KV-DISTILL: Nearly Lossless Context Compression for LLMs

Anonymous ACL submission

Abstract

Sequence-to-sequence tasks often benefit from long contexts, but the quadratic complexity of self-attention in standard Transformers renders this non-trivial. During generation, temporary representations – stored in the so-called KV cache – account for a large portion of GPU memory usage and scale linearly with context length. We introduce KV-DISTILL, a Transformer compression framework that distills long context KV caches into significantly shorter representations in a *question-independent* fashion. KV-DISTILL can be trained as a parameter-efficient adaptor for pretrained models, and enables the compression of arbitrary spans of a context while preserving pre-trained model capabilities. We treat a compressed-uncompressed cache as a student-teacher pairing and apply a KL-type divergence to match the generated outputs. KV-DISTILL outperforms other compression techniques in worst-case extractive tasks and approaches uncompressed performance in long context question answering and summarization, and it can be fine-tuned on domain-specific contexts to reduce lengths by up to 99% while preserving downstream performance. We demonstrate the generalizability of KV-DISTILL across various model sizes and architectures.[1]

1 Introduction

Harnessing the full potential of attention-based large language models (LLMs) often requires them to condition on long contexts. However, use of expansive contexts is complicated by the quadratic complexity of self-attention. In particular, during generation, one must maintain a store of all past key and value representations of past tokens (called the KV cache) that grows linearly with sequence length. The memory burden imposed by the KV cache is significant, and often limits the length of the sequences that a model can handle.

Much work has been devoted to architectural improvements to attention in order to reduce memory during generation. Strategies include augmenting sequences with memory tokens (Rae et al., 2020; Wu et al., 2022), sparsifying attention patterns (Beltagy et al., 2020), and using conditional computation to only process essential tokens (Ainslie et al., 2023). However, such techniques have seen little widespread adoption due to performance drops on downstream tasks, or inefficient training/inference procedures. Even when given long contexts without compression, LLMs fail to fully utilize them (Qin et al., 2022; Liu et al., 2024; Lu et al., 2024). Together this suggests long contexts may allow for significant compression while yielding large memory savings.

In what follows, we suppose that a prompt to a LLM is composed of contextual text(s) followed by a question whose answer is dependent on the provided context. KV compression can be divided into two paradigms: *question-aware*, and *question-independent*. In question-aware compression, we have access to the question that we need answered, and can compress the context with this in mind. In question-independent compression, we do not know what questions will be asked in the future. For instance, consider a scenario in which a fixed textual context will be used to respond to many questions; the goal of question-independent compression is to compress this context once for reuse across many question.

Prior work in training-free context compression has primarily focused on which representations in the KV cache to select for eviction, with excellent results (Zhang et al., 2023; Li et al., 2024). However, in practice, we observe that the performance of this selection procedure suffers greatly in the question-independent paradigm. Furthermore, we anticipate that there is room for performance improvements in general-purpose context compression when the model is trained to handle for com-
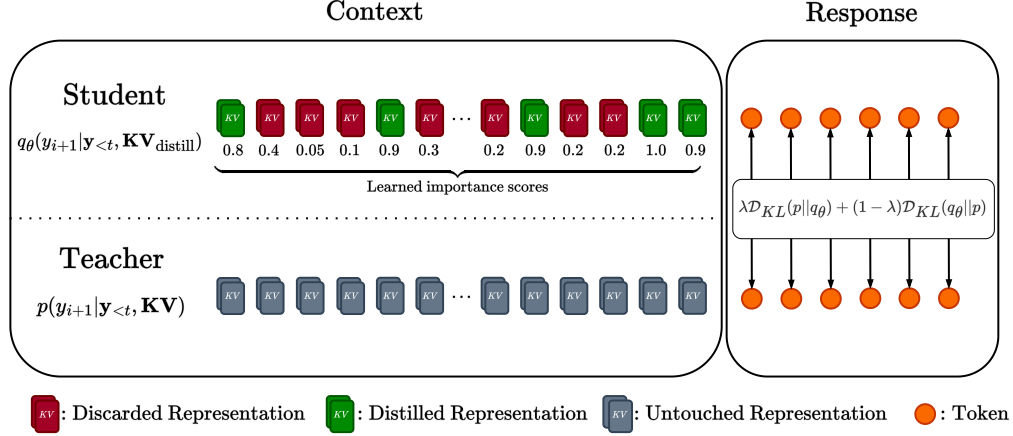
---

[1]Our code and checkpoints will be made available at https://example.com

Figure 1: We subselect tokens from the KV cache and distill into the smaller subset

pression.

Prior workin trainable context compression have typically utilized a combination of cross-entropy and autoencoding objectives to pre-train general context compressors (Qin et al., 2024; Ge et al., 2024; Rae et al., 2020), which are suitable for question-independent compression. These loss functions have led to significant performance loss at high compression rates.

In this work we design a general-purpose trainable context compression method for LLMs that outperforms prior methods in both the *question-independent* and *question-aware* paradigms. Our method, KV-DISTILL , accomplishes this, while also maintaining pretrained model capabilities, being suitable for long contexts, and having minimal performance penalty on downstream tasks. KV-DISTILL can support coherent, useful generation at compression ratios as high as 1000x.

To achieve this we train a scorer which retains the most important context tokens, while applying a parameter efficient adapter to conditionally modify important tokens' activations in-place. We further apply a token-level KL-type divergence to match the next-token prediction distributions, treating the compressed cache as a student, and the uncompressed cache as a teacher. KV-DISTILL only need be applied once to a fixed context, has zero overhead during auto-regressive decoding, and can compress arbitrary (sub)spans of a given context. We show improvements on several model families, considering extractive and abstractive tasks, with both short and long contexts, and at multiple model scales. KV-DISTILL is general purpose and has broad applicability to the LLM community.

2 Background

2.1 Key-Value Cache

Transformer-based language models (LMs) (Vaswani et al., 2017) use self-attention to aggregate context information and make predictions. A decoder-only transformer LM *autoregressively* predicts new tokens, and each step requires the LM to obtain the key and value states of all past tokens. To avoid re-computing the KV state of past tokens, most LM implementations (e.g. Wolf et al. (2020)) cache the key and values states, in a structure called the KV cache. When making new predictions, self-attention is performed on query states of the new token and the KV -cache, and the new token's key and value representations are appended to the KV cache. Because the KV cache grows proportional to the number of tokens generated, maintaining the full KV cache in memory is a primary bottleneck when conditioning on large contexts. The goal of this work is to alleviate this by *compressing KV cache in the dimension of sequence length*, especially in the question-independent regime

2.2 Related Work

Much prior work has tackled the problem of reducing the complexity of the self-attention mechanism itself. Previous work tries to sparsify the attention patterns(Beltagy et al., 2020; Zaheer et al., 2020), use recurrence attention(Yang et al., 2019), or kernelize the attention matrix(Choromanski et al., 2021), but they require a considerable amount of further training.

Similar to our work, one line of work involves compressing the hidden states (KV cache) of past

2

tokens into a shorter sequence of representations. For example, some methods learns "soft representations" of context (Qin and Eisner, 2021). Mu et al. (2023) compress particular prompts into much shorter "gist tokens", but do not attempt more general context compression. Furthermore, their method demonstrates poor generalizability, as performance does not scale with the number of gist tokens used. Zeng et al. (2023) propose to recognize and prioritize some important tokens (VIP tokens) during inference. Most relevant here, the following methods employ a similar idea of dynamically compressing the context prior to inference.

2.3 Trainable Compression

Ge et al. (2024) design In-Context Autoencoder (ICAE) to compress long contexts for use in large language models (LLMs). ICAE consists of two main components: a learnable encoder and a fixed decoder. The encoder compresses the input context into a small number of memory slots. These memory slots are then used by the frozen LLMs (decoder) to reconstruct the context or respond to prompts. ICAE is pretrained using autoencoding and language modeling objectives on a large pre-training corpus and further fine-tuned using instruction data to maintain instruction-tuning. However, there is still a gap in downstream task performance when using an ICAE-compressed context, compared to an uncompressed context, and the method falters under high compression ratios.

Qin et al. (2024) propose DODO to compress sub-select KV activations to a set of "nugget" tokens, which grow proportionally with the length of context sequence. Their method is trained with auto-encoding or language modeling objectives. However, DODO models operate at a fixed compression ratio, require training both an encoder and decoder, and still show a large gap in downstream task performance when compared to an uncompressed context.

2.4 Training-Free Compression

Zhang et al. (2023) propose H2 to reduce memory usage during generation. H2 identifies "heavy-hitter" tokens, which significantly influence attention scores during inference. Specifically, H2 calculates the accumulated attention for each key and retains the top-k key-value pairs with the highest scores.

In the question-aware setting, the accumulated attention scores include attention scores from tokens in the question attending to the context. This effectively uses the question to scan for important details in the context. This allows H2 to maintain nearly uncompressed performance at moderate compression ratios, by focusing on tokens most relevant to the current question. However, performance still degrades when compression ratios exceed 20×.

In the question-independent paradigm, the H2 selection mechanism is applied solely to the context (as opposed to the context and question in the question-aware setting). We then allow the question to attend to only to this compressed context. We empirically observe that in the question-independent paradigm, H2 performance plummets drastically, highlighting the need for improved question-independent compression methods. Lastly, H2 offers no way to further improve compressive performance given prior domain knowledge.

Similarly SnapKV (Li et al., 2024) uses the attentions of a window of recent tokens to determine which context tokens are "heavy-hitters"; in the question-independent setting, this is undesirable, as the last tokens of a context may not necessarily provide additional information regarding attention patterns. In the question-independent paradigm we find that SnapKV performs similarly to H2 , so do not compare against it in the remainder of this paper.

3 Key-Value Distillation

We consider a transformer-based language model (Vaswani et al., 2017), denoted by LM, that is defined on the vocabulary $\mathcal{V}$. The KV-DISTILL process is then: (1) a set of important tokens in the input context is determined; (2) an adapted language model LM_θ is used to encode the context into a KV cache, and sub-select the aforementioned important tokens from the generated KV cache; and (3) the unmodified LM conditions on the compressed KV cache to auto-regressively generate it's output.

3.1 Important State Selection for Cache Compression

Let $\mathbf{c} = \{w_i\}_{i=1}^N$ represent a context consisting of N tokens, where $w_i \in \mathcal{V}$ and $\vec{c} \in \mathcal{V}^N$. In a typical scenario, LM predicts a sequence of new tokens, denoted by $\vec{y}$, conditioning on $\mathbf{c}$. For example, $\mathbf{c}$ may be a prompt and LM generates $\vec{y}$ as a response. Future token prediction draws on information from past tokens via *attention* by having LM encode the

context tokens into key and value hidden states $\vec{X}_l^{(K)}, \vec{X}_l^{(V)} \in \mathbb{R}^{N \times d}$, which taken together form the KV cache (Section 2.1), where d is the dimension of the transformer and l is the layer of LM. We may drop the subscript l and superscripts $^{(K)}$ and $^{(V)}$ and use $\vec{X}$ to generally denote the key/value states of transformers at any layer.

Transformers assume that $\vec{X}$ fully describes and represents the context $\vec{c}$. However, attending to $\vec{X}$ can be inefficient when $\vec{c}$ is long. Therefore, we further assume that retaining a subset of key/value states is sufficient for approximating the next-token distribution conditioned on all key/value states. That is, we could *retain* rows from $\vec{X}$ to form $\tilde{\vec{X}} \in \mathbb{R}^{k \times d}$, where $k \le N$ is the number of selected rows. We use a subset of the tokens' hidden states to represent the complete context, which is plausible because representations in $\tilde{\vec{X}}$ are conditioned on the prior context. Suppose we determine the $(i_1, \dots, i_k)$-th tokens are to be retained in layer l. We use a hard selection matrix $\vec{S}_l \in \{0,1\}^{k \times N}$ to derive (layer-specific) $\tilde{\vec{X}}_l$ from $\vec{X}_l$ by

$$\tilde{\vec{X}}_l = \vec{S}_l \vec{X}_l, \quad \vec{S}_l = [\vec{e}_{(i_1)}, \dots, \vec{e}_{(i_k)}], \quad (1)$$

where $\vec{e}_{(i)} \in \{0,1\}^N$ is the i-th standard basis vector. Note that this formulation does not require that the same tokens be selected across each layer.

The problem of determining which indices $(i_1, \dots, i_k)$ to retain still remains. We would like the subselection $\vec{S}$ to retain most of the context information given a fixed k. One possibility is to use a feedforward neural network to measure the importance of each token position:

$$\vec{s} = \text{FFN}_\theta \left(\vec{X}'_\eta \right) \quad (2)$$

where θ is the parameters of the FFN, $\vec{s} \in \mathbb{R}^N$ and $\vec{s}_i$ indicate the "importance score" of the i-th token and $\vec{X}'_\eta$ indicates the hidden states at the η-th layer. The indices $i_{1:k}$ can then be derived by taking the tokens with the top-k scores. We can control the percent of the KV cache retained by scaling k with the length of the context. In our we retain the same $i_{1:k}$ across all layers and take $\eta = 6$.

The above selection procedure is rendered non-differentiable by the top-k operator. We may propagate gradients to the scorer by decaying the attention weights of tokens attending to $\tilde{\vec{X}}$ inversely proportionally to their computed importance scores. More precisely, let $\vec{z} \in \mathbb{R}^d$ represent the hidden
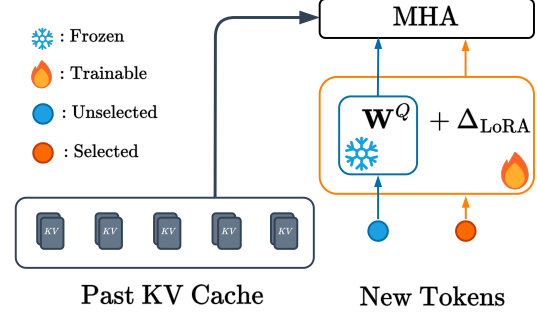

Figure 2: Selected tokens are routed to trainable, LoRA-adapted $\vec{W}^Q$ and $\vec{W}^O$ matrices ($\vec{W}^O$ is omitted in this figure); all other tokens pass through the original (frozen) model parameters.

state of a single token attending to $\tilde{\vec{X}}$ with unnormalized attention weights α:

$$\alpha = \left(\vec{z} \vec{W}^Q \right) \left(\tilde{\vec{X}}^{(K)} \right)^\top \quad (3)$$

we decay α to produce scorer-informed attention weights α':

$$\alpha' = \sigma(\vec{s}) \odot \alpha, \quad (4)$$

where $\odot$ denotes the element-wise (Hadamard) product and σ the sigmoid function. We note that the above formulation is one of many possible scoring functions that can be used with KV-DISTILL, that could be learnable or parameter-free, and could potentially have layer-wise specificity. We leave exploring different scoring functions as future work.

3.2 Architecture

After performing sub-selection to determine important token indices, we pass the context $\vec{c}$ through a modified LM_θ that uses *conditional computation* to condense the context into $\tilde{\vec{X}}$. This allows for the representations of important tokens to be "packed" with information from unselected tokens, and is strictly more expressive than only subselection. We instantiate LM_θ with LoRA adaptors (Hu et al., 2022) to minimize the number of trainable parameters.

More importantly, within LM_θ, the subselected tokens are routed to trainable $\vec{W}^Q, \vec{W}^O$ matrices, where $\vec{W}^Q, \vec{W}^O$ are the query and output matrices of transformers, while discarded tokens are routed to the original (frozen) matrices, as shown in Figure 2. This has the effect of informing LM_θ as to which tokens are selected, allowing for specialized aggregation of the value representations for selected tokens. This method of informing LM_θ has

4

minimal overhead (the LoRA matrices account for under 500MB of GPU memory for a 27B parameter model), and only a single set parameters must be maintained in memory.

We anticipate that other architectural forms could make KV-DISTILL effective. However, we find that applying conditional computation to inform LM_θ of selected tokens is important to the performance of KV-DISTILL . We find that some methods of informing the model of selected tokens, such as by adding a trainable embedding to these tokens, do not work well (see Appendix B). The particular architecture chosen has the advantage of lower memory usage during training, and provides excellent performance. We leave the task of finding even more efficient architectures to future work.

3.3 Objective Function

After generating compressed cache $\tilde{\vec{X}}$, we aim to match the output of LM when conditioned on $\tilde{\vec{X}}$ to the output of LM when conditioned on $\mathbf{X}$. Previous compression methods (Ge et al., 2024; Qin and Van Durme, 2023) rely on the autoencoding objective to pretrain LM_θ. However, given that LM predicts future tokens during inference, there is a discrepancy in pretraining and downstream usage, which could result in performance loss. Instead we propose matching the next-token probability distribution of tokens conditioned on $\vec{X}$ and $\tilde{\vec{X}}$. Consider a generative language model that predicts the next token $\vec{y}_t$ conditioned on the past tokens $\vec{y}_{<t}$ and a fixed context $\vec{c}$ that is represented by either $\vec{X}$ or $\tilde{\vec{X}}$. We would like to minimize the difference between their next-token distributions, i.e. $p\left(\vec{y}_t \mid \vec{y}_{<t}, \vec{X}\right)$ and $q_\theta\left(\vec{y}_t \mid \vec{y}_{<t}, \tilde{\vec{X}}\right)$. Let q_θ indicate the distribution that conditions on the distilled KV cache $\tilde{\vec{X}}$. Also note that the only learnable parameters in this formulation arise from *encoding* $\tilde{\vec{X}}$; during auto-regressive generation we use the original frozen parameters of LM.

Given probability distributions p, q_θ, we use the forward and reverse KL divergences to measure their similarity. With simplified notations we have:

$$\mathcal{D}_{\text{KL}}(p\|q_\theta) = \mathbb{E}_{y \sim p(\cdot)}\left[\log\left(\frac{p(y)}{q_\theta(y)}\right)\right]$$

$$\mathcal{D}_{\text{KL}}(q_\theta\|p) = \mathbb{E}_{y \sim q_\theta(\cdot)}\left[\log\left(\frac{q_\theta(y)}{p(y)}\right)\right] \quad (5)$$

The mode-seeking and mean-seeking behavior of the reverse- and forward- KL divergences respectively is well known. To incorporate both behaviors into the objective, we sum the forward and reverse divergences:

$$\mathcal{L}(\theta) = \lambda \cdot \mathcal{D}_{\text{KL}}(p\|q_\theta) + (1-\lambda) \cdot \mathcal{D}_{\text{KL}}(q_\theta\|p), \quad (6)$$

where a hyperparameter λ controls the balance between forward and reverse KL divergence.

Given both p and q_θ are categorical distribution, both KL divergences in eq. (5) can be analytically solved. Tthe L1-norm of the gradient of the reverse divergence dominates nearly everywhere. As such we propose scaling the forward and reverse terms by having $\lambda > 0.5$ in eq. (6). The benefit of λ is confirmed with the ablations in Appendix B.

4   Experiments

To assess the efficacy of KV-DISTILL , we conduct experiments on LLAMA-2 7B, LLAMA-3 8B, MISTRAL 7B, GEMMA-2 9B and GEMMA-2 27B. In all cases we use the instruction-tuned model. A KV-DISTILL model is obtained by distilling on a large corpus to obtain strong general-purpose context compressors.

Data   We curate a large instruction dataset from Self-Instruct, P3, LongAlpaca, and Super-Natural Instructions (Soboleva et al., 2023; Wang et al., 2022a; Sanh et al., 2021; Chen et al., 2023; Wang et al., 2022b). Training instances are split into (Context, Instruction, Answer) triples.   In cases where the context is sufficiently long (more than 1536 tokens), we pad to a multiple of 1536 and fold the context to a batch of $N \times 1536$ instances, compress the resulting KV cache, and then unfold the cache. Empirically, we observe little performance degradation when applying folding during pretraining, while allowing the model to see longer examples. We also always leave the first few (< 10) tokens of the context uncompressed, as we find that retaining them improves performance; this is not a new observation, see Han et al. (2024) and Xiao et al. (2023).

Training   The general training procedure is as follows: (1) pass the (Context, Instruction, Answer) triple through the original model to obtain target logits, (2) apply the KV-DISTILL architecture to obtain logits conditioned on the compressed cache (we compress the context, and leave the instruction uncompressed), (3) apply Equation 6 between the obtained and target logits.

We use rank-stabilized LoRA on the Q, K, V, O matrices with $r = 128$ to train LM_θ (Hu et al.,
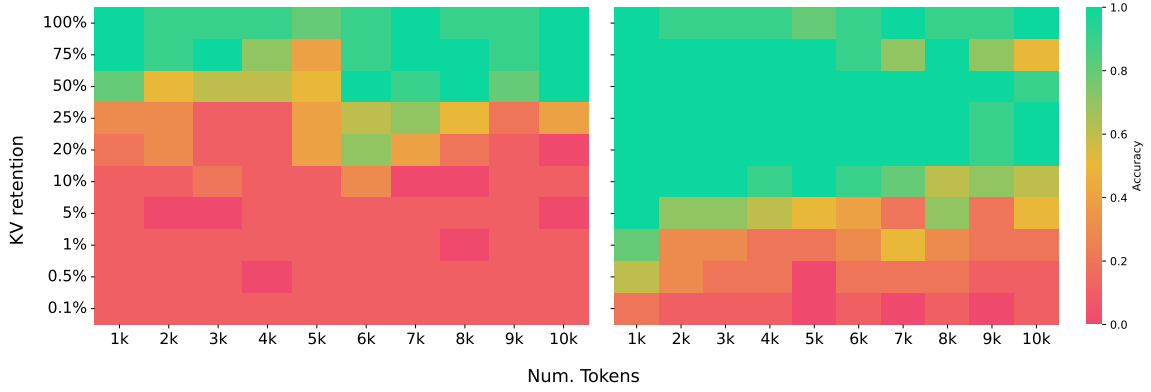
Figure 3: Needle-in-a-Haystack results; The x-axis shows the length of the document, the y-axis indicates the compression ratio applied, and the color the accuracy of retrieval under those settings averaged across different locations in the document. **Left:** H$_2$I. **Right:** KV-DISTILL .

2022; Kalajdzievski, 2023). Note that the K, V are trainable for all tokens, not just selected tokens. The behavior of the Q, O adapters is discussed in Section 3.2. Optimization is done using Deepspeed Stage 2, and the AdamW optimizer (Rasley et al., 2020). During pretraining, we sample KV retention fractions between 0.1-80%. As such, all KV-DISTILL models support arbitrary retention rates. All models are distilled on a cluster of 8 NVIDIA A100 80GB GPUs. All models except GEMMA 27B converged within 3 days, while GEMMA 27B took 4 days. See Appendix A for further details.

Dataset	Average	Max
SQuAD	225	1k
QuALITY	6k	9k
SQuALITY	7k	11k
GovReport	10k	71k

Table 1: Evaluation Dataset Statistics

Evaluation In all evaluation datasets, we have a natural (Context, Question) pairing. We always compress the context and leave the question uncompressed. All evaluations are performed with greedy decoding. Summary statistics regarding the context length of evaluation dataset are provided in Table 1.

Methods Tested We evaluate against DODO (Qin et al., 2024), ICAE (Ge et al., 2024), and H$_2$ in both the question-aware (H$_2$A) and question-independent (H$_2$I) forms. Please refer to Sec 2.2 for further description about the selected methods. In practice, to improve the performance of H$_2$, we also retain a set of sink tokens as described in Xiao et al. (2023).

5 Results

5.1 Needle-In-a-Haystack

Motivation The Needle-in-a-Haystack test (Kamradt, 2023) evaluates a model's ability to accurately retrieve information from a sentence (a "needle") embedded within a large document (a "haystack"), in which a sentence is randomly positioned. In Figure 3, we show the results of H$_2$I (left) and KV-DISTILL (right) at various compression ratios at different document lengths. The accuracy is computed across the placement of the needle within a document. Crucially, at compression time, the model does not know that a is needle being sought, nor that a needle is placed within the context.

Results We see that KV-DISTILL significantly outperforms H$_2$I at almost all compression ratios and document lengths. In particular, KV-DISTILL demonstrates near-perfect accuracy even after removing 90% of the KV .

5.2 Extractive Question Answering

Motivation SQuAD is an extractive question-answering task. We hypothesize that extractive tasks will suffer the largest performance loss under context compression. As such, we choose to use performance on SQuAD as a proxy for general-purpose compressive ability of a model. In all the following experiments, we choose the pretraining checkpoint with the best SQuAD performance for further experimentation. To assess accuracy we generate an answer conditioned on the compressed context, checking if the generated response is contained in the ground-truth answer.

Results Table 2 contains SQuAD accuracy re-

6

Model		% KV	0-Shot Acc.
LLAMA3	BASE	100%	$87.6 \pm .6\%$
	KVD	25%	$86.6 \pm .7\%$
	KVD	20%	$86.0 \pm .7\%$
	H_2A	25%	$84.0 \pm .7\%$
	H_2A	20%	$83.0 \pm .7\%$
	H_2I	25%	$56.6 \pm .9\%$
	H_2I	20%	$51.7 \pm 1\%$
	DODO	20%	$73.3 \pm .8\%$
LLAMA2 7B	BASE	100%	$82.5 \pm .7\%$
	KVD	25%	$79.1 \pm .8\%$
	KVD	20%	$77.6 \pm .8\%$
	H_2A	25%	$77.9 \pm .7\%$
	H_2A	20%	$76.7 \pm .7\%$
	H_2I	25%	$55.2 \pm .9\%$
	H_2I	20%	$50.3 \pm 1\%$
	ICAE	57%	$75.0 \pm .8\%$
GEMMA 9B	BASE	100%	$85.15 \pm .7\%$
	KVD	25%	$84.55 \pm .7\%$
	KVD	20%	$83.1 \pm .7\%$
GEMMA 27B	BASE	100%	$85.3 \pm .8\%$
	KVD	25%	$83.1 \pm 1\%$
	KVD	20%	$82.2 \pm 1\%$
MISTRAL 7B	BASE	100%	$87.1 \pm .6\%$
	KVD	25%	$84.1 \pm .7\%$
	KVD	20%	$82.5 \pm .7\%$

Table 2: Zero-shot accuracy on SQuAD at selected KV retention ratios.

sults. We see that in all cases, KV-DISTILL models perform within a few percentage points of base models, even under a "worst-case" task. Furthermore, KV-DISTILL models significantly outperform prior trainable methods (ICAE, DODO), even when retaining less of the KV cache. KV-DISTILL models significantly outperform H_2I, demonstrating the ability of the pretraining objective and architecture to create reusable compressed KV representations. KV-DISTILL models also enjoy a slight improvement over H_2A at similar compression ratios, demonstrating the effectiveness of its compression at capturing almost all salient information in the context, even without question-awareness.

When retaining under 20% of KV, we observe rapid declines in performance across all methods, indicating the difficulty of the task under high context compression. Lastly, we note that initial pretraining hyperparameters for all models were set based on initial experimentation with LLAMA-3 and SQuAD; as such, we anticipate that performance of most models can be improved with hyper-
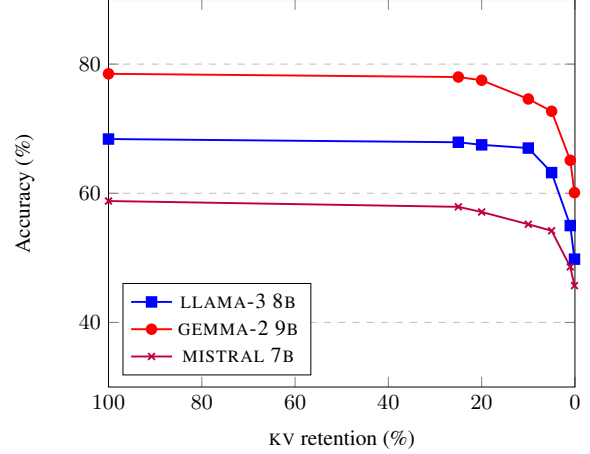

Figure 4: Accuracy on QuALITY

parameter tuning during the pre-training process.

5.3 Long Context Question Answering

Motivation QuALITY is a long document multiple-choice question answering dataset that assesses reading comprehension. We use QuALITY to assess the decision making capabilities of models equipped with distilled contexts. To assess QuALITY accuracy, we use the same evaluation procedure used by LLAMA-3 (AI@Meta, 2024).

Results Figure 4 shows the experiment results on QuALITY, with data points at the following retention rates highlighted: $\{100\%, 25\%, 20\%, 10\%, 5\%, 1\%, 0.1\%\}$. We observe that KV-DISTILL performs similarly to the uncompressed cache, with only minor losses in performance at 10x compression. Although not included in Figure 4, 0% cache retention results in accuracy of 32.4%, 25.8%, and 24.4% for the LLAMA-3, MISTRAL, and GEMMA-2 models respectively, demonstrating the neccessity of the context for the task. Impressively, we see significant improvements over the random accuracy even when distilling to as few as 7 tokens from a 7k input passage; for example, on LLAMA-3 we observe only a 20% drop in accuracy despite eliminating 99.9% of the context.

5.4 Long Context Abstractive Summarization

Motivation SQuALITY is a question-focused summarization dataset based on the same collection of long documents as the QuALITY benchmark. We use it to evaluate the abstractive summarization capabilities of models trained with distilled contexts. We compute the Rouge-L scores (Lin, 2004) between the generated summaries and ground-truth answers, following the same evaluation protocol
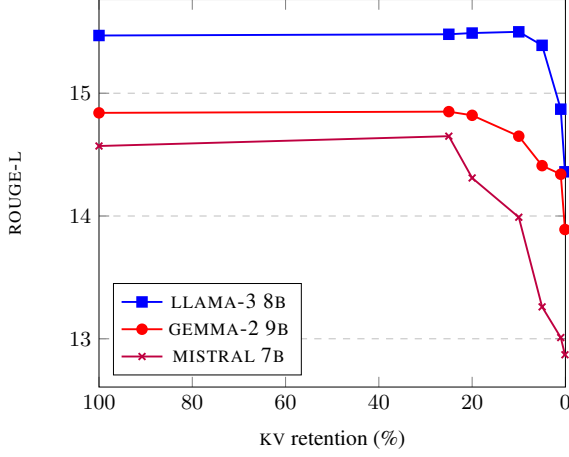
7

Figure 5: Rouge-L on SQuALITY

| KV retention | H$_2$A | H$_2$I | KVD | |
			Zero Shot	Finetune
100%	23.7	23.7	23.7	23.7
20%	22.8	20.6	22.3	**23.5**
10%	22.4	18.6	21.8	**23.3**
5%	21.9	18.5	21.1	**23.2**
1%	21.1	18.3	20.1	**22.8**

Table 3: ROUGE-L on GovReport summarization.

used by LLAMA-3 (AI@Meta, 2024).

Result Figure 5 show Rouge-L performance on SQuALITY. We observe that KV-DISTILL models perform as well or better than uncompressed models when retaining more that 20% of the KV cache. When retaining under 20%, we observe different performance falloff behaviors for different models; in particular, we observe that textscLlama-3 and GEMMA-2 have stable performance until 100x compression, at which point performance dips drastically. This difference in the behavior of the compression-performance trade-off could be attributed to the larger vocabulary sizes of LLAMA-3 and GEMMA-2, which allows the KL-loss to capture more fine-grained features of the output distribution during pretraining. These results demonstrate that KV-DISTILL can support very high compression ratios with minimal performance penalty on abstractive tasks.

5.5 Finetuned Long Context Summarization

Motivation GovReport is a long document summarization dataset that consists (Report, Summary) pairs written by government research agencies. In contrast to the evaluations on QuALITY and SQuALITY (which are performed in a zero-shot fashion using the best pretraining checkpoint), we perform additional finetuning distillation with Equation 6 on the GovReport training set before evaluation. As with SQuALITY, we use GovReport to assess the abstractive summarization ability of models equipped with KV-DISTILL .

Results Table 3 shows results for GovReport for both query-aware H$_2$A and query-independent H$_2$I paradigms, as well as KV-DISTILL prior to fine-tuning (zero-shot) and KV-DISTILL after finetuning

on LLAMA-3. We observe that KV-DISTILL and query-aware H$_2$ perform close to each other on this evaluation in the zero-shot setting, while KV-DISTILL outperforms H$_2$I at all compression ratios. However, upon finetuning, we observe a practical improvement in performance with KV-DISTILL , with little degradation from uncompressed performance across all compression rates. In particular, we note the improvement in performance is greater at more severe compression ratios, confirming the utility of KV-DISTILL in supporting ultra-high compression ratios.

6 Discussion and Conclusion

We develop a method to reduce the memory requirements of long-context conditioned LM generation. Our method sub-selects tokens from the KV cache, and applies a token-level KL-type loss between the output of the LM when conditioned on sub-selected tokens and when conditioned on the uncompressed cache. We evaluate our method on long-context extractive and abstractive tasks, and demonstrate improved performance over competing compression methods. We further demonstrate that continued training on domain-specific data can allow for use of compression ratios as high as 100x with negligible losses in performance.

As part of this work we release distilled checkpoints across various model language families. These artifacts allow efficient text generation conditioned on significantly larger inputs than before, with much lower memory burden, and support compression ratios as high as 1000x. We anticipate these artifacts will be of great practical benefit, enabling exciting new applications and research directions in language processing.

7 Limitations

The time-consuming and stochastic nature of distilling a model means that it cannot be guaranteed that the process will work well across all model families. Furthermore, we are unsure as to the root

8

cause of performance discrepancies between model architectures after distillation; this issue merits further research. Lastly, the 8k token context-capacity of LLAMA-3 limited many of our experiments, and is small by the standards of currently available language models; to address this, we will be releasing a KV-DISTILL LLAMA-3.1 model with a 128k token context capacity.

References

AI@Meta. 2024. Llama 3 model card.

Joshua Ainslie, Tao Lei, Michiel de Jong, Santiago Ontañón, Siddhartha Brahma, Yury Zemlyanskiy, David Uthus, Mandy Guo, James Lee-Thorp, Yi Tay, Yun-Hsuan Sung, and Sumit Sanghai. 2023. CoLT5: Faster Long-Range Transformers with Conditional Computation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Yukang Chen, Shaozuo Yu, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Long alpaca: Long-context instruction-following models. https://github.com/dvlab-research/LongLoRA.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2021. Rethinking Attention with Performers. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Tao Ge, Jing Hu, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context Autoencoder for Context Compression in a Large Language Model. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. LM-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008, Mexico City, Mexico. Association for Computational Linguistics.

Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Damjan Kalajdzievski. 2023. A rank stabilization scaling factor for fine-tuning with LoRA. *Preprint*, arXiv:2312.03732.

G Kamradt. 2023. Needle in a haystack–pressure testing llms.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

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.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics (TACL)*, 12:157–173.

Taiming Lu, Muhan Gao, Kuai Yu, Adam Byerly, and Daniel Khashabi. 2024. Insights into LLM long-context failures: When transformers know but don't tell. *Preprint*, arXiv:2406.14673.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to Compress Prompts with Gist Tokens. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

Guanghui Qin and Jason Eisner. 2021. Learning How to Ask: Querying LMs with Mixtures of Soft Prompts. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Guanghui Qin, Yukun Feng, and Benjamin Van Durme. 2022. The nlp task effectiveness of long-range transformers. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Guanghui Qin, Corby Rosset, Ethan C. Chau, Nikhil Rao, and Benjamin Van Durme. 2024. Dodo: Dynamic Contextual Compression for Decoder-only LMs. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*.

Guanghui Qin and Benjamin Van Durme. 2023. Nugget: Neural Agglomerative Embeddings of Text. In *Proceedings of International Conference on Machine Learning (ICML)*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. 2020. Compressive Transformers for Long-Range Sequence Modelling. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *Preprint*, arXiv:2110.08207.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language model with self generated instructions.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick Von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Y. Wu, M. N. Rabe, D. Hutchins, and C. Szegedy. 2022. Memorizing Transformers. In *Proceedings of International Conference on Learning Representations (ICLR)*.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

Zhanpeng Zeng, Cole Hawkins, Mingyi Hong, Aston Zhang, Nikolaos Pappas, Vikas Singh, and Shuai Zheng. 2023. VCC: Scaling Transformers to 128K Tokens or More by Prioritizing Important Tokens. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Proceedings of Conference on Neural Information Processing Systems (NeurIPS)*.

A Training & Evaluation Details

We train all KV-DISTILL models with the following parameters at bf16 precision on 8 NVIDIA A100s. Please see Table 4 for further details. The

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	5e-5
Batch Size	32
LoRA Rank	128
λ	0.6
η	6

Table 4: Hyperparameters for training

QuALITY, SQuALITY, GovReport, and SQuAD evaluations are performed on the test set, if public, else results are reported on the development set. To measure SQuAD accuracy, we generate up to 128 tokens, normalize the output by stripping punctuation, and check if the correct answer is contained in the generated answer. For SQuALITY and QuALITY, we follow the evaluation procedure of AI@Meta (2024). For GovReport, we prompt the model to summarize the report, and then greedily generate 630 tokens.

B Pretraining Objective Ablations

We assess the necessity of both the forward and reverse terms in the loss by evaluating SQuAD performance on multiple different pre-training losses with varying λ values in Equation 6. In Table 5 we observe that using either the pure forward or reverse divergences performs markedly worse than using a mixture of both. Furthermore, using solely the auto-encode + cross-entropy (used in ICAE and DODO), performs significantly worse than Equation 6, demonstrating the significant benefits that the weighted distillation objective provide. We also note that replacing the routing mechanism with a learnable embedding (added to important tokens) does not perform well.

Loss	SQuAD Acc. (%)
$\lambda = 1$	83.4%
$\lambda = 0.6$	86.0%
$\lambda = 0.4$	85.3%
$\lambda = 0$	82.7%
AE + LM	79.1%
NO ROUTING	67.4 %

Table 5: Effect of Pretraining Loss wrt LLAMA-3.

C Qualitative Analysis of Generated Text

In Figure 6 we present representative summaries of the short story "Galactic Ghost". LLAMA-3 was prompted to summarize the short story in a few sentences. As the KV retention ratio decreases, we observe an increase in statements that are unfaithful to the original context. Particularly, at extremely high compression ratios, the model begins to hallucinate details, but impressively still maintains understanding of the general plot of the story. Note the varying degrees of *semantic compression* in summaries generated under high compression ratios; qualitatively, we observe that models with severely compressed contexts generally understand the text, but resort to making vague statements.

(a) 100% KV retention

The story follows Willard, a space explorer who is left alone after his friend and space mate, Dobbin, dies. Willard is haunted by the memory of Dobbin's final words, "I saw the Ghost Ship," and becomes obsessed with finding the ship. He spends years searching for it, and eventually, he sees a strange, ghostly ship that seems to be heading towards him. As he gets closer, he realizes that it's not just a hallucination, but a real ship from Earth that's been traveling through space for centuries. In the end, Willard is taken aboard the ship, but learns that the ship and its crew are mere shells of their former selves, and that they are doomed to remain in space forever. Ultimately, Willard is forced to confront the reality that he'll never return to Earth.

(b) 2.5% KV retention

The story follows the journey of Willard, a space traveler who has been alone in space for years after the death of his friend and partner, Dobbin. Willard is struggling to come to terms with his isolation and the loss of his friend, and is haunted by the legend of the Ghost Ship, a mythical vessel that is said to roam the solar system, never to be seen again. One day, Willard spots a strange, rocket-shaped object approaching him, and as it draws closer, he realizes it is the Mary Lou. The ship is now a mere shadow of its former self, and Willard is invited to board it, where he is greeted by a rescue team who have been sent to retrieve him. As he adjusts to life on the rescue ship, Willard comes to terms with his past and the reality of his situation, and ultimately accepts that he will never return to Earth. The story concludes with Willard's poignant realization that he and the crew of the rescue ship are the Ghosts, doomed to roam the solar system forever.

(c) .25% KV retention

The story follows the journey of Willard, a space explorer who has been traveling through the galaxy for years, searching for a new home for humanity. He has lost his friend and companion, Dobbin, and is struggling to come to terms with his own mortality. One day, he discovers a faint light in the distance, which he believes to be a sign of life. As he approaches the light, he realizes it is a ghost ship, a ship that has been drifting through space for millions of years, its crew and passengers long dead. The ship is a relic of a bygone era, and Willard is forced to confront the reality that he will never be able to return to Earth, as the ship would pass through it, making it impossible to survive. In the end, Willard comes to accept his fate, realizing that he and the crew of the ghost ship are doomed to drift through space forever, a reminder of the transience of human existence.

Figure 6: LLAMA-3 was tasked with summarizing a 6k token short story at low KV retention rates. Inaccuracies in the summary are highlighted yellow, and were determined by hand.