Temporal Alignment of Time Sensitive Facts with Activation Engineering

Sanjay Govindan, Maurice Pagnucco, Yang Song
University of New South Wales, Sydney

Abstract

Large Language Models (LLMs) are trained on diverse and often conflicting knowledge spanning multiple domains and time periods. Some of this knowledge is only valid within specific temporal contexts, such as answering the question, "Who is the President of the United States in 2022?" Ensuring LLMs generate time-appropriate responses is crucial for maintaining relevance and accuracy. In this work we explore activation engineering as a method for temporally aligning LLMs to improve factual recall without any training. Activation engineering has predominantly been used to steer subjective and qualitative outcomes such as toxicity or behaviour. Our research is one of few that uncovers the bounds of activation engineering on objective outcomes. We explore an activation engineering technique to anchor LLaMA 2, LLaMA 3.1, Qwen 2 and Gemma 2 to specific points in time and examine the effects of varying injection layers and prompting strategies. Our experiments demonstrate up to a 44% and 16% improvement in relative and explicit prompting respectively, achieving comparable performance to the fine-tuning method proposed by Zhao et al. (2024). Notably, for LLaMA 2 and LLaMA 3.1 our approach achieves similar results to the fine-tuning baseline while being significantly more computationally efficient and requiring no pre-aligned datasets.

1 Introduction

Large Language Models (LLMs) encode and train on a large corpus of information that the end user can query (Petroni et al., 2019; Cohen et al., 2024; Vaswani et al., 2017). Their training sets can span a large timeframe leading to overlapping and conflicting answers for time sensitive queries such as "Who is the President of the United States of America?". Questions like these can have different answers throughout time with the correct answer (Joe Biden in 2024 and Donald Trump in 2025) being temporally sensitive; relevant to the time it is asked (Ge et al., 2024; Dhingra et al., 2022; Luu et al., 2022).

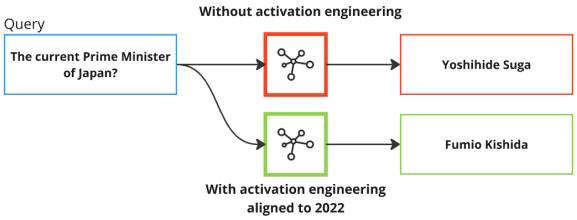

Figure 1: When asked "Who is the current Prime Minister of Japan" LLaMA2-7b outputs Yoshihide Suga. Applying activation engineering as temporal alignment assistance for the year 2022 produces the correct set of facts for LLaMA-7b.

Temporally sensitive questions require LLMs to correctly understand the time they are answering for. Without temporal alignment LLMs are recalling facts based on training distributions leading to a chaotic sense of time for factual recall (Zhao et al., 2024). This leads to errors such as GPT2-XL recalling that the current Prime Minister of Australia is Malcolm Turnbull. This is at odds with the knowledge the LLM has on hand. The model, if prompted carefully can recall that Scott Morrison is the serving Prime Minister of Australia in 2022, a more recent and temporally relevant answer. The confusion about whom the current Prime Minister of Australia demonstrates how temporal misalignment can lead to erroneous factual recall (Luu et al., 2022).

Conflicting answers throughout time can lead to a range of recall errors. To overcome these errors many methods aim to overwrite the subject, relationship, object (SRO) facts held within an LLM (Geva et al., 2023; Cohen et al., 2024; Petroni et al., 2019; Yu et al., 2023; Dai et al., 2022); for example, overwriting the object associated with the President of the United States of America and altering the probability of the output tokens to preference Don-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 7640–7657

ald Trump over Joe Biden. These methods fall under the categories of continual learning (Abel et al., 2023; Ke et al., 2023; Jin et al., 2022), knowledge editing (De Cao et al., 2021; Yu et al., 2024; Hartvigsen et al., 2023; Meng et al., 2022, 2023), and retrieval augmented generation (RAG) (Lewis et al., 2020), which all strive to provide correct information to the end user.

However, overwriting SRO fact sets ignores the opportunity to realign LLMs temporally to recall the correct facts already known by the model. Building on the temporal alignment work of (Zhao et al., 2024) we explore the capability of injecting vectors into the residual stream during inference (activation engineering) as a technique to *temporally align models within their existing knowledge cut-off timeframe and correct for relative temporal statements* such as "Who is the current Prime Minister of Japan?". Figure 1 demonstrates that when LLaMA2-7b (which has a knowledge cutoff date of September 2022) is asked this question the preferred response is Yoshida Suga (who served as Prime Minister between 2020 and 2021). However, when temporally aligned to the year 2022, using activation engineering the preferred response is Fumio Kishida; who was the current serving Prime Minister of Japan in 2022. The distinction within this scenario is that we are aligning to knowledge the model already possesses. While Zhao et al. (2024) uses fine-tuning to temporally align, we explore the effectiveness of *activation engineering* (AE); reducing the computational requirements to temporally align, increases the flexibility and responsiveness for end users, and requires less pre-aligned data to reference and train.

In this work we hypothesise that AE is a more efficient method to temporally align models compared to fine-tuning, reducing the amount of training and data required, whilst providing similar outcomes to temporal alignment via fine-tuning. We experiment with the activation methods of Turner et al. (2023); Rimsky et al. (2024) because of their effectiveness in reducing toxicity, their ease of integration into LLMs and efficiency at inference. While Turner et al. (2023) and Rimsky et al. (2024)'s research focuses on qualitative aspects of LLMs such as toxicity and topic fixation, our research on the other hand, aims to understand how AE can affect time-sensitive factual recall, which to the best of our knowledge is a new research perspective. Furthermore, while Turner et al. (2023) and Rimsky et al. (2024) look at single layer activation engineering, we explore the capabilities of multi-layer vector injections for aligning time for factual recall.

We apply AE over two datasets, Head of Governments (HOG) and Temporal Alignment Question Answer (Taqa) (Zhao et al., 2024). The HOG dataset was created as part of this research to be a small and domain specific dataset. The Taqa dataset (Zhao et al., 2024) on the other hand was selected to test the effectiveness of AE on a larger, more diverse dataset; exploring the generality of AE on temporal alignment and providing a benchmark to compare against.

We run sweep tests throughout LLaMA2-7b, 13b and 70b models varying the layers of activation and phrases injected into the residual stream. Next, we test the effect of AE on a single layer and then compound the effect by applying AE to multiple layers. We then compare the results of AE against explicit prompting (e.g., In 2022 the President of the United States of America is?), relative prompting (e.g., The current President of the United States of America is?) and fine-tuning; following the methodology of Zhao et al. (2024). We take our findings from the smaller HOG and Taqa-1000 experiments and scale them to the entire Taqa-9000 testing set where we compare our summary Taqa results to Zhao et al. (2024) demonstrating a similar alignment to a specific year, but with less computational overhead.

2 Related Studies

2.1 Temporal alignment

Temporal alignment (Zhao et al., 2024) is a relatively new field and distinct from temporal reasoning which aims to understand and influence how time is logically treated by LLMs; such as what date is it from 8 months from now? (Tan et al., 2023; Yuan et al., 2024). Temporal alignment on the other hand aims to influence the recall of facts so that they are referenced from a specific point in time providing a time sensitive contextually correct answer.

Zhao et al. (2024) is one of the few papers we could find that explores temporal alignment of LLMs. In their research they develop a benchmark dataset Temporal Alignment Question Answer (Taqa) and explored a fine-tuning method for aligning LLMs to a specific year using implicit factual statements. They demonstrate an improvement in alignment and recall of facts from more

recent periods compared with explicit and relative prompting. Other methods such as Mend (Mitchell et al., 2022), Memit (Meng et al., 2023) and Rome (Meng et al., 2022) are focussed on knowledge editing (Yin et al., 2024; Ge et al., 2024; Dong et al., 2022; Li et al., 2024), correcting knowledge via hypernetworks or main network editing. These methods overlook the opportunity to correct the fact with minimal intervention via temporal alignment, which can also be used as a tool to minimise the amount of overwriting required in the first place.

2.2 Activation Engineering

We are influenced by the AE approach pioneered by Rimsky et al. (2024) and Turner et al. (2023) as their method demonstrates efficiency in steering models towards a desired outcome. Their methods are different to prior AE techniques (Dathathri et al., 2019; Hernandez et al., 2024) as they use feed forward mechanisms without any pre-training or major model alterations. They demonstrate that AE is effective in reducing toxicity and creating topical fixations producing the desired outcome with minimal intervention.

The AddAct (Turner et al., 2023) method explores single contrasting pairs to assist in the preference of topic and LLM perspectives, without any significant data gathering or processing. Rimsky et al. (2024) on the other hand uses a sizable set of crafted contrasting statements which are then added back into the residual stream. To the best of our knowledge AE has not been explored to correct factual information or align time within LLMs.

3 Datasets

A variety of datasets have been developed to examine the properties and methods pertaining to LLMs. Recall, biases and reasoning (Thorne et al., 2018; Zhong et al., 2023) are a few categories of datasets constructed to address research in these areas. These datasets typically look at Subject Relationship Object (SRO) mappings, ignoring time as a factor for those relationships. For temporal research our dataset requires a record of changes over time for an SRO fact set, expanding the dataset to one that maps SROT: Subject, Relationship, Object and Time. Specifically, we require a dataset that contains consistent time-exclusive answers, where the answer A is only valid between $t \in [t_s, t_e]$, and A has a consistent set of answers through time.

To the best of our knowledge, there are only two datasets that meet this requirement (Herel et al., 2024; Zhao et al., 2024). Other datasets such as Atoke (Yin et al., 2024), MQuake (Thorne et al., 2018), ChronoEdit (Ge et al., 2024) contain question-answer sets through time but do not continuously record the change of answers over time. These datasets are focussed on knowledge editing for time sensitive questions typically exploring one hop knowledge editing effects. For our experiments we have chosen to first create a smaller Head of Government (HOG) dataset to provide an easier measure and more efficient experimentation feedback loop. Then, to validate our findings are domain agnostic, we test against Temporal Alignment Question Answer (Taqa) (Zhao et al., 2024) which contains a much wider knowledge domain. We chose Taqa over Herel et al. (2024) as Taqa is originally benchmarked against a fine-tuning alignment approach which provides a fair baseline for us to compare against.

3.1 Head of Governments (HOG)

We augment an existing ideologies (Herre, 2022) dataset to test AE's effect on large periods of time with relatively consistent changes. The dataset contains over 175 countries, recording the heads of government between the years 1945 and 2020 inclusive. This set has 175 temporally relevant questions (Who is the current head of government x?), and up to 13,125 explicit temporal questions (e.g., In the year Y, who is head of government for X?).

3.2 Temporal Alignment Question Answer (Taqa) Dataset

To further validate our temporal alignment experiments we utilise the Taqa dataset (Zhao et al., 2024). The Taqa dataset test set contains 9000 temporally relevant questions (e.g., who is the most recent winner of the Stanley Cup?) and up to 113,000 explicit temporal questions (In the year 2010, who was the most recent winner of the nationals skating championship?).

Whilst Taqa presents a diverse set of question answer pairs, similarities between answers over time and a low recall for LLaMA2 models makes this dataset challenging to work with. A range of questions have very similar answers with just a single token change between the years. These include questions such as, "What edition of the Producers Guild of America Awards was last held?", whereby the answer is incremented by one in most years. Similarly, we note a series of false positives asso-

ciated with using F1 token evaluation leading to computational inefficiencies and misleading evaluation. Questions such as "When was the latest Awit Awards ceremony held?", where the correct answer is a year, such as 2022, leading to any reasonable answer within the 21st century having an F1 score of approximately 0.5. Furthermore, some of these questions could be considered challenging for smaller models such as LLaMA2 7b and 13b which might not have been trained on an extensive body of knowledge.

To improve development efficiency and limit the effect of false positives provided from partial answers we filter for questions that have an F1 Score above 0.5 when answered relatively by LLaMA2-7b. We capture questions LLaMA2-7b, 13b and 70b can answer confidently, creating a more sensitive testing dataset and minimise the testing feedback loop. The Taqa dataset reduces from 9000 to 2930 question answer pairs. We further reduce this to the first 1000 question answer pairs within the 2930 filtered set. This dataset is denoted as Taqa-1000 whilst the full Taqa dataset is denoted as Taqa-9000.

4 Methods

As outlined in Algorithm 1 and Figure 2, during the prediction of an answer for a temporally sensitive question p_u such as, "Who is the current President of the United States of America?", we inject a pre-defined steering vector ae that represents the specific year in to layers l of the model.

These steering vectors are created from running temporal phrases $\{p,\}$ such as a year number through the model and extracting the activation vectors h before a selected layer. Using h, we apply a positive or negative coefficient c to phrase's vector increasing or reducing the influence of the phrase, and producing h_a. Negative coefficients are used to reduce the influence of a phrase, whilst positive coefficients are used to increase the influence of a phrase upon the output of a model.

Having developed a set of vectors h that have been multiplied by a coefficient, we sum the set of phrase vectors together to produce a single vector ae that encapsulates a time period we aim to align to. We add the alignment vector ae to the residual stream of query prompt p_u at the same layer l the original activation vectors h were extracted from. Ultimately, this small vector "nudge" can change the probability of output tokens, steering the model to recall information from a specific period of time.

Algorithm 1 Activation engineering

Require: $\{p,\}$ = steering prompts list
Require: p_u = user prompt
Require: $\{l,\}$ = target layer list
Require: $\{c,\}$ = coefficient list
Require: a = alignment position (front)
Require: M = pre-trained language model
Ensure: S = steered output

$mtl = max(len(p)$ for p in $\{p, ...\})$ max token length
for each l in $(\{l, ...\})$ **do**
  $ae = \{\}$ empty activation vector
  for each p, c in $(\{p,\}, \{c,\})$ **do**
    $(p) \leftarrow$ pad right to match mtl
    $h \leftarrow M.forward(p).activations[l]$
    $h_a \leftarrow h \times c$
    $ae \leftarrow ae + h_a$
  end for
  $q \leftarrow M.forward(p_u).activations[l]$
  $S \leftarrow M.continue\_forward(ae + q@a)$
end for

Injection alterations Our method changes how to inject these vectors into the model, creating a list of layers to inject into $\{l, ...\}$. We perform two types of injections. Where prior studies (Rimsky et al., 2024; Turner et al., 2023) have focussed on single layer activation engineering we explore the capabilities of multi-layer activation engineering on factual recall. We hypothesise that small nudges throughout a model can provide a more stable outcome for tasks related to factual recall as the vectors are applied to multiple layers providing a more consistent steering of the model. **Single-layer:** Applying the vector to a single layer only. **Multi-layer:** Applying vectors additively from layer 4 onwards. e.g., applying vectors to layer 4 and 5, or applying vectors to layer 4, 5 and 6.

Temporal prompts We alter the prompts {p,} required to create the steering vectors. Specifically we alter the prompt in three ways. **Year only:** Only the year preferenced for alignment (e.g., 2010). We test year only as a way to efficiently temporally align the model, providing a single year number to see how a basic injection can align the output. A coefficient c of 4 is applied for single layer and 1 for multi layer. **Context phrase:** The preference year with context about the number (e.g., the year is 2010). Given the prompting experiments conducted by (Zhao et al., 2024), regarding implicit

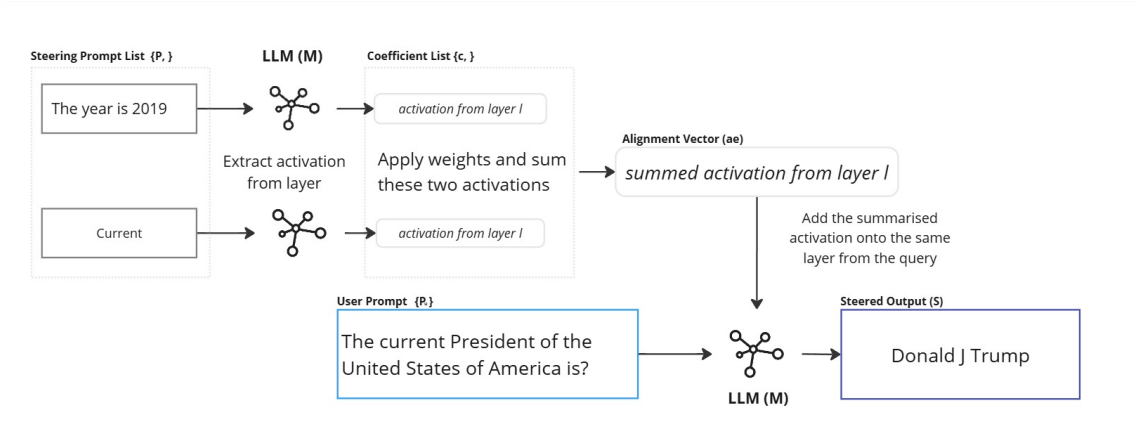

Figure 2: Activation Engineering in LLMs. A set of vectors is extracted from layer l, multiplied by a coefficient and added together. Finally, this vector is added into a temporal question to temporally align the model.

and explicit alignment prompts we broaden our testing to include context about the numbers injected into the residual stream adding "The year is". A c of 4 is applied for single layer and 1 for multi layer. **Contrasting pair:** A preference year as a positive, and "recent" as a negative. e.g., 2021 having a c of 4, and "recent" having a c of -2 for a single layer and c of 2 and -1 for multi layer.

5 Experiments

5.1 Models and Prompting

We focus on LLaMA2 7b, 13b and 70b, but provide generalisation experiments across LLaMA3.1-8b, Qwen2-2b and Gemma2-2b. With regard to setting up the prompts we mimic the setup used by Zhao et al. (2024). When adding these question answer examples for *relative benchmark* tests we keep the question answer examples generic and remove any time from the prompting. For *explicit benchmark* tests we use prompts that reference the year we wish to align to prefixing our question answer examples with the year of interest. Appendix G demonstrates the prompts used.

5.2 Evaluation Criteria

Similar to (Zhao et al., 2024) and the QA methodologies of (Kwiatkowski et al., 2019) we use an averaged F1 and F1 max score for evaluation. An average F1 score demonstrates the effectiveness of aligning on a single year whilst the F1 max score monitors for a significant loss of information over the other years. If the F1 max score decreases drastically compared to benchmark tests, we can assume the method is having an overall negative effect on the model's output. The F1 max score is calculated between 1945 and 2020 and 2000-2023 for the HOG and Taqa datasets respectively.

5.3 Baselines

We first evaluate three baseline approaches, relative prompting, explicit prompting and fine-tuning on the LLaMA models, without involving AE. Fine-tuning mimics the alignment technique and closely follows the hyperparameters of Zhao et al. (2024) and aligns the model to a specific year using implicit factual statements from the Taqa training set. Whilst we aim to mimic the specific techniques of Zhao et al. (2024), due to computational limitations we are only able to apply full parameter fine-tuning to LLaMA2-7b and 13b models. For LLaMA2-70b we apply PEFT LoRA (Hu et al., 2021) fine-tuning and reduced the batch size of all training to 8.

Figure 3, focuses on LLaMA2-7b's relative and explicit prompting illustrate that the alignment bias of relative and explicit prompts. Earlier years generally produce the worst relative and explicit F1 scores, whilst more recent years exhibit the best F1 scores for the HOG dataset. This isn't the case for the Taqa-9000 dataset which demonstrates 2015 being an easier year to align with explicit prompting. These findings are consistent across 13b and 70b models. The Taqa-9000 dataset illustrates lower performance for explicit prompting alignment for years 2020 to 2022, compared to the gains experienced from explicit prompting in the year 2015. Furthermore, the marginal gains for 13b and 70b compared to 7b between relative and explicit prompting (Table 1) most likely stems from the filter conditions applied to Taqa-1000 which favoured question answers pairs that LLaMA2-7b could answer with a high F1 score. Using Taqa-9000 (Table

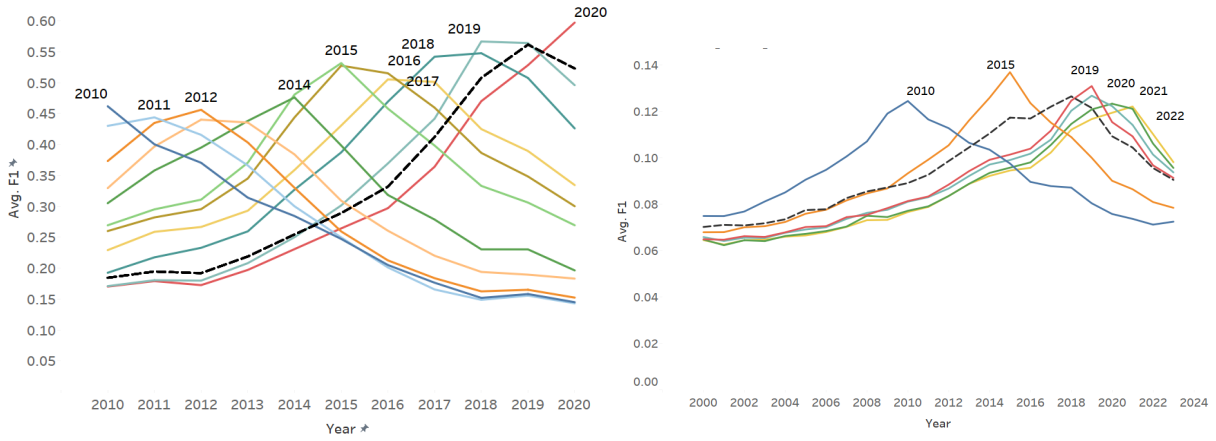

Figure 3: Left (HOG Dataset), right (Taqa-9000) benchmarking F1 scores for LLaMA2-7b for both relative (checked line) and explicit prompts.

2) we note larger models produce better F1 Scores and the gain between explicit and relative scores scales with the size of the model.

5.4 Results

Table 1 demonstrates our AE method improves results by up to +10% points compared to relative prompting and +5% points compared to explicit prompting. For both the HOG (Appendix B.1) and Taqa-1000 datasets, we note that multi-layer with a contrasting pair prompt is an effective technique for aligning LLaMA2 13b and 70b models whilst the 7b model benefits from a mixture of alignment techniques. The F1 max scores for all test cases are similar to the relative F1 max scores (Appendix B.1, B.2 and B.3). This suggests that the loss of information in other time periods is made up by the correction of information from temporal alignment via AE. This is in contrast to the explicit prompting which has a decrease in F1 max score compared to the relative prompt. Overall, Table 2 demonstrates that AE and fine-tuning produce very similar scores, with differences of up to $\pm 1\%$ between the methods. When examining LLaMA2-70b, we note that AE can outperform the PEFT LoRA fine-tuning method and explicit prompting. If PEFT LoRA can achieve a similar score to AE, we speculate that full parameter fine-tuning would surpass the best AE F1 scores.

Time and computational efficiencies Fine-tuning LLaMA2 models (7b, 13b, 70b) for temporal alignment demands significant time and computational resources, whereas AE achieves similar results with lower GPU and time requirements. Generating steering vectors for alignment prompts takes only 0.05 seconds, enabling AE to quickly align models. In contrast, fine-tuning requires extensive data processing and training time (10–15 minutes for 7b and 13b, 30 minutes for 70b with PEFT LoRA); for 2 epochs, using Zhao et al. (2024) hyperparameters. Fine-tuning also demands substantial GPU power, with configurations ranging from 1xA100-80G (7b), 2xA100-80G (13b) and 3xA100-80G (70b, using PEFT LoRA). AE requires only inference-capable GPUs, with 7b running on a V100-32G, 13b on 1xA100-80G, and 70b on 2xA100-80G. Overall, AE offers a more efficient alternative to fine-tuning for temporal alignment.

Layer ablation To limit the degrees of freedom within our problem space we conducted an ablation study applying AE to individual layers throughout 7b, 13b and 70b parameter models. Specifically we injected the "year only" vector, testing individual layers between 4-29 for 7b and 4-39 for 13b and layers 4-29 for 70b. Layers earlier than 4 or later than 29 for 7b and 39 for 13b were deemed inconsequential to test; too low, and the model is still encoding the input, too high and the model is only refining the probability of tokens (Geva et al., 2023).

Figure 4 highlights that single layer injection only works for lower layers (4-14 for 7b & 13b, and 9-24 for 70b), with higher layers (14-29 for 7b & 13b, 24 and beyond for 70b) seemingly ignoring the activation vector injection. This indicates that the influence of steering vectors for temporal alignment could be capped to first third of LLaMA2 models. The application of AE at higher layers leads to a reversion to the original relative prompt preference year (Figure 4). Lower layers for all

	LLaMA 7b		LLaMA 13b		LLaMA 70b	
	2021	**2022**	**2021**	**2022**	**2021**	**2022**
Benchmark						
Relative Prompting	25.0	22.3	20.9	18.5	23.9	20.7
Explicit Prompting	26.5	23.9	23.6	23.1	29.7	27.3
Fine-tuning	27.2	24.4	**28.7**	**24.5**	32.9*	29.4*
Single layer						
Year only	28.4 (L6)	25.9 (L8)	24.9 (L6)	23.8 (L10)	31.0 (L16)	29.4 (L16)
Context and year	27.9 (L4)	26.0 (L6)	24.5 (L6)	22.9 (L12	30.3 (L4)	28.2 (L4))
Contrasting Pair	28.6 (L6)	26.4 (L6)	21.6 (L6)	23.3 (L8)	31.2 (L16)	29.7 (L16)
Multi-layer						
Year only	28.7 (L4-7)	**28.2 (L4-7)**	25.0 (L4-10)	24.2 (L4-10)	33.8 (L4-20)	31.1 (L4-20)
Context and year	28.6 (L4-8)	26.0 (L4-9)	24.8 (L4-10)	23.5 (L4-13)	33.3 (L4-20)	31.4 (L4-17)
Contrasting Pair	**29.0 (L4-10)**	26.3 (L4-5)	**25.7 (L4-11)**	**24.4 (L4-11)**	**34.5 (L4-20)**	**31.6 (L4-20)**

Table 1: F1 scores for single layer and multi-layer experiments for Taqa-1000. 'L' denotes which layer(s) the optimal score came from, and * denotes that the fine-tuning method was PEFT LoRA.

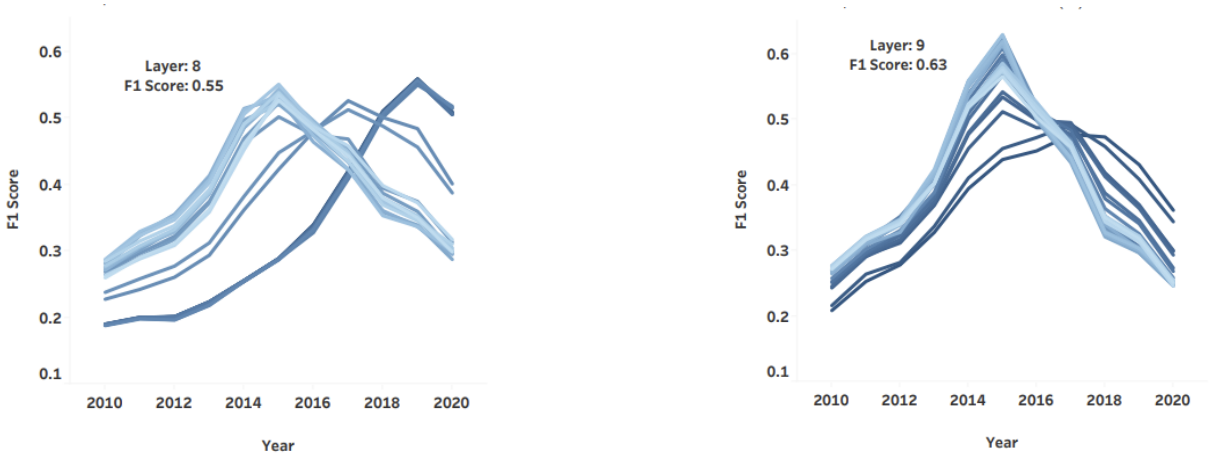

Figure 4: Left (LLaMA2-7b), right (LLaMA2-70b) single layer AE effect on the HOG dataset, using "year only" prompting aligning to the year 2015. Layers 4-29 are present. Lighter colours denote lower layers (4-11), and darker colours denote higher layers (12-29). The labels denote the best result and layer.

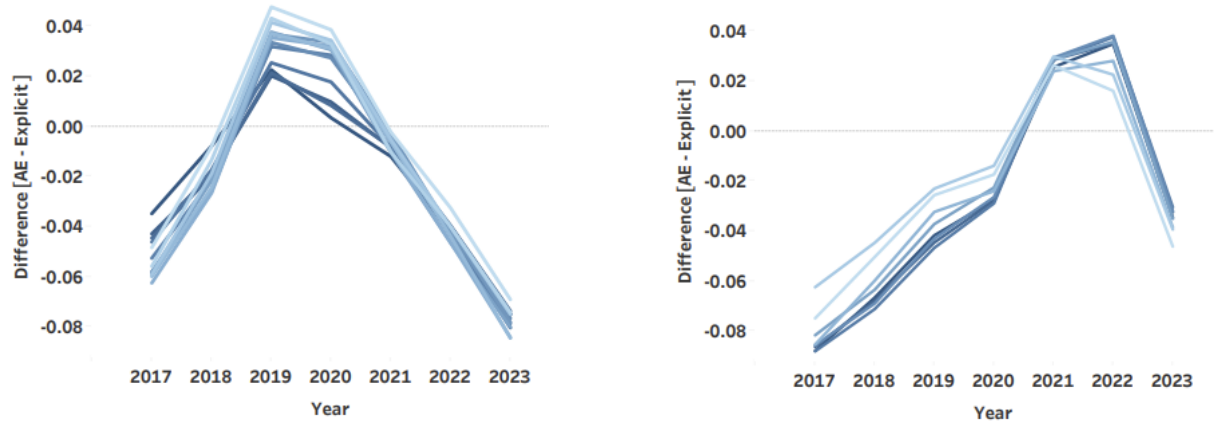

Figure 5: Left (Single layer), right (Multi layer) alignment to 2022 with AE applied to LLaMA2-70b. AE is applied to different layers. The Y-axis is the difference in F1 score between our AE method and explicit prompting. Darker colours denote higher layers. For multi-layer approach, layer 4 is the first layer, and the colour denotes the last layer included. The maximum layer count in both graphs is 26.

	LLaMA 7b		LLaMA 13b		LLaMA 70b	
	2021	2022	2021	2022	2021	2022
Benchmark						
Relative Prompting	10.5	9.6	10.7	9.8	14.1	13.0
Explicit Prompting	12.2	11.0	12.2	11.7	16.7	16.0
Fine-tuning	12.7	12.0	14.0	13.7	19.3*	17.7*
Multi-layer						
Contrasting Pair	13.1	12.3	13.6	12.9	20.0	19.1

Table 2: F1 scores for multi-layer experiments for Taqa-9000. * indicates PEFT LoRA fine-tuning.

models provide better steering as evidenced in the results of sweep tests from aligned years 2010 to 2020 (Appendix A). The loss of influence exhibited by AE in higher layers is most likely due to a change in layer behaviour (Geva et al., 2023); early layers attend to the inputs creating semantic representations. The mid-layers attend to those semantic representations to extract relevant information. Higher layers are tasked with refining the prediction for the next token. For LLaMA2-70b, the application of AE on layers below layer 9 produces results that are worse than AE application to later layers such as 20 and 21. This is most likely due to the scale of the model, which proportionally has more layers dedicated to encoding user inputs (Geva et al., 2023).

Multi layer stability The sweep profiles for single layer AE (Figure 4) demonstrate that sensitivity is only a concern for later layers. From layers 4-12 there is a reduction of -3 from the F1 score of the optimal layer. Beyond layer 12 we note a significant drop in the F1 score for the preferenced year 2015. In contrast, Figure 5 highlights how latter layers using a multi layer strategy produce some of the best F1 scores. The multi-layer strategy seemingly reduces the risk of missing the optimal singular layer to influence and suggests that the compounding effect of smaller activations on the residual stream can increase the overall F1 score.

Isolated influence We test the effect of AE on both numerical (Cobbe et al., 2021) and time invariant datasets (Joshi et al., 2017) using LlaMA2-13b. For the invariant question sets we see an increase in F1 scores after applying AE. Numerical scores have a small drop in performance for exact answers, however overall AE has minor influence on answering non-temporal questions. Appendix E

Generalisation We explore the generalisation of this AE methods across a range of models. We note that this AE technique is effective for LLaMA2 and LLaMA3.1 models however produces suboptimal outcome for Gemma2-2b and Qwen2-7b (Appendix C). For Gemma 2 and Qwen2, AE approaches the explicit score and does not surpass fine-tuning. Applying AE on the instruct variants of Gemma2 and Qwen2, AE outperforms the explicit prompt but doesn't outperform fine-tuning suggesting AE can override generic answers ("this information is not available"), encouraging the model to generate an answer. Whilst the results for these models are suboptimal we note that the multilayer approach for AE is still the most effective method for temporal alignment. We speculate that difference in architectures, and potentially training data between these models is a key factor in the effectiveness of AE. Noting the qualitative studies (Rimsky et al., 2024; Turner et al., 2023) applying AE can have a generalisable effect using more complex activation vectors in latter layers, we speculate that temporal alignment could potentially be generalised to other models with more complex activation vectors.

Activation Engineering assisting fine-tuning With the surprise reduction in performance across Gemma 2 and Qwen 2, we explored the idea of AE as an assistant to fine-tuning. We speculated that the answers produced from the Gemma 2 and Qwen 2 base models were missing the concise answers that would lead to an improved F1 scores. Applying AE to the temporally fine-tuned Gemma 2 and Qwen 2 models, we note that AE outperforms relative scores, but is unable to exceed explicit scores of the temporally fine-tuned model. Appendix F.

6 Conclusion

Temporal alignment can improve the factual recall accuracy of models but can currently only be achieved through fine-tuning, a computational and time intensive process. We have demonstrated similar results to the fine-tuning process using activation engineering (AE) on LLaMA2-7b, 13b and 70b models. AE can produce similar results to fine-tuning with less data preparation and less upfront computation and time requirements. Our ablation studies have shown that AE effects a model's perception of time through specific tweaks in the residual stream of lower layers in an LLM. Generalisation of AE to LLaMA3.1, Qwen2 and Gemma2 models has shown mixed results, with AE being effective for LLaMA3.1. These results could be the result of differences in architecture or training data.

7 Limitations

In this study, AE has been applied to isolated LLMs which have no access to external information. The information produced from our tests are a result of the model's pre-training. For LLaMA2-7b, 13b and 70b these models have a knowledge cut off date of September 2022 limiting our AE experiments to this timeframe. This study has not investigated the effect of AE upon external information integration systems into LLMs, such as RAG and knowledge editing with hyper networks.

Generalisation of AE to other models Gemma2, Qwen2, and LLaMA3.1 demonstrates mixed results that could be attributed to differences in architecture or training data; as these new models have been trained on more recent data exhibit strong alignment to recent information. Attention architectural differences between the models could be a factor in how AE is integrated into the residual stream and feed through layers. Running sweeps test (of both layers and coefficients) for Gemma2 and Qwen2 produces muted results leading us to believe that more complex representations feed into higher layers maybe be required to produces improved outcomes. We however did not explore this avenue in this study.

In addition, we used the hyperparameters defined by Zhao et al. (2024) when developing our fine-tuning baseline. We did not optimise these parameters for any difference in our fine-tuning dataset that may arise. Furthermore, this study was limited in its use of larger GPUs to fine tune LLaMA-70b. Our results for LLaMA2-70b fine-tuning was derived from PEFT LoRA fine-tuning which we speculate produced a suboptimal output compared to full parameter fine-tuning.

References

David Abel, Andre Barreto, Benjamin Van Roy, Doina Precup, Hado P van Hasselt, and Satinder Singh. 2023. A definition of continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 50377–50407. Curran Associates, Inc.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv [cs.CL]*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiou Ge, Ali Mousavi, Edouard Grave, Armand Joulin, Kun Qian, Benjamin Han, Mostafa Arefiyan, and Yunyao Li. 2024. Time sensitive knowledge editing through efficient finetuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 583–593, Bangkok, Thailand. Association for Computational Linguistics.

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

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*, volume 36, pages 47934–47959. Curran Associates, Inc.

David Herel, Vojtech Bartek, and Tomas Mikolov. 2024. Time awareness in large language models: Benchmarking fact recall across time. *arXiv [cs.CL]*.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2024. Inspecting and editing knowledge representations in language models. In *First Conference on Language Modeling*.

Bastian Herre. 2022. Identifying ideologues: A global dataset on political leaders, 1945–2020. *Br. J. Polit. Sci.*, 53(2):1–9.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4764–4780, Seattle, United States. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pretraining of language models. In *The Eleventh International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Trans. Assoc. Comput. Linguist.*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-Tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. PMET: Precise model editing in a transformer. *Proc. Conf. AAAI Artif. Intell.*, 38(17):18564–18572.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2022. Time waits for no one! analysis and challenges of temporal misalignment. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5944–5958, Seattle, United States. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass editing memory in a transformer. *The Eleventh International Conference on Learning Representations (ICLR)*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. 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 809–819.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv [cs.CL]*.

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

Xunjian Yin, Jin Jiang, Liming Yang, and Xiaojun Wan. 2024. History matters: Temporal knowledge editing in large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19413–19421.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. MELO: Enhancing model editing with neuron-indexed dynamic LoRA. *Proc. Conf. AAAI Artif. Intell.*, 38(17):19449–19457.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959, Singapore. Association for Computational Linguistics.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 1963–1974, New York, NY, USA. ACM.

Bowen Zhao, Zander Brumbaugh, Yizhong Wang, Hannaneh Hajishirzi, and Noah Smith. 2024. Set the clock: Temporal alignment of pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15015–15040, Bangkok, Thailand. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

Supplementary material

A Different alignment years

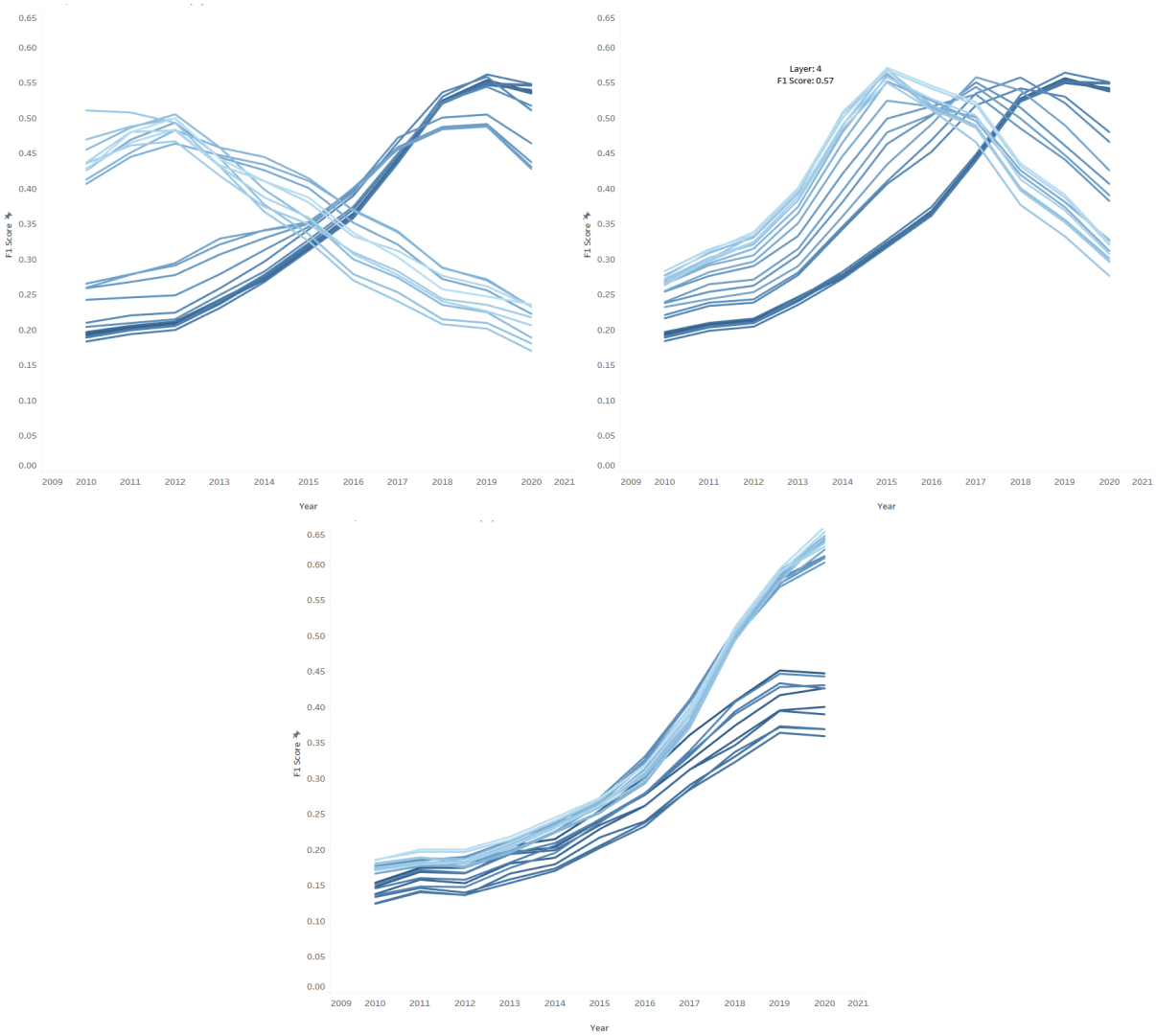

Figure 6: Hog dataset using LLaMA2-7b aligned to different years, left (2010), middle (2015) and right (2020).Injecting into lower (lighter coloured) layers proves easier to steer the model towards a preference year. Attempting to inject activation vectors into higher layers tends to revrt the model to it's original preference year as defined by the relative prompt.

B Results

B.1 Hog Results

	LLaMA 7b			LLaMA 13b			LLaMA 70b		
	2010	**2015**	**2020**	**2010**	**2015**	**2020**	**2010**	**2015**	**2020**
Benchmark									
Relative Prompting	18.4	28.9	52.3	18.8	32.1	55.1	20.0	33.7	43.4
Explicit Prompting	46.2	53.2	59.7	45.3	53.2	60.4	55.6	58.9	68.3
Single layer									
Year only	50.0 (L8)	54.4 (L10)	58.9 (L4)	51.1 (L8)	57.1 (L4)	65.2 (L4)	63.0 (L13)	62.6 (L12)	65.5 (L8)
Context and year	**52.6 (L13)**	**55.7 (L4)**	57.1 (L4)	55.9 (L10)	59.3 (L12)	64.8 (L10)	63.0 (L9)	62.6 (L12)	64.6 (L13)
Year and recent	**52.6 (L10)**	55.1 (L14)	59.3 (L12)	55.6 (L10)	59.3 (L10)	66.3 (L4)	61.9 (L10)	62.6 (L14)	64.6 (L8)
Multi-layer									
Year only	52.4 (L4-11)	53.8 (L4-10)	59.1 (L4-6)	53.1 L(4-12)	59.3 (L4-11)	66.7 (L4-8) 62.6	(L4-29)	62.2 (L4-23)	68.7 (L4-20)
Context and year	43.1 (L4-5)	52.7 (L4-8)	56.1 (L4-6)	52.9 (L4-10)	58.1 (L4-9)	64.8 (L4-5)	63.4 (L4-20)	62.4 (L4-20)	65.5 (L4-29)
year and recent	49.0 (L4-5)	54.9 (L4-5)	60.6 (L4-5)	56.1 (L4-11)	61.4 (L4-13)	67.9 (L4-5)	63.1 (L4-14)	63.3 (L4-11)	69.4 (L4-29)

Table 3: Average F1 score for all HOG Results.

	LLaMA 7b			**LLaMA 13b**			**LLaMA 70b**		
	2010	**2015**	**2020**	**2010**	**2015**	**2020**	**2010**	**2015**	**2020**
Benchmark									
Relative Prompting	70.2	70.2	70.2	72.9	72.9	72.9	71.4	71.4	71.4
Explicit Prompting	69.5	75.4	71.2	71.4	73.6	71.3	75.5	75.2	74.8
Single Sweep									
Year only	74.3	76.4	73.0	74.4	77.3	74.9	75.5	74.9	74.7
Context and year	73.0	76.6	72.2	74.2	74.7	74.1	77.0	75.2	73.5
Year - recent	75.6	74.7	72.4	74.6	75.3	75.9	76.5	75.5	74.0
Compounding									
Year only	73.5	75.4	73.2	75.9	77.0	76.0	76.8	75.9	74.8
Context and year	72.1	74.8	71.4	73.8	72.4	72.4	76.3	75.9	74.8
Year - recent	73.7	74.7	72.1	74.6	75.6	76.0	77.5	77.8	74.4

Table 4: F1 max score for HOG dataset.

B.2 Taqa-1000 results

	LLaMA 7b			LLaMA 13b			LLaMA 70b		
	2020	**2021**	**2022**	**2020**	**2021**	**2022**	**2020**	**2021**	**2022**
Benchmark									
Relative Prompting	26.0	25.0	22.3	23.3	20.9	18.5	26.1	23.9	20.7
Explicit Prompting	27.2	26.5	23.9	26.0	23.6	23.1	30.2	29.7	27.3
Fine-Tuning	28.0	27.2	24.4	28.3	27.4	24.5	29.6*	32.9*	29.4*
Single layer									
Year only	29.2 (L4)	28.4 (L6)	25.9 (L8)	26.0 (L6)	24.9 (L6)	23.8 (L10)	31.0 (L4)	31.0 (L16)	29.4 (L16)
Context and year	29.1 (L6)	27.9 (L4)	26.0 (L6)	25.2 (L4)	24.5 (L6)	22.9 (L12)	30.5 (L8)	30.3 (L4)	28.2 (L4)
Contrasting Pair	29.3 (L6)	28.6 (L6)	26.5 (L6)	26.3 (L6)	25.2 (L6)	22.3 (L8)	31.2 (L4)	31.2 (L16)	29.7 (L16)
Multi layer									
Year only	28.8 (L4-9)	28.7 (L4-7)	28.2 (L4-7)	26.8 (L4-10)	25.0 (L4-10)	24.2 (L4-10)	32.6 (L4-23)	33.8 (L4-20)	31.1 (L4-20)
Context and year	28.9 (L4-5)	28.6 (L4-8)	26.0 (L4-9)	25.2 (L4-8)	24.8 (L4-10)	23.5 (L4-13)	32.2 (L4-32)	33.3 (L4-20)	31.4 (L4-17)
Contrasting Pair	29.0 (L4-11)	29.0 (L4-10)	26.3 (L4-5)	27.2 (L4-12)	25.6 (L4-11)	24.4 (L4-12)	33.2 (L4-20)	34.5 (L4-20)	31.6 (L4-20)

Table 5: Average F1 score for Taqa-1000 dataset. The * denotes PEFT LoRA fine-tuning instead of full parameter fine-tuning.

	LLaMA 7b			LLaMA 13b			LLaMA 70b		
	2020	**2021**	**2022**	**2020**	**2021**	**2022**	**2020**	**2021**	**2022**
Benchmark									
Relative Prompting	83.8	83.8	83.8	59.8	59.8	59.8	65.5	65.5	65.5
Explicit Prompting	70.1	69.2	68.3	56.9	58.2	57.9	59.3	57.7	55.1
Fine-Tuning	76.8	74.5	75.4	64.6	65.0	64.1	71.1*	70.7*	70.2*
Single Sweep									
Year Only	76.6	74.5	72.9	57.6	58.4	56.9	64.3	68.4	62.9
Context and Year	76.9	76.6	75.5	58.3	58.7	59.2	65.9	68.0	67.7
Contrasting Pair	76.3	74.4	74.2	58.5	59.2	57.0	68.0	64.7	62.7
Multi layer									
Year Only	74.5	75.0	73.4	58.2	58.3	58.1	67.3	66.3	65.0
Context and Year	78.7	74.1	72.8	57.4	57.5	57.4	67.0	66.4	65.7
Contrasting Pair	75.1	74.1	74.7	60.1	59.3	58.1	67.5	65.9	64.9

Table 6: F1 max score for Taqa-1000 dataset. The * denotes PEFT LoRA fine-tuning.

B.3 Taqa-9000 results

	LLaMA 7b			LLaMA 13b			LLaMA 70b		
	2020	**2021**	**2022**	**2020**	**2021**	**2022**	**2020**	**2021**	**2022**
Benchmark									
Relative Prompting	10.9	10.5	9.6	11.6	10.7	9.8	14.9	14.1	13.0
Explicit Prompting	12.2	12.2	11.0	12.8	12.2	11.7	17.4	16.7	16.0
Fine-tuning	13.6	12.7	12.0	14.4	14.0	13.7	17.7*	19.3*	17.7*
Multi layer									
Contrasting Pair	13.4	13.1	12.3	13.9	13.6	12.9	19.8	20.0	19.1

Table 7: Average F1 score for Taqa-9000 dataset. The * denotes PEFT LoRA fine-tuning.

	LLaMA 7b			LLaMA 13b			LLaMA 70b		
	2020	**2021**	**2022**	**2020**	**2021**	**2022**	**2020**	**2021**	**2022**
Benchmark									
Relative Prompting	38.0	38.0	38.0	34.7	34.7	34.7	43.0	43.0	43.0
Explicit Prompting	35.3	34.6	34.6	33.7	34.0	34.0	37.7	37.0	35.5
Fine-tuning	40.9	41.1	41.0	39.2	40.0	39.0	47.0*	47.1*	46.8*
Multi layer									
Contrasting Pair	38.4	38.0	37.9	36	35.9	31.5	44.3	43.9	42.4

Table 8: F1 max score for Taqa-9000 dataset. The * denotes PEFT LoRA fine-tuning.

C Generalisation

	LLaMA3.1-8b			Qwen2-7b			Gemma2-2b		
	2020	**2021**	**2022**	**2020**	**2021**	**2022**	**2020**	**2021**	**2022**
Benchmark									
Relative Prompting	20.66	18.84	17.28	20.48	20.62	18.3	18.91	19.06	17.07
Explicit Prompting	26.09	27.01	26.67	22.53	23.36	20.36	22.72	23.83	22.84
Fine-tuning	n/a	28.61	n/a	n/a	22.86	n/a	n/a	24.02	n/a
Single Sweep									
Year Only	26.07 (L8)	27.01 (L4)	27.04 (L4)	21.50 (L8)	20.99 (L4)	19.10 (L8)	21.77 (L4)	20.21 (L8)	21.46 (L10)
Context and Year	25.36 (L10)	26.99 (L10)	26.16 (L4)	21.39 (L4)	21.35 (L8)	19.00 (L8)	22.22 (L4)	22.55 (L8)	21.44 (L4)
Contrasting Pair	25.92 (L7)	27.36 (L7)	27.02 (L9)	21.13 (L4)	20.93 (L4)	19.27 (L16)	21.93 (L4)	22.32 (L8)	21.69 (L8)
Multi layer									
Year Only	27.09 (L4-7)	28.88 (L4-5)	26.63 (L4-5)	21.08 (L4-9)	21.11 (L4-11)	18.99 (L4-9)	21.79 (L4-7)	22.50 (L4-7)	21.70 (L4-7)
Context and Year	26.27 (L4-11)	27.83 (L4-10)	26.52 (L4-11)	21.39 (L4-6)	21.89 (L4-10)	19.46 (L4-12)	21.76 (L4-7)	22.50 (L4-12)	21.59 (L4-13)
Contrasting Pair	27.07 (L4-5)	27.44 (L4-9)	26.44 (L4-5)	21.33 (L4-12)	21.36 (L4-9)	19.24 (L4-12)	21.49 (L4-9)	22.74 (L4-7)	22.37 (L4-5)

Table 9: F1 score for Taqa-1000 dataset using alternative models. Fine-tuning was only applied to 2021

	LLaMA3.1-8b-Instruct			Qwen2-7b-Instruct			Gemma2-2b-Instruct		
	2020	**2021**	**2022**	**2020**	**2021**	**2022**	**2020**	**2021**	**2022**
Benchmark									
Relative Prompting	13.99	16.29	16.15	12.68	13.6	12.07	13.82	15.9	16.32
Explicit Prompting	19.9	20.99	20.32	12.75	13.67	10.75	14.73	17.82	16.22
Fine-tuning	n/a	24.77	n/a	n/a	19.85	n/a	n/a	22.29	n/a
Single Sweep									
Year Only	15.44 (L4)	16.81 (L4)	14.42 (L4)	14.37 (L12)	14.57 (L6)	13.04 (L12)	14.84 (L4)	17.10 (L4)	16.41 (L8)
Context and Year	16.78 (L12)	17.36 (L12)	16.00 (L12)	14.78 (L4)	15.63 (L4)	13.83 (L4)	14.63 (L4)	17.56 (L14)	16.12 (L12)
Contrasting Pair	16.67 (L4)	18.00 (L4)	15.70 (L4)	14.13 (L12)	14.90 (L4)	12.96 (L14)	14.61 (L6)	17.39 (L8)	16.23 (L10)
Multi layer									
Year Only	16.42 (L4-12)	17.29 (L4-12)	15.37 (L4-5)	14.00 (L4-5)	14.80 (L4-5)	12.79 (L4-5)	15.03 (L4-12)	17.86 (L4-8)	16.44 (L4-11)
Context and Year	17.54 (L4-12)	17.95 (L4-10)	15.39 (L4-5)	14.98 (L4-5)	16.07 (L4-5)	13.83 (L4-7)	14.63 (L4-10)	17.54 (L4-14)	15.81 (L4-12)
Contrasting Pair	16.96 (L4-8)	17.01 (L4-11)	14.93 (L4-11)	14.12 (L4-5)	14.99 (L4-5)	12.92 (L4-5)	14.90 (L4-10)	17.48 (L4-9)	16.42 (L4-12)

Table 10: F1 score for Taqa-1000 dataset using alternative instruct models

D Efficiency

Investigating efficiencies of AE over fine-tuning we assessed the wall time required for a set of inferences. We compare relative and AE alignment methods over the use of single layer, and multi layer sweeping to assess the wall time difference between these techniques. Overall we note that a single layer injection increases the inference time by 1.2x whilst multi layer injection increases the time by 2.5x. The fine-tuning process requires a large amount of data to be processed and a large amount of time spent searching for hyperparameters and fine-tuning the model; our experience indicated at least 5–20 minutes for smaller models such as LLaMA 7b and 13b and up to 30 minutes for LLaMA2-70b training with PEFT LoRA on 3 H100 94G GPUs.

For 7b and 13b models we conduct our testing on a single A100 80G card. For the 70b model, we conduct our testing on 2 A100 80G cards. We note that the development of a single vector made of a contrasting "year and recent" statement is a once off overhead which can be computed on average in under 0.05 seconds. Figure 11 demonstrates the wall time required for the application of this vector into 7b, 13b and 70b models across a single layer and multi-layer application using the 'contrasting prompt'.

Model	**Relative Prompt (s)**	**Single Layer (s)**	**Multi Layer (s)**
LLaMA2-7b	0.69	0.91	1.86
LLaMA2-13b	0.87	1.05	2.35
LLaMA2-70b	1.95	2.29	4.91

Table 11: Wall time (seconds) required for inference with different activation engineering methods for a statement with 10 tokens.

E Isolated effect

Experiment, when conflicted AE has the alignment year 2022, and the prompt has the alignment year 2015 using Taqa-1000.

	Invariant F1	Invariant Contains Answer	TriviaQA F1	gsm8k exact answer	gsm8k F1
No AE	35.0	75.0	43.1	9.4	53.0
AE	21.3	75.1	49.3	10.3	52.0

Table 12: Effect of Activation Engineering (AE) on invariant and TriviaQA datasets.

Experiment	**F1 (2015)**	**F1_Max**
No AE – Explicit prompt for 2015	27.5	58.2
AE – Conflicts	27.5	59.9
AE – Aligns	27.7	59.0

Table 13: Effect of AE alignment and conflict on F1 score for 2015.

F Fine-tuning with AE assistance

llama3.1-8b-Instruct	**2020**	**2021**	**2022**
FT 2021, Activation on year	26.43	26.41	25.99
FT 2021 with relative	22.17	24.77	24.68
FT 2021, with explicit	26.82	26.60	27.73
Gemma2-2b	**2020**	**2021**	**2022**
FT 2021, Activation on year	22.74	24.60	24.09
FT 2021 with relative	21.58	24.07	22.49
FT 2021, with explicit	23.29	24.92	24.54

Table 14: Performance of llama3.1-8b-Instruct and Gemma2-2b temporally fine-tuned to 2021, with scores reflecting relative, explicit and AE across the years 2020, 2021 and 2022.

G Prompting

The following was used as a preface to illicit relative answers from models:

Answer the following question: What is the capital of France?
The answer is: Paris
Answer the following question: Who wrote Harry Potter?
The answer is: J.K. Rowling
Answer the following question: Where did the Titanic sink?
The answer is: Atlantic Ocean
Answer the following question: What is the gravity of earth?
The answer is: 9.807 m/s^2
Answer the following question: Is the speed of light faster than the speed of sound?
The answer is: Yes
Answer the following question: {x}
The answer is:

The following was used as a preface to elicit explicit answers from models. The variable {year} is replaced with the year of interest, e.g. 2020, 2021, 2022, etc.:

Answer the following question: What is the capital of France?
As of year {year}, the answer is: Paris
Answer the following question: Who wrote Harry Potter?
As of year {year}, the answer is: J.K. Rowling
Answer the following question: Where did the Titanic sink?
As of year {year}, the answer is: Atlantic Ocean
Answer the following question: What is the gravity of earth?
As of year {year}, the answer is: 9.807 m/s^2
Answer the following question: Is the speed of light faster than the speed of sound?
As of year {year}, the answer is: Yes
Answer the following question: [x]
As of year {year}, the answer is: