4	56.7	59.3	49.8	47.9	45.6	47.4	
XNLI	32.2	33.3	33.1	42.5	58.1	61.4	52.3	49.6	49.1	47.7	
MEDNLI	33.6	37.8	35.9	49.8	77.7	75.7	56.8	56.7	55.2	54.8	
SST2	94.2	92	93.2	95.8	94.6	94.7	95.2	94.6	94.6	94.2	93.1
SST5	38.4	43.2	45.2	40.7	46.8	51.2	45.9	44.8	45.1	45.6	
MRPC	33.8	46.6	48.3	59.8	82.4	77.5	77.9	80.6	78.2	67.9	78.2
QQP	66.2	76.1	73.2	80.4	83.7	84.1	82	80.1	79.7	80.2	83.3
PAWS	59.1	56.6	57.2	74	90.7	89.3	86.3	88.1	87.2	80.6	57
COPA	83	87	86	86	85	83	90	91	87	88	84
CMSQA	39.9	29.9	26.2	30.3	63.7	60	60.1	63.4	62.1	49.2	
AGNEWS	85.7	86.8	88.2	88.9	90.7	92.4	90.4	90.4	90.1	88.2	93.5
GSM8K	11	12.3	12.4	14.3	12.6	14.1	15.6	14	14.2	13.3	
DROP	24.4	27.6	28.5	27.4	36.5	39.2	32.7	32.2	31.9	31.4	
BOOLQ	71.2	73.4	75.5	77.6	82.8	82.4	81.8	80.4	81.1	77.5	74.1
COLA	60.1	70.3	67	70.3	77.4	77.5	74.4	71.9	73.8	72.4	
AVG	42.7	50.1	51.0	59.1	68.5	69.1	62.3	62.5	62.7	60.6	
HELD-IN AVG	56.2	59.0	58.8	64.5	73.3	73.4	71.8	71.3	71.0	68.2	

Table 12: 8-shot ICL with LLaMA-7B. l is the number of gist tokens. Red highlights datasets or tasks that are held-out from our multi-task pre-training collection. HELD-IN AVG is the average performance excluding these dataests.

Dataset	SBERT	BM25	BSR	GIST[Finetune]		GIST[Multitask]			
				$l=1$	$l=3$	$l=1$	$l=3$	$l=6$	$l=15$
SMCALFLOW (CG)	4.5	13.9	15.5	11.6	14.8	7.2	11	13.9	13.4
SMCALFLOW (IID)	41.5	48.8	60.7	66.2	66	48.9	50	51.5	52.4
MTOP	55	59.5	65.6	68.2	69.2	61.3	62.4	65	62.6
COGS (CG)	30.7	34	58.2	71.6	74	44.5	46.7	48.5	49.4
COGS (IID)	39.5	44.2	69.8	81.3	84.2	56.2	57.1	63	62.4
QNLI	60.3	59.3	81.6	91.6	92.2	86	86.6	85	85.7
MNLI	62.5	62.8	81.9	83.3	80.8	81.6	80.8	80.7	77.9
RTE	76.2	74.7	75.8	85.6	84.1	85.9	82.7	83.4	80.5
WANLI	22.4	23.8	21.7	27.5	27.3	23.8	23.3	21.7	21.9
XNLI	38.7	40.2	37.9	40.9	40.2	39.3	40	40	39.4
SST2	92.3	92.4	95.1	94.3	95.1	94.8	94.6	94.4	93.1
SST5	46.8	46.7	42.5	46.4	48.2	44.4	42.7	46.9	42.6
MRPC	57.4	63	72.8	87.3	86	86.8	87	86	76.2
QQP	78	77.4	85.1	85.9	86.9	84.3	83.4	83.7	84.4
PAWS	59.6	58.2	77	92.4	92	89.9	89.8	88.8	84.7
COPA	51	53	52	51	52	52	52	52	53
CMSQA	44.2	41.3	41.7	65.1	62.4	64.5	67.7	67.5	59.5
AGNEWS	91.5	91.8	91	93.1	94	92.7	93.1	92.7	91.2
BOOLQ	74.8	76.8	73.2	81.7	81.7	82.8	81.8	82.5	80.2
COLA	71.1	65.1	72.9	80.2	79.1	75.2	75	76.6	73.2
AVG	54.9	56.3	63.6	70.3	70.5	65.1	65.4	66.2	64.2
HELD-IN AVG	66.6	66.3	72.5	79.8	79.6	78.5	78.2	78.5	75.6

Table 13: 8-shot ICL with LLaMA-13B. l is the number of gist tokens. Red highlights datasets or tasks that are held-out from our multi-task pre-training collection. HELD-IN AVG is the average performance excluding these dataests.

Dataset	SBERT	BM25	BSR	GIST[Finetune]		GIST[Multitask]			
				$l=1$	$l=3$	$l=1$	$l=3$	$l=6$	$l=15$
SMCALFLOW	10.6	117.2	2212.3	1.0	1.4	1.0	1.4	2.1	3.8
MTOP	6.6	36.8	540.5	0.8	1.1	0.8	1.1	1.5	2.6
COGS	10.0	82.7	528.5	1.0	1.4	1.3	1.5	2.2	3.9
QNLI	18.2	1416.9	10933.5	1.5	2.0	1.5	2.3	3.7	7.0
MNLI	20.7	1469.9	9565.4	1.6	2.0	1.5	2.5	3.3	6.9
RTE	1.0	68.8	696.7	0.6	0.8	0.8	0.8	0.7	0.9
WANLI	19.5	1351.0	6556.2	1.5	2.0	1.5	2.2	3.8	6.9
XNLI	20.0	1856.5	24424.5	1.7	2.0	1.6	2.2	3.3	7.0
MEDNLI	4.8	285.6	3357.4	0.7	0.9	0.7	1.1	1.3	2.1
SST2	22.4	1119.4	3639.3	1.5	2.1	1.5	2.3	3.4	7.0
SST5	5.0	296.0	609.4	0.8	1.2	0.6	0.9	1.1	1.7
MRPC	1.3	89.6	255.7	0.5	0.7	0.7	0.7	0.8	1.0
QQP	20.1	1336.2	8862.5	1.6	2.1	1.6	2.1	3.6	6.9
PAWS	20.9	1350.4	6712.2	1.6	2.0	1.7	2.2	3.5	6.9
COPA	0.3	4.9	22.6	0.6	1.0	0.6	0.7	0.7	0.8
CMSQA	4.2	290.7	1124.5	0.7	0.9	0.6	0.9	1.2	1.9
WINOGRANDE	22.6	1208.3	2707.8	4.6	2.0	1.4	2.2	3.1	6.4
AGNEWS	20.8	2098.1	11812.8	1.5	2.0	1.6	2.5	3.4	6.9
GSM8K	3.2	138.6	1605.9	0.6	0.9	0.6	0.8	1.0	1.7
DROP	29.7	5068.6	22340.1	1.5	2.0	1.5	2.2	3.2	6.8
BOOLQ	3.8	413.5	4876.0	0.7	0.9	0.6	0.9	1.1	1.8
COLA	3.8	109.7	644.4	0.7	0.9	0.6	0.9	1.0	1.7

Table 14: Time (in ms) to select 8-shots for the various datasets using the different training-free methods. The time for SBERT is higher than gisting-based retrieval because our implementation for it does not use FAISS indexing.

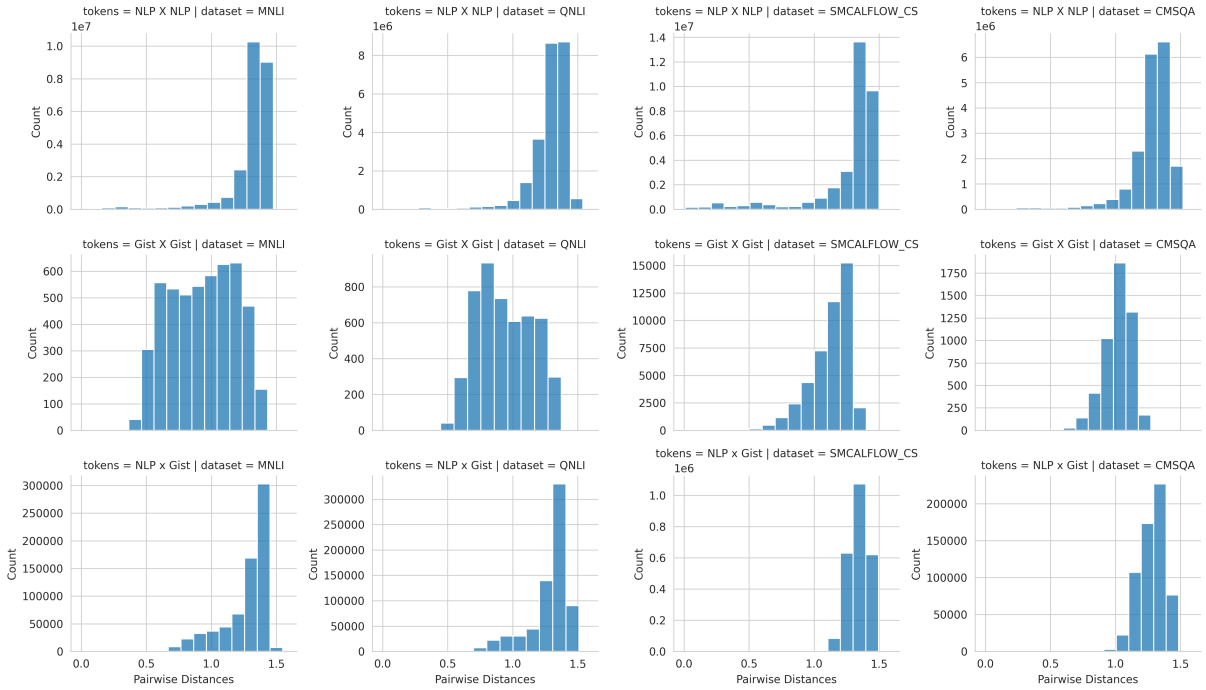

Figure 6: Pairwise Distances between Gist and NLP token activations.