Learning Distribution-Wise Control in Representation Space for Language Models

Chunyuan Deng [1] **Ruidi Chang** [1] **Hanjie Chen** [1]

Abstract

Interventions in language models (LMs) are applied strategically to steer model behavior during the forward pass. Learnable interventions, also known as representation fine-tuning, aim to apply pointwise control within the concept subspace and have proven effective in altering high-level behaviors. In this work, we extend this approach to the distribution level, enabling the model to learn not only pointwise transformations but also the surrounding regions of the concept subspace. We demonstrate that these methods perform effectively in early layers, with larger standard deviations correlating strongly with improved performance. Across eight commonsense reasoning and seven arithmetic reasoning benchmarks, our distribution-wise interventions consistently outperform pointwise interventions in controllability and robustness. These results illustrate that distribution-wise interventions provide a more comprehensive method for steering model behavior and enabling finer-grained control over language models. The code is at: https://github.com/chili-lab/D-Intervention.

1. Introduction

As language models (LMs) continue to grow in complexity and capability, understanding and controlling their behavior has become increasingly critical (Olah et al., 2015; Geva et al., 2021; Yu et al., 2023; Geiger et al., 2023). Recent advances in interpretability research have highlighted the potential of model interventions–targeted modifications to model behavior during forward passes–as a promising approach to achieving this control (Meng et al., 2022; Conmy et al., 2023; Ghandeharioun et al., 2024). These interventions allow researchers to steer model behavior, often by manipulating representation in the model's latent space.

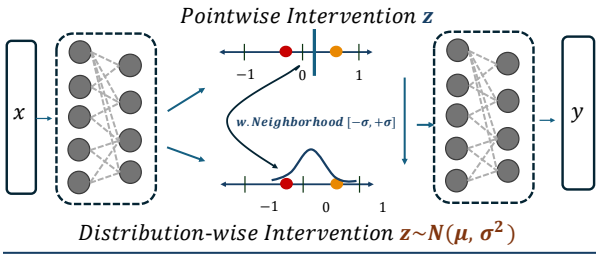


Figure 1. **Distribution-wise vs. Pointwise Intervention.** This is an intuitive yet effective adaptation, as previous research suggests that the concept space is continuous (Gandikota et al., 2023).

A core challenge in intervention-based methods lies in *advancing from low-level control to high-level control*. Low-level control includes steering the model to output antonyms/synonyms, or binary labels (e.g., positive/negative sentiment). While these tasks served as foundational benchmarks for early intervention research, the field needs to address more complex, high-level behaviors (Zhang & Nanda, 2024). These tasks require interventions that operate at a deeper, more abstract level, capturing the intricate relationships and dependencies within the model's hidden representations.

Learnable interventions, or *representation fine-tuning* (Wu et al., 2024b; Yin et al., 2024), have emerged as a promising solution. It enables more powerful high-level control, often outperforming parameter-efficient fine-tuning (PEFT) methods on tasks like commonsense question answering (QA), mathmetical reasoning and alignment tasks, while using 10x-100x fewer parameters (Houlsby et al., 2019; Hu et al., 2022; 2023; Liu et al., 2024). This highlights the potential to modify model behavior in a finer-grained manner.

A potential improvement for these methods is to explore an ideal "concept space". This space should be continuous, as previous research suggests that once an intervention vector is identified, its magnitude can be adjusted to control its effect (Gandikota et al., 2023; Zou et al., 2023; Turner et al., 2023). This implies that even if an intervention is learned, its neighboring regions should also produce relevant effects.

[1]Department of Computer Science, Rice University. Correspondence to: Chunyuan Deng <chunyuan.deng@rice.edu>, Hanjie Chen <hanjie@rice.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

The key research question then arises—how to effectively explore this region.

As shown in Figure 1, one intuitive method is to *replace deterministic nodes with stochastic ones to directly learn the latent distribution*. A common way to achieve this is through reparameterization (Kingma & Welling, 2014), which enables efficient gradient-based optimization by decoupling randomness from the model's parameters. Specifically, the stochastic node is expressed as a differentiable function of a base distribution and the model's parameters. This allows for standard backpropagation while preserving the ability to sample from the learned distribution.

Building on these insights, we propose a simple yet effective improvement to enhance the exploration of the concept space in intervention-based methods. Specifically, we replace a deterministic node (neural network), with two separate networks. These networks independently learn the mean (μ) and log variance ($\log \sigma^2$) of the latent distributions through gradient descent. This approach serves as a valid **drop-in replacement** for existing methods.

We follow the experimental setup of prior work (Hu et al., 2023; Wu et al., 2024b) and conduct comprehensive experiments, spanning eight commonsense reasoning benchmarks and seven mathematical reasoning benchmarks. We test performance on Llama-family models (Touvron et al., 2023b; Dubey et al., 2024) under both layer-wise and all-layer configurations.

In our layer-wise experiments, we observed an intriguing performance gain: replacing deterministic nodes with stochastic counterparts in early layers significantly improved model performance, yielding gains of $+4\%$ to $+6\%$. Furthermore, we found that these gains strongly correlate with the learned variance of the stochastic nodes, suggesting that broader neighborhood exploration during training leads to enhanced performance.

We then apply insights from the intervention layer analysis to experiments across all layers. By varying the ratio of layers subjected to distribution-wise intervention, we identify an effective strategy: replacing the first few layers with stochastic nodes while retaining determinism in subsequent layers. This approach achieves consistent performance gains across all 15 benchmarks, along with significant improvements in robustness compared to pointwise intervention. These findings highlight the superiority of learned distribution-wise intervention over its pointwise counterpart.

Our key contributions are:

- We propose a simple yet effective intervention method by replacing deterministic nodes with stochastic ones, enabling better exploration of the concept space.
- We demonstrate that this approach significantly improves model performance by intervening early layers.
- We find that a mixed strategy—replacing only the first few layers with stochastic nodes while keeping the later layers deterministic—yields the best results in terms of both performance and robustness.

2. Related Work

Reparametrization. The reparameterization trick is a widely used technique that enables neural networks to learn from sampling through gradient descent. It is commonly applied in variational autoencoders (VAE) (Kingma & Welling, 2014; Tian et al., 2020) and variational information bottlenecks (VIB) (Tishby et al., 2000; Tishby & Zaslavsky, 2015; Alemi et al., 2017) to help models learn latent distributions. VIB has been shown to effectively control the word embedding layer in language models (Li & Eisner, 2019; Chen & Ji, 2020) or downstream tasks (White et al., 2020; Mahabadi et al., 2021; Behjati et al., 2023). The term "variational" in VAE/VIB refers to variational inference, where KL divergence is used to measure how close the approximation is to the true distribution. However, in our setting, we remove the KL loss term, making it no longer variational inference. Instead, it becomes a relaxed approach that allows language models to learn distributions without constraints.

Representation Fine-tuning and PEFT. Recently, representation fine-tuning has emerged as a method to provide high-level control over LMs' behavior. Two concurrent works, ReFT (Wu et al., 2024b) and LoFiT (Yin et al., 2024), address this issue from different perspectives. ReFT builds on the theory of distributed alignment search (DAS) (Geiger et al., 2021; 2023), demonstrating that orthogonal properties in latent subspaces are crucial for enabling task-independent control *between transformer blocks*. LoFiT, on the other hand, draws inspiration from traditional intervention research, focusing on a paradigm of localization and editing (Meng et al., 2022; Stolfo et al., 2023). Both approaches share a common goal: bringing interpretability research into the realm of high-level behavior control, comparable to parameter-efficient fine-tuning (PEFT) methods like adapters (Houlsby et al., 2019; Pfeiffer et al., 2020; Fu et al., 2021; Hu et al., 2023; Zhang et al., 2023) and LoRA (Hu et al., 2022; Liu et al., 2024; Zhang et al., 2024). Their results show that learnable interventions can match or even outperform PEFT methods, often with significantly fewer resources. Our method aligns more closely with ReFT, as it involves learning latent distributions between encoder and decoder (i.e., between transformer blocks). Therefore, we will primarily evaluate our stochastic vs. deterministic node approach within the ReFT framework.

Intervention-based interpretability. Intervention-based interpretability focuses on manipulating a model's internal states to understand how LMs represent various behaviors (Subramani et al., 2022; Zou et al., 2023; Turner et al., 2023; Li et al., 2023a). By intervening on specific linear subspaces of latent representations, researchers have revealed that human-interpretable concepts (Rumelhart et al., 1986), such as linguistic features (e.g., gender, number) (Hewitt & Manning, 2019; Lasri et al., 2022; Wang et al., 2023; Hanna et al., 2023; Arora et al., 2024; Huang et al., 2024) and logical reasoning, are often encoded linearly within these models (Wu et al., 2023; Deng et al., 2024; Gur-Arieh et al., 2025). Techniques like concept erasure and subspace interventions have been instrumental in disentangling such attributes (Belrose et al., 2023; Ravfogel et al., 2022), enabling targeted modifications to improve model fairness, explainability, and task performance (Nanda et al., 2023; Park et al., 2024). These works showcase that the representation space of language models encodes rich, structured information that is highly relevant to tasks, enabling more effective and targeted interventions.

3. Preliminary

We now first introduce the background of our intervention methods.[1] First, we outline the formulation of transformer-based decoder LMs and their layer-wise hidden representations. Then, we will provide a unified view of intervention from an information-theoretic perspective.

3.1. Transformer Architecture

Transformer-based autoregressive LMs (Vaswani et al., 2017) aim to predict the probability of a sequence of tokens. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the input sequence, where each x_i represents a token in the sequence. Let $Y = \{y_1, y_2, ..., y_m\}$ denote the output sequence. Overall, the goal of next-token prediction in language modeling can be formally represented as estimating $P(Y|X)$.

The hidden representations at each layer of the model act as latent variables Z, encoding intermediate abstractions that bridge X and Y:

$$Z^{(l)} = \{z_1^{(l)}, z_2^{(l)}, \ldots, z_n^{(l)}\}, \quad l = 1, 2, \ldots, L,$$

where l indexes the layers of the language model, and $z_i^{(l)}$ represents the hidden state of token x_i at layer l.

3.2. Layer-Wise Representation Transformation

Each layer in a language model is designed to transform the latent representations $Z^{(l)}$ using contextual information from neighboring tokens. These transformations can be expressed as:

$$Z^{(l+1)} = Attn(Z^{(l)}) + FFN(Attn(Z^{(l)})),$$

where FFN represents a feed-forward networks (FFN) with the input from $Z^{(l)}$, and $Attn(\cdot)$ represents the self-attention module within the transformer block.

3.3. Intervention in Language Models

In this work, we provide an *information-theoretical view* of model intervention. In established information theory research, a common method for estimating mutual information is achieved by inserting an auxiliary network (e.g., a variational autoencoder) at a specific layer.

Interestingly, in intervention research, interventions in a language model are formalized as a function f_ϕ, parameterized by ϕ, that transforms hidden representations $Z^{(l)}$ at layer l to modified representations $\hat{Z}^{(l)}$:

$$\hat{Z}^{(l)} = f_\phi(Z^{(l)}). \tag{1}$$

We observe a **connection** here: the auxiliary network used in information theory for mutual information estimation can be viewed as a specific form of intervention. It represents a special case where the transformation function f_ϕ is a learnable variational autoencoder.

In practice, the goal of intervention can be viewed as improving downstream task performance. For learnable interventions, this translates to minimizing the cross-entropy (CE) loss between the predicted and true outputs:

$$\mathcal{L}_{CE} = -\mathbb{E}_{(X,Y)}\left[\log P(Y|f_\phi(Z^{(l)}))\right]. \tag{2}$$

The cross-entropy loss can be directly recognized as the conditional entropy:

$$\mathcal{L}_{CE} = -\mathbb{E}_{(X,Y)}\left[\log P(Y|f_\phi(Z^{(l)}))\right] = H(Y|f_\phi(Z^{(l)})). \tag{3}$$

The mutual information between Y and the transformed representations is defined as:

$$I(Y; f_\phi(Z^{(l)})) = H(Y) - H(Y|f_\phi(Z^{(l)})). \tag{4}$$

Rearranging this expression:

$$H(Y|f_\phi(Z^{(l)})) = H(Y) - I(Y; f_\phi(Z^{(l)})). \tag{5}$$

Substituting this into the cross-entropy loss:

$$\mathcal{L}_{CE} = H(Y|f_\phi(Z^{(l)})) = H(Y) - I(Y; f_\phi(Z^{(l)})). \tag{6}$$

Given that $H(Y)$ is constant with respect to the intervention parameters ϕ, minimizing the cross-entropy loss is equivalent to maximizing the mutual information:

$$\arg\min_\phi \mathcal{L}_{CE} \equiv \arg\max_\phi I(Y; f_\phi(Z^{(l)})). \tag{7}$$

[1] In this work, we primary focus on learnable intervention between layers (i.e. transformer block).

This formulation captures the fundamental goal of interventions: *to transform the internal representations such that they become maximally informative about the target output*. Conceptually, this optimization seeks the intervention that best preserves and amplifies the signal relevant to the task while potentially filtering out irrelevant information.

4. Distribution-Wise Intervention

In this section, we first introduce the motivation for distribution-wise control and provide a detailed description of the improvements, specifically replacing a deterministic node with a stochastic node that can learn from sampling.

4.1. Motivation

Many previous studies have found that the effect of an intervention can be controlled by adjusting its magnitude (Gandikota et al., 2023; Turner et al., 2023; Han et al., 2023). The intervention effect should not be limited to a single point; rather, its surrounding neighborhood must also exhibit relevant effects. This suggests that the impact of an intervention propagates across related regions. A useful analogy is the transition from autoencoders (AE) to variational autoencoders (VAE). VAEs replace deterministic nodes with stochastic sampling, allowing the model to learn latent distributions directly (Kingma & Welling, 2014). We explore applying the technique to intervention research, investigating whether they can help learn better interventions.

4.2. Stochastic Intervention Reparameterization

To effectively learn distributions through stochastic nodes, we employ the reparameterization trick (Kingma & Welling, 2014). This technique enables gradient-based optimization through sampling by reformulating the random sampling process as a deterministic function of the distribution parameters and an auxiliary noise variable.

Consider a simple deterministic MLP layer that transforms input representation Z through:

$$\hat{Z} = \text{MLP}(Z) = W^T Z + b. \tag{8}$$

We replace this with a stochastic layer that learns a distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. Instead of directly sampling from this distribution, which would break gradient flow, we reparameterize the sampling process:

$$\boldsymbol{\mu} = \text{MLP}_\mu(Z), \tag{9}$$

$$\log \sigma^2 = \text{MLP}_{\log \sigma^2}(Z), \tag{10}$$

$$\boldsymbol{\sigma} = \exp\left(\tfrac{1}{2} \log \sigma^2\right), \tag{11}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, I), \tag{12}$$

$$\hat{Z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}. \tag{13}$$

Here, MLP_μ and $\text{MLP}_{\log \sigma^2}$ learn the distribution parameters, while $\odot$ denotes element-wise multiplication. The stochasticity comes from $\boldsymbol{\epsilon}$ (random noise), and it allows gradients to flow through $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ during backpropagation while maintaining the stochastic nature of the transformation through $\boldsymbol{\epsilon}$.

4.3. Training Objective

Given a frozen base language model $\mathcal{M}$ and trainable stochastic intervention layers $\{\mathcal{I}_l\}_{l=1}^L$ inserted between transformer blocks, we minimize the cross-entropy loss over the next-token prediction task:

$$\mathcal{L} = -\mathbb{E}_{(X,Y)} \left[\log P_{\mathcal{M} \circ \mathcal{I}}(Y | f_\phi(Z^{(l)})) \right],$$

where $\mathcal{M} \circ \mathcal{I}$ denotes the composed system of the frozen language model with our stochastic intervention layers. During training, gradients flow through the reparameterized stochastic networks back to the learnable parameters $\{\phi_\mu^{(l)}, \phi_\sigma^{(l)}\}_{l=1}^L$ of the MLP_μ and MLP_σ networks..

4.4. Model-Specific Clamping

To address numerical instability issues arising from large sampling variances, we introduce model-specific clamping based on the weight distributions of the target language model. Given a language model $\mathcal{M}$ with intervention at layer l, we define the clamping boundaries using the statistics of adjacent layer weights. Let $W^{(l)}$ and $W^{(l+1)}$ denote the weight matrices before and after the intervention layer respectively. We define the clamping bounds as:

$$v_{\text{min}} = \min(\min(W^{(l)}), \min(W^{(l+1)})), \tag{14}$$

$$v_{\text{max}} = \max(\max(W^{(l)}), \max(W^{(l+1)})). \tag{15}$$

This model-specific clamping ensures that the interventions remain within the natural range of the model's weight distributions, helping to maintain stability while preserving the model's learned representations. The bounds are computed once before training and remain fixed throughout the intervention process.

Capturing Uncertainty with Stochastic Layers

During training, optimizing both the mean and variance of the learned distribution enables the model to effectively capture uncertainty in the intervention space. This stochastic approach offers two key advantages:

- It facilitates exploration of the intervention neighborhood through sampling.
- It allows the model to learn and represent uncertainty in intervention effects, improving its robustness and adaptability.

5. Experiment Setup

To evaluate our distribution-level intervention methods compared with pointwise intervention, we evaulate our methods on more than ten datasets with full combination of different hyperpameter tuning. Generally, we follow the standard setup of previous SOTA methods like ReFT (Wu et al., 2024b), and our codebase is built on pyenve (Wu et al., 2024c). ReFT's evaluation framework is also derived from prior work like (Hu et al., 2023; Liu et al., 2024; Wu et al., 2024a). Similar to these previous work, we evaluate Llama-series model (Touvron et al., 2023a;b; Dubey et al., 2024) ranging from Llama-7B/13B to Llama-3-8B. We conducted all experiments using a single NVIDIA RTX A6000 GPU with mixed precision (`bfloat16`) enabled.

Our evaluation is divided into two parts: (i) *layer-wise setting* and (ii) *all-layer setting*. First, we analyze layer-wise control by experimenting with different types of interventions. We then explore how replacing these interventions with distribution-level controls affects performance. Finally, we evaluate the interventions in an all-layer setting and compare with the results in previous literature.

5.1. Baselines

For the *layer-wise setting*, in addition to RED (Wu et al., 2024a) and ReFT, we also include simple MLP and SwiGLU (Shazeer, 2020) as baselines to evaluate the impact of distribution-level intervention.[1] Concrete formats are provided below.

For the *all-layer setting*, we perform a comparative analysis of ReFT and previous parameter-efficient fine-tuning (PEFT) methods. These include: Prefix-tuning (Li & Liang, 2021), RED (Wu et al., 2024a), LoRA (Hu et al., 2022), DoRA (Liu et al., 2024) and ReFT (Wu et al., 2024b).

Intervention Functions

Pointwise intervention function f_ϕ:

- **MLP**: $\hat{Z} = W^T Z + b$
- **RED**: $\hat{Z} = W \odot Z + b$
- **SwiGLU**: $\hat{Z} = (W \odot Z + b) \odot GELU(Z)$
- **ReFT**: $\hat{Z} = Z + R(W^T Z + b - R^T Z)$

Distribution-wise intervention function f'_ϕ:

- **D-MLP**: $\hat{Z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
- **D-RED**: $\hat{Z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
- **D-SwiGLU**: $\hat{Z} = \boldsymbol{\mu} \odot GELU(Z) + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
- **D-ReFT**: $\hat{Z} = Z + R(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} - R^T Z)$

[1] We denote the distribution-level variants of *ReFT* as *D-ReFT*, and similar notation applies to other methods

5.2. Benchmark

We evaluate our methods on seven commonsense reasoning benchmarks and seven arithmetic reasoning benchmark.

For commonsense reasoning, we have BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC-e, ARC-c (Clark et al., 2018) and OBQA (Mihaylov et al., 2018). The input format is multi-choice QA, given a context or a question with multiple answer choices. The output is as simple as the selected choice, without CoT rationales.

For arithmetic reasoning, we have AddSub (Hosseini et al., 2014), SingleEQ (Koncel-Kedziorski et al., 2015), MultiArith (Roy & Roth, 2015), AQuA (Ling et al., 2017), GSM8K (Cobbe et al., 2021), MAWPS (Koncel-Kedziorski et al., 2016), and SVAMP (Patel et al., 2021). For the arithmetic reasoning benchmarks, CoT rationale are given before the final answer.

For all benchmarks, We use the same prompt template as in Hu et al. (2023); Wu et al. (2024b). We also remove leading and trailing whitespace in the dataset.

5.3. Hyperparameter Tuning

For the commonsense reasoning benchmark, we train the model using the `Commonsense170K` dataset. For arithmetic reasoning benchmarks, we use the `Math10K` dataset. These datasets are combined training sets from their original benchmarks. We use a portion of the training set from GSM8K as a development set to tune the best hyperparameters and apply this set of hyperparameters to report the test scores. We do not optimize directly on the test set. This setting is the same as that used by Wu et al. (2024b).

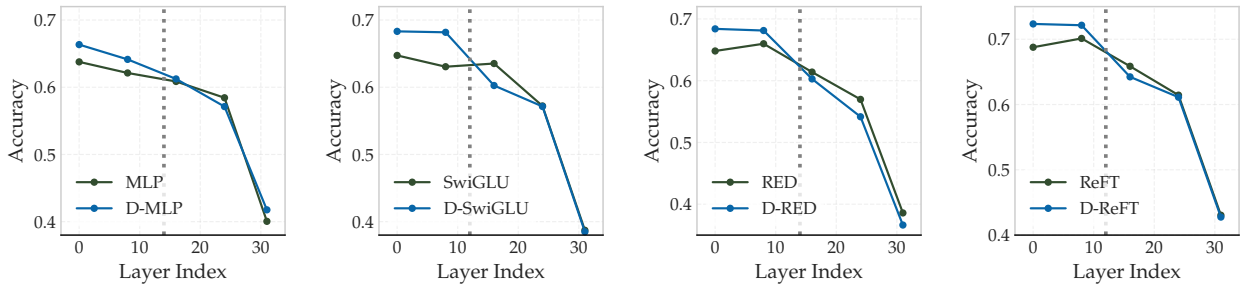
Key parameters include the *intervention layer* (l), *noise scale* (ϵ), *subspace rank* (r), *intervention position* (p), *batch size* (bs), *training epochs* (e), and *learning rate* (lr). These parameters are tuned on the development set, but an ablation study is not included in the main text. Detailed values are provided in Appendix B.

6. Layer-Wise Intervention

Previous work often reports results using the all-layer setting (i.e., interventions applied across all layers). In this study, we first conduct layer-wise ablation experiments to identify where the performance gains come from.

6.1. Abalation Study: D-Intervention Layer l

We evaluate distribution-level controllability in arithmetic reasoning across seven benchmarks: AddSub, SingleEQ, MultiArith, AQuA, GSM8K, MAWPS, and SVAMP. All



(a) MLP (b) SwiGLU (c) RED (d) ReFT

Figure 2. **Performance of different layer-wise D-interventions with respect to the intervention layer**. We report the average score of Llama-3-8B on seven arithmetic benchmarks: AddSub, SingleEQ, MultiArith, AQuA, GSM8K, MAWPS, and SVAMP. Notably, D-intervention at early layers yields the best performance, highlighting a significant discrepancy across layers.

methods are tuned on the development set, and we report the average results over three runs with different seeds.

Results. As shown in Figure 2, ReFT maintains superior performance compared to RED, SwiGLU, and MLP across all four methods. Both standard intervention and its distribution-level variant reveal a consistent pattern: deeper layers lead to significant performance declines. The accuracy drops from 0.7 to approximately 0.4 as interventions move to later layers, demonstrating that later-layer interventions are more challenging than those in earlier layers.

This follows from data processing inequality (see appendix A), which states that deep layers cannot recover information lost in earlier layers. Thus, intervention should be applied early to preserve useful information before transformations degrade it. Across MLP, SwiGLU, RED, and ReFT, their *D-variants consistently boost accuracy by around +4% in early layers*, which is higher than the improvements from LoRA → DoRA → ReFT (Liu et al., 2024; Wu et al., 2024b). This highlights that distribution-level intervention is an effective improvement when applied early in the network.

6.2. Ablation Study: D-Intervention Noise ϵ

We then conduct an ablation study on the stochastic node $\boldsymbol{\epsilon}$. In our method, $\boldsymbol{\epsilon} \odot \boldsymbol{\sigma}$ plays a key role in transforming a deterministic node into a stochastic one to learn the distribution. Here, $\boldsymbol{\sigma}$ represents the standard deviation matrix of the latent variable Z, while $\boldsymbol{\epsilon}$ controls the magnitude of this learning effect. Adjusting $\boldsymbol{\epsilon}$ allows us to explore the true distribution or the ideal concept space.

Results. We vary the scaling factor λ applied to $\boldsymbol{\epsilon}$ from 0 to 3.0, using a step size of 0.2, in the D-ReFT setting. As shown in Figure 3, performance improves from ReFT to D-ReFT with (scaling factor of $\boldsymbol{\epsilon} = \mathbf{1}$) where the best results are achieved. This showcase that default setting with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ still has stable performance. However,

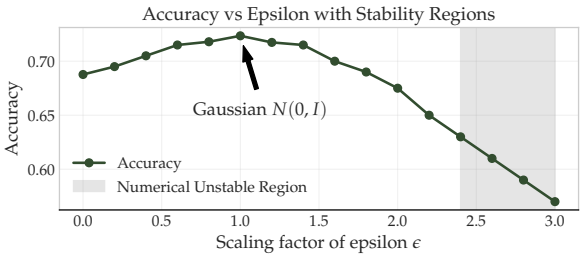
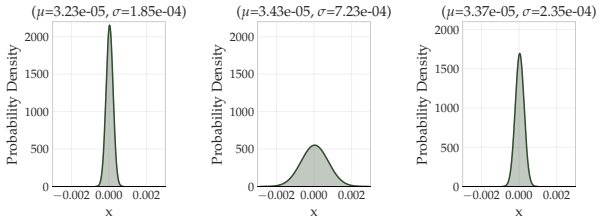


Figure 3. **Accuracy with different scaling factors for ϵ in D-ReFT.** When the scaling factor for $\boldsymbol{\epsilon} = 0$, the method reduces to the original ReFT. For scaling factors of $\boldsymbol{\epsilon} > 2.4$, D-ReFT enters a region prone to numerical instability due to large variance.



(a) $Acc = 0.688$ (b) $Acc = 0.723$ (c) $Acc = 0.694$

Figure 4. **Correlation between accuracy and std σ**. The scaling factors λ of $\boldsymbol{\epsilon}$ are 0.2, 1.0, and 1.8 for subfigures (a), (b), and (c).

increasing $\boldsymbol{\epsilon}$ further introduces higher variance, making training more difficult and causing numerical instability.

We explore the correlation between accuracy and the standard deviation of the learned distribution. Using all test set samples, we compute the average accuracy for different standard deviations. As shown in Figure 4, accuracy and standard deviation exhibit a positive correlation: *distributions with higher standard deviation tend to perform better, while lower standard deviation is associated with worse performance.* This suggests that distribution-level interventions are effective because they explore the neighborhood region of pointwise interventions.

Table 1. Performance comparison of LLaMA-1 7B/13B, Llama-2 7B and Llama-3 8B against existing PEFT methods on eight commonsense reasoning datasets. *Performance results of all baseline methods are taken from Wu et al. (2024b); Liu et al. (2024). **D-ReFT**$_{25\%}$, for example in Llama-3-8B, represents replace first 8 layers to D-ReFT and remain the left 24 layers as ReFT intervention.

Model	PEFT	Params (%)	Accuracy (↑)								
			BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	Avg.
ChatGPT*	—	—	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA-7B	PrefT*	0.039%	64.3	76.8	73.9	42.1	72.1	72.9	54.0	60.6	64.6
	Adapter^{S}*	1.953%	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	Adapter^{P}*	3.542%	67.9	76.4	78.8	69.8	78.9	73.7	57.3	75.2	72.3
	LoRA*	0.826%	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	DoRA*	0.838%	68.5	82.9	79.6	84.8	80.8	81.4	65.8	81.0	78.1
	ReFT*	0.031%	69.3	84.4	80.3	93.1	84.2	83.2	68.2	78.9	80.2
	D-ReFT$_{25\%}$ (Ours)	0.046%	**72.1**	**87.4**	**81.1**	**93.7**	**85.4**	**84.7**	**71.7**	**80.4**	**82.2**
LLaMA-13B	PrefT*	0.031%	65.3	75.4	72.1	55.2	68.6	79.5	62.9	68.0	68.4
	Adapter^{S}*	1.586%	71.8	83.0	79.2	88.1	82.4	82.5	67.3	81.8	79.5
	Adapter^{P}*	2.894%	72.5	84.9	79.8	92.1	84.7	84.2	71.2	82.4	81.5
	LoRA*	0.670%	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	DoRA*	0.681%	72.4	84.9	81.5	92.4	84.2	84.2	69.6	82.8	81.5
	ReFT*	0.025%	72.1	86.3	81.8	95.1	87.2	86.2	**73.7**	84.2	83.3
	D-ReFT$_{25\%}$ (Ours)	0.037%	**74.3**	**87.1**	**83.3**	**95.2**	**89.3**	**87.1**	73.6	**85.9**	**85.1**
Llama-2 7B	LoRA*	0.826%	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA*	0.838%	**71.8**	83.7	76.0	89.1	82.6	83.7	68.2	82.4	79.7
	ReFT*	0.031%	71.1	83.8	80.8	**94.3**	84.5	85.6	72.2	82.3	81.8
	D-ReFT$_{25\%}$ (Ours)	0.046%	71.3	**86.7**	**81.8**	94.1	**87.3**	**86.1**	**73.0**	**84.2**	**83.6**
Llama-3 8B	LoRA*	0.700%	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA*	0.710%	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	ReFT*	0.026%	75.1	90.2	82.0	**96.3**	87.4	92.4	81.6	87.5	86.6
	D-ReFT$_{25\%}$ (Ours)	0.039%	**78.3**	**93.4**	**83.7**	96.1	**89.7**	**94.9**	**83.1**	**89.4**	**89.1**

7. All-layer Intervention

We evaluate D-ReFT's performance across all network layers, focusing on how layer allocation impacts results. Since the strongest improvements occur in early layers, we conduct an ablation study to test a mixed strategy of pointwise and distribution-level interventions. Specifically, we vary how many early layers are replaced with D-ReFT. We compare four configurations: D-ReFT$_{25\%}$, D-ReFT$_{50\%}$, D-ReFT$_{75\%}$, and D-ReFT$_{100\%}$, where the subscript indicates the percentage of early layers replaced with D-ReFT.

7.1. Commonsense Reasoning

Table 1 shows the performance comparison on eight different commonsense reasoning benchmarks. The results show that D-ReFT$_{25\%}$ consistently outperforms its original pointwise version ReFT and existing PEFT methods across all LLaMA models. These results highlight the effectiveness of D-ReFT in enhancing reasoning performance while maintaining efficiency. Table 2 presents the results of mixed strategies for ReFT and D-ReFT. We observe a consistent trend: applying D-ReFT to the first 25% of layers and ReFT to the remaining layers yields the best performance,

Table 2. Accuracy (%) in the commonsense tasks when replacing ReFT to D-ReFT intervention for the top % layers.

Model $\mathcal{M}$	ReFT	**D-ReFT**$_{25\%}$	D-ReFT$_{50\%}$	D-ReFT$_{75\%}$	D-ReFT$_{100\%}$
LLaMA-7B	80.2	82.2	81.0	79.4	79.3
LLaMA-13B	83.3	85.1	83.9	83.0	82.6
LLaMA-2-7B	81.8	83.6	82.0	81.9	80.9
LLaMA-3-8B	86.6	89.1	86.7	85.9	85.1

while applying D-ReFT to greater than some threshold (i.e., introducing too much randomness) during training would make the model hard to converge. This suggests that a mixed strategy—using stochastic nodes in early layers and deterministic nodes in later layers—is optimal for language models.

Interpretation. This pheonmenon could be understood as early layers in language models has the rich relevant information of the input X. By applying distribution-level interventions to these layers, the model retains richer, more flexible representations of uncertainty, avoiding premature over-commitment to specific features. This stochasticity allows the model to propagate diverse hypotheses downstream, which later layers—specializing in high-level reasoning and task-specific logic—can refine using deterministic, pointwise interventions.

Table 3. Performance comparison of LLaMA-1 7B, Llama-2-7B and Llama-8B against existing PEFT methods on four arithmetic reasoning datasets. **D-ReFT**$_{25\%}$, for example in Llama-3-8B, represents replace first 8 layers to D-ReFT and remain the left 24 layers as ReFT intervention.

Model	PEFT	Params (%)	Accuracy (↑)							
			MultiArith	GSM8K	SVAMP	MAWPS	AddSub	AQuA	SingleEq	Avg.
LLaMA-7B	LoRA	0.826%	88.7	25.4	46.9	74.2	82.7	22.7	78.1	59.8
	RED	0.039%	89.1	23.2	45.2	73.1	83.1	21.6	**79.3**	58.9
	ReFT	0.031%	89.3	25.0	44.7	76.5	**83.3**	23.2	77.6	59.9
	D-ReFT$_{25\%}$ (Ours)	0.046%	**89.5**	**26.0**	**47.7**	**77.2**	83.0	**26.4**	78.9	**61.2**
LLaMA-2-7B	LoRA	0.670%	88.4	**30.3**	48.4	77.4	83.2	25.6	80.2	61.9
	RED	0.031%	90.5	29.2	**51.6**	75.9	83.2	26.2	81.6	62.0
	ReFT	0.025%	90.2	29.6	49.7	77.7	84.8	24.0	81.9	62.3
	D-ReFT$_{25\%}$ (Ours)	0.037%	**91.7**	30.2	51.0	**78.2**	**85.6**	**26.7**	**82.5**	**63.7**
LLaMA-3-8B	LoRA	0.670%	98.0	61.3	71.7	89.0	92.7	30.3	91.5	76.3
	RED	0.031%	97.8	58.2	72.0	88.9	92.9	**31.0**	92.7	75.9
	ReFT	0.025%	**98.5**	60.0	72.3	89.1	93.2	30.0	93.1	76.6
	D-ReFT$_{25\%}$ (Ours)	0.037%	97.4	**61.7**	**73.6**	**91.6**	**93.4**	30.3	**94.1**	**77.4**

7.2. Arithmetic Reasoning

We evaluate the performance of our proposed distribution-level intervention method based on ReFT against existing PEFT and intervention methods (LoRA, RED, and ReFT) on arithmetic reasoning datasets using LLaMA-1 7B, LLaMA-2 7B, and LLaMA-3 8B (see Table 3). Our method consistently outperforms baselines while maintaining parameter efficiency. Notably, D-ReFT$_{25\%}$ achieves the highest average accuracy across all model sizes and excels in key tasks such as GSM8K and SingleEq. These results demonstrate its effectiveness in improving arithmetic reasoning with minimal parameter overhead, making it a promising lightweight replacement for existing ReFT methods.

7.3. Influence of Params (%) in Intervention

A potential concern is that D-interventions introduce extra parameters for variance calculation, potentially explaining the performance gains. We conduct an ablation study on the *subspace rank* (r) to control how many parameters are used in the experiment. We set the rank to 8, 16, 32, 64, and 128, increasing the parameter count by 2× in each setting.

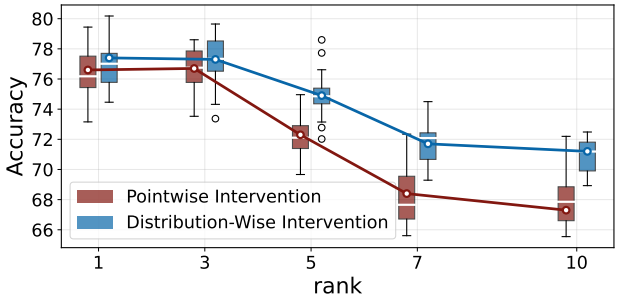


Figure 5. Accuracy with different choices of rank in Llama-3-8B for arithmetic reasoning tasks.

Analyzing the results of low-rank settings (Figure 5), we first find that ReFT's performance does not improve with higher ranks (more parameters); instead, all methods suffer from a decrease in performance. This suggests that parameter quantity alone fails to drive gains. Both ReFT and D-ReFT peak at ranks 8 and 16, underscoring the efficacy of targeted interventions in lower-dimensional subspaces over sheer parameter growth.

7.4. Robustness Evaluation

During our preliminary studies, we tried both synonym replacement (using WordNet (Miller, 1994)) and paraphrase generation (using back translation). However, our empirical analysis revealed that these semantically-preserving transformations produced insufficient perturbation magnitude to effectively distinguish between intervention methodologies. Therefore, we implemented a more challenging setting to evaluate the robustness of ReFT and D-ReFT variants on arithmetic tasks by randomly deleting N non-arithmetic words from the benchmark and observing the influence on accuracy.

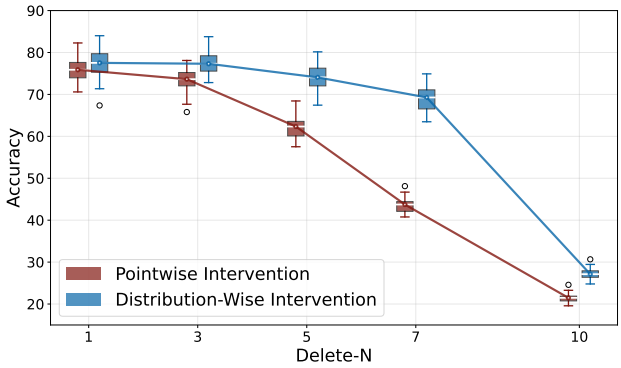


Figure 6. Accuracy with randomly deleting N words for testset in Llama-3-8B for arithmetic reasoning tasks.

As D-ReFT learns a distribution over ReFT intervention, we find it exhibits greater robustness to these perturbations (see Figure 6). When fewer than 8 words are deleted, the accuracy for D-ReFT remains stable, while its pointwise variant drops by approximately 30%. Although deleting more than 10 words leads to a significant accuracy decline for both methods, the distribution-level variants still demonstrate notable robustness against adversarial attacks. This suggests that distribution-level interventions additionally provide advantages in terms of robustness.

8. Test-Time Stochasticity: Controlled Temperature Scaling

While our distribution-wise interventions demonstrate significant improvements during training, a critical question remains: how should the learned stochastic distributions be utilized during inference? We then investigate *controlled stochasticity* through temperature scaling, which allows fine-grained control over the degree of randomness at test time.

We introduce a temperature parameter τ that scales the learned variance during inference:

$$\hat{Z} = \boldsymbol{\mu} + (\tau \cdot \boldsymbol{\sigma}) \odot \boldsymbol{\epsilon}, \quad \epsilon \sim \mathcal{N}(0, I) \tag{16}$$

where $\tau \geq 0$ controls the stochasticity level:

$$\tau = 0 : \text{Deterministic inference} \quad (\hat{Z} = \mu) \tag{17}$$

$$\tau = 1 : \text{Training-time stochasticity} \tag{18}$$

$$\tau > 1 : \text{Increased exploration} \tag{19}$$

$$0 < \tau < 1 : \text{Reduced stochasticity} \tag{20}$$

We set the τ to $0, 0.5, 1, 2$ as the ablation study for this section. Specifically we introduce the experiment of instruction-tuning to observe the task difference w.r.t different setting. We use Alpaca-Eval v1.0 (Li et al., 2023b) for instruction tuning. By default, version 1.0 calculates the win rate against text-davinci-003, with GPT-4 serving as the judge. The prompt template is provided by Alpaca-Eval, and all models in the Alpaca-Eval benchmark use this template for evaluation. For training, we use UltraFeedback (Cui et al., 2024), a high-quality instruction-tuning dataset that covers various aspects like general IT knowledge, truthfulness, honesty, and helpfulness to assess model performance. This setup aligns with the previous work on RED and ReFT.

We adopt the recommended hyperparameter settings from the paper for baseline methods like ReFT. For D-ReFT, we directly applied the params used in the math arithmetic learning datasets. All results are reported over three runs.

Results. Table 4 reveals distinct patterns in how temperature scaling affects different task categories. Lower temperature values ($\tau < 1$) consistently improve performance on commonsense and arithmetic reasoning tasks, with deterministic inference ($\tau = 0$) achieving the largest gains of $+0.7$ and $+0.6$ points respectively. Conversely, these same low-temperature settings degrade instruction following performance, suggesting that deterministic processing may be too rigid for the diverse response patterns required in instruction-based tasks. Higher temperature values ($\tau \geq 1$) reverse this trend, with $\tau = 2$ providing substantial improvements in instruction following while simultaneously degrading performance on reasoning tasks that benefit from more focused, consistent processing.

Table 4. Performance of D-ReFT with Different Temperature Values Across Task Types.

Method	Commonsense	Arithmetic	Instruction Following
D-ReFT (Baseline)	88.6	76.6	82.4
+ $\tau = 0$	89.3	79.2	80.1
+ $\tau = 0.5$	88.7	78.4	81.8
+ $\tau = 1$	86.9	76.8	83.7
+ $\tau = 2$	85.1	75.2	85.9

9. Conclusion

In this work, we introduced a distribution-wise intervention framework that extends traditional pointwise intervention methods for modifying language model representations. By replacing deterministic nodes with stochastic ones, our approach enables more robust and fine-grained control in the latent space. Through comprehensive evaluations across multiple commonsense and arithmetic reasoning benchmarks, we demonstrated that distribution-level interventions significantly improve controllability and robustness, particularly in early layers of the model. Our results suggest that incorporating distribution-aware modifications into model training could be a promising direction for improving interpretability and steering model behavior with greater precision and enable finer control.

Acknowledgment

We thank the anonymous reviewers for their valuable comments. We thank Zhengxuan Wu for the kind assistance with the ReFT codebase, which is convenient for intervention research. We also thank the members of the Chili Lab for their valuable suggestions for the work and writing.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HyxQzBceg.

Arora, A., Jurafsky, D., and Potts, C. Causalgym: Benchmarking causal interpretability methods on linguistic tasks, 2024. URL https://arxiv.org/abs/2402.12560.

Behjati, M., Fehr, F., and Henderson, J. Learning to abstract with nonparametric variational information bottleneck. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1576–1586, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.106. URL https://aclanthology.org/2023.findings-emnlp.106.

Belrose, N., Schneider-Joseph, D., Ravfogel, S., Cotterell, R., Raff, E., and Biderman, S. LEACE: perfect linear concept erasure in closed form. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Bisk, Y., Zellers, R., LeBras, R., Gao, J., and Choi, Y. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6239.

Chen, H. and Ji, Y. Learning variational word masks to improve the interpretability of neural text classifiers. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4236–4251, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.347. URL https://aclanthology.org/2020.emnlp-main.347.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Cui, G., Yuan, L., Ding, N., Yao, G., He, B., Zhu, W., Ni, Y., Xie, G., Xie, R., Lin, Y., Liu, Z., and Sun, M. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=BOorDpKHiJ.

Deng, C., Li, Z., Xie, R., Chang, R., and Chen, H. Language models are symbolic learners in arithmetic, 2024. URL https://arxiv.org/abs/2410.15580.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., and et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Fu, C., Huang, H., Chen, X., Tian, Y., and Zhao, J. Learn-to-share: A hardware-friendly transfer learning framework exploiting computation and parameter sharing. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3469–3479. PMLR,

2021. URL http://proceedings.mlr.press/v139/fu21a.html.

Gandikota, R., Materzynska, J., Zhou, T., Torralba, A., and Bau, D. Concept sliders: Lora adaptors for precise control in diffusion models, 2023. URL https://arxiv.org/abs/2311.12092.

Geiger, A., Lu, H., Icard, T., and Potts, C. Causal abstractions of neural networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 9574–9586, 2021.

Geiger, A., Wu, Z., Potts, C., Icard, T., and Goodman, N. D. Finding alignments between interpretable causal variables and distributed neural representations, 2023. URL https://arxiv.org/abs/2303.02536.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446.

Ghandeharioun, A., Caciularu, A., Pearce, A., Dixon, L., and Geva, M. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5uwBzcn885.

Gur-Arieh, Y., Mayan, R., Agassy, C., Geiger, A., and Geva, M. Enhancing automated interpretability with output-centric feature descriptions, 2025. URL https://arxiv.org/abs/2501.08319.

Han, C., Xu, J., Li, M., Fung, Y., Sun, C., Jiang, N., Abdelzaher, T., and Ji, H. Word embeddings are steers for language models, 2023. URL https://arxiv.org/abs/2305.12798.

Hanna, M., Belinkov, Y., and Pezzelle, S. When language models fall in love: Animacy processing in transformer language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12120–12135, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.744. URL https://aclanthology.org/2023.emnlp-main.744.

Hewitt, J. and Manning, C. D. A structural probe for finding syntax in word representations. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

Hosseini, M. J., Hajishirzi, H., Etzioni, O., and Kushman, N. Learning to solve arithmetic word problems with verb categorization. In Moschitti, A., Pang, B., and Daelemans, W. (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 523–533, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1058. URL https://aclanthology.org/D14-1058.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a.html.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E.-P., Bing, L., Xu, X., Poria, S., and Lee, R. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5254–5276, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.319. URL https://aclanthology.org/2023.emnlp-main.319.

Huang, J., Wu, Z., Potts, C., Geva, M., and Geiger, A. Ravel: Evaluating interpretability methods on disentangling language model representations, 2024. URL https://arxiv.org/abs/2402.17700.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR*

2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014. URL http://arxiv.org/abs/1312.6114.

Koncel-Kedziorski, R., Hajishirzi, H., Sabharwal, A., Etzioni, O., and Ang, S. D. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597, 2015. doi: 10.1162/tacl_a_00160. URL https://aclanthology.org/Q15-1042.

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., and Hajishirzi, H. MAWPS: A math word problem repository. In Knight, K., Nenkova, A., and Rambow, O. (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152–1157, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1136. URL https://aclanthology.org/N16-1136.

Lasri, K., Pimentel, T., Lenci, A., Poibeau, T., and Cotterell, R. Probing for the usage of grammatical number. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8818–8831, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.603. URL https://aclanthology.org/2022.acl-long.603.

Li, K., Patel, O., Viégas, F. B., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023a.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023b.

Li, X. L. and Eisner, J. Specializing word embeddings (for parsing) by information bottleneck. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2744–2754, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1276. URL https://aclanthology.org/D19-1276.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

Ling, W., Yogatama, D., Dyer, C., and Blunsom, P. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Barzilay, R. and Kan, M.-Y. (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015.

Liu, S., Wang, C., Yin, H., Molchanov, P., Wang, Y. F., Cheng, K., and Chen, M. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3d5CIRG1n2.

Mahabadi, R. K., Belinkov, Y., and Henderson, J. Variational information bottleneck for effective low-resource fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=kvhzKz-_DMF.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

Miller, G. A. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11,*

1994, 1994. URL https://aclanthology.org/H94-1111/.

Nanda, N., Lee, A., and Wattenberg, M. Emergent linear representations in world models of self-supervised sequence models. In Belinkov, Y., Hao, S., Jumelet, J., Kim, N., McCarthy, A., and Mohebbi, H. (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL https://aclanthology.org/2023.blackboxnlp-1.2.

Olah, C. et al. Understanding lstm networks. 2015.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=UGpGkLzwpP.

Patel, A., Bhattamishra, S., and Goyal, N. Are NLP models really able to solve simple math word problems? In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168.

Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7654–7673, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.617. URL https://aclanthology.org/2020.emnlp-main.617.

Ravfogel, S., Twiton, M., Goldberg, Y., and Cotterell, R. Linear adversarial concept erasure. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18400–18421. PMLR, 2022. URL https://proceedings.mlr.press/v162/ravfogel22a.html.

Roy, S. and Roth, D. Solving general arithmetic word problems. In Màrquez, L., Callison-Burch, C., and Su, J. (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1743–1752, Lisbon, Portugal, 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1202. URL https://aclanthology.org/D15-1202.

Rumelhart, D. E., McClelland, J. L., Group, P. R., et al. *Parallel distributed processing, volume 1: Explorations in the microstructure of cognition: Foundations*. The MIT press, 1986.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6399.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social IQa: Commonsense reasoning about social interactions. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454.

Shazeer, N. Glu variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Stolfo, A., Belinkov, Y., and Sachan, M. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7035–7052, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.435. URL https://aclanthology.org/2023.emnlp-main.435.

Subramani, N., Suresh, N., and Peters, M. Extracting latent steering vectors from pretrained language models. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.48. URL https://aclanthology.org/2022.findings-acl.48.

Tian, R., Mao, Y., and Zhang, R. Learning VAE-LDA models with rounded reparameterization trick. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1315–1325, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.101. URL https://aclanthology.org/2020.emnlp-main.101.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle, 2015. URL https://arxiv.org/abs/1503.02406.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method, 2000. URL https://arxiv.org/abs/physics/0004057.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023a. URL https://arxiv.org/abs/2302.13971.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., and et al. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL https://arxiv.org/abs/2307.09288.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=NpsVSN6o4ul.

White, A. S., Rastogi, P., Duh, K., Van, B., and Inference, D. . Improving fine-tuning on low-resource corpora with information bottleneck. 2020. URL https://api.semanticscholar.org/CorpusID:219978318.

Wu, M., Liu, W., Wang, X., Li, T., Lv, C., Ling, Z., Zhu, J., Zhang, C., Zheng, X., and Huang, X. Advancing parameter efficiency in fine-tuning via representation editing, 2024a. URL https://arxiv.org/abs/2402.15179.

Wu, Z., Geiger, A., Icard, T., Potts, C., and Goodman, N. D. Interpretability at scale: Identifying causal mechanisms in alpaca. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Wu, Z., Arora, A., Wang, Z., Geiger, A., Jurafsky, D., Manning, C. D., and Potts, C. Reft: Representation finetuning for language models. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b.

Wu, Z., Geiger, A., Arora, A., Huang, J., Wang, Z., Goodman, N., Manning, C., and Potts, C. pyvene: A library for understanding and improving PyTorch models via interventions. In Chang, K.-W., Lee, A., and Rajani, N. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pp. 158–165, Mexico City, Mexico, 2024c. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-demo.16.

Yin, F., Ye, X., and Durrett, G. Lofit: Localized fine-tuning on LLM representations. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Yu, Y., Buchanan, S., Pai, D., Chu, T., Wu, Z., Tong, S., Haeffele, B. D., and Ma, Y. White-box transformers via sparse rate reduction. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez,

L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Zhang, F. and Nanda, N. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

Zhang, R., Han, J., Liu, C., Gao, P., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., and Qiao, Y. Llama-adapter: Efficient fine-tuning of language models with zero-init attention, 2023. URL https://arxiv.org/abs/2303.16199.

Zhang, R., Qiang, R., Somayajula, S. A., and Xie, P. AutoLoRA: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5048–5060, Mexico City, Mexico, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.282.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., Goel, S., Li, N., Byun, M. J., Wang, Z., Mallen, A., Basart, S., Koyejo, S., Song, D., Fredrikson, M., Kolter, J. Z., and Hendrycks, D. Representation engineering: A top-down approach to ai transparency, 2023. URL https://arxiv.org/abs/2310.01405.

A. Data Processing Inequality

The Data Processing Inequality (DPI) is a fundamental concept in information theory that describes the loss of information through a processing pipeline. In deep neural network, the mutual information satisfies:

$$I(Y;X) \geq I(Y;Z) \tag{21}$$

where $I(\cdot;\cdot)$ denotes the mutual information between two random variables.

Intuitively, this inequality implies that processing data, represented by the transition from Z to Y, cannot increase the information about the original variable X. In other words, no transformation or operation on Z can recover information about X that has already been lost. In the context of LMs, this concept is particularly relevant. The hidden representations $Z^{(l)}$ at layer l encode information about the input sequence X. As these representations are transformed layer by layer through attention mechanisms and feed-forward networks, the DPI suggests that the mutual information between the input X and the final output Y is non-increasing as we move deeper into the network:

$$I(Y;X) \geq I(X;Z^{(L)}) \geq \cdots \geq I(X;Z^{(1)}) \geq I(Y;\hat{Y}) \tag{22}$$

This highlights a trade-off in deep architectures: while deeper layers may refine representations for specific tasks, they cannot recover information lost in earlier layers.

By the chain rule of mutual information, minimizing $\mathcal{L}_{\text{CE}}$ maximizes $I(Y;\hat{Y})$ while implicitly encouraging the learned representations $Z^{(L)}$ to retain sufficient task-relevant information about Y.

B. Hyperparameter Configuration

In this section we will discuss all the hyperparameter setting for previous method.

Hyperparameter Configuration

We investigate the following hyperparameters for our experiments:

- **Intervention Layer** (l): The specific layer in the model where the intervention is applied. This is chosen based on the architecture and the desired impact on the model's behavior.
- **Noise Scale** (ϵ): The magnitude of noise added during the intervention. This controls the level of perturbation introduced to the model's activations.
- **Subspace Rank** (r): The rank of the subspace used for the intervention. This determines the dimensionality of the subspace in which the intervention operates.
- **Intervention Position** (p): The position within the layer where the intervention is applied (e.g., before or after a specific operation like activation or normalization).
- **Batch Size** (bs): The number of samples processed in each batch during training. This affects the stability and speed of the training process.
- **Training Epochs** (e): The total number of times the model is trained over the entire dataset. This influences the convergence and generalization of the model.
- **Learning Rate** (lr): The step size at which the model's parameters are updated during training. This controls the speed and stability of the learning process.

B.1. ReFT

ReFT Hyperparameter Configuration

We investigate the following hyperparameters for our experiments:

- **Learning Rate** (lr): $9e-4$.
- **Subspace Rank** (r): 8 or 16 works best. Higher rank like 256 does not introduce the boost.
- **Intervention Position** (p): $f7+l7$: first seven token and last seven token.
- **Batch Size** (bs): 8. We also enable gradient checkpoint with accumulated steps = 4. We are unable to ablate this params due to memory constraints.
- **Training Epochs** (e): 12 works best. In our experiments decrease training epochs to 9 lead to performance drop.
- **Intervention Layer** (l): Depends on the experiment setting, but we find early layers work best.

B.2. D-ReFT

ReFT Hyperparameter Configuration

We investigate the following hyperparameters for our experiments:

- **Learning Rate** (lr): $1e-3$ or $3e-3$.
- **Subspace Rank** (r): 8 or 16 works best. Higher rank like 256 does not introduce the boost.
- **Intervention Position** (p): $f7+l7$: first seven token and last seven token.
- **Batch Size** (bs): 8. We also enable gradient checkpoint with accumulated steps = 4. We are unable to ablate this params due to memory constraints.
- **Training Epochs** (e): 9 works best, which showcases that D-ReFT also have better convergence property.
- **Intervention Layer** (l): Depends on the experiment setting, but we find early layers work best.

B.3. LoRA

ReFT Hyperparameter Configuration

We investigate the following hyperparameters for our experiments:

- **Learning Rate** (lr): $4e-4$.
- **Alpha** (α): 16.
- **Subspace Rank** (r): 16 works best.
- **Intervention Position** (p): all: all token positions are intervented.
- **Batch Size** (bs): 8. We also enable gradient checkpoint with accumulated steps = 4. We are unable to ablate this params due to memory constraints.
- **Training Epochs** (e): 6 works best.
- **Intervention Layer** (l): Depends on the experiment setting, but we find early layers work best.

B.4. RED.

RED Hyperparameter Configuration

We investigate the following hyperparameters for our experiments:

- **Learning Rate** (lr): $7e-4$.
- **Intervention Position** (p): all: all token positions are intervented.
- **Batch Size** (bs): 8. We also enable gradient checkpoint with accumulated steps = 4. We are unable to ablate this params due to memory constraints.
- **Training Epochs** (e): 9 works best.
- **Intervention Layer** (l): Depends on the experiment setting, but we find early layers work best.

B.5. SwiGLU.

SwiGLU Hyperparameter Configuration

We investigate the following hyperparameters for our experiments:

- **Learning Rate** (lr): $6e-4$.
- **Intervention Position** (p): all: all token positions are intervented.
- **Batch Size** (bs): 8. We also enable gradient checkpoint with accumulated steps = 4. We are unable to ablate this params due to memory constraints.
- **Training Epochs** (e): 9 works best.
- **Intervention Layer** (l): Depends on the experiment setting, but we find early layers work best.

B.6. MLP.

MLP Hyperparameter Configuration

We investigate the following hyperparameters for our experiments:

- **Learning Rate** (lr): $5e-4$.
- **Intervention Position** (p): all: all token positions are intervented.
- **Batch Size** (bs): 8. We also enable gradient checkpoint with accumulated steps = 4. We are unable to ablate this params due to memory constraints.
- **Training Epochs** (e): 9 works best.
- **Intervention Layer** (l): Depends on the experiment setting, but we find early layers work best.

C. Datasets.

In this section we will introduce the benchmarks we are use in this paper. Generally, we follow the setting of Hu et al. (2023); Wu et al. (2024b), using eight commonsense benchmarks and seven arithmetic benchmarks to evaluate.

C.1. Commonsense Reasoning

BoolQ. The BoolQ (Clark et al., 2019) dataset is a collection of natural language questions and corresponding passages designed for the task of binary question answering. It contains over 15,000 examples where each question can be answered with a simple "yes" or "no" based on the information provided in the accompanying passage. The dataset is derived from real user queries and web pages, making it a valuable resource for training and evaluating models on understanding context and reasoning over text. BoolQ is widely used in NLP research to benchmark the performance of models in tasks requiring comprehension, inference, and binary classification. Its challenging nature stems from the need for models to grasp nuanced relationships between questions and passages, making it a key dataset for advancing question-answering systems.

An example of data from the BoolQ dataset

Instructions: Please answer the following question with true or false, question: does ethanol take more energy make that produces?
Answer format: true/false.

PIQA. The PIQA (Bisk et al., 2020) dataset is a benchmark designed to evaluate a model's understanding of physical commonsense reasoning in everyday scenarios. It consists of questions that require reasoning about how objects interact, are used, or are manipulated in the physical world. Each question presents two possible solutions to a practical problem, and the task is to select the most appropriate one based on real-world physics and intuition. PIQA challenges models to go beyond textual knowledge and incorporate an understanding of physical properties, causality, and affordances, making it a valuable resource for advancing AI systems in tasks that require grounded, real-world reasoning.

An example of data from the PIQA dataset

Instructions: Please choose the correct solution to the question: How do I ready a guinea pig cage for it's new occupants?
Solution1: Provide the guinea pig with a cage full of a few inches of bedding made of ripped paper strips, you will also need to supply it with a water bottle and a food dish.
Solution2: Provide the guinea pig with a cage full of a few inches of bedding made of ripped jeans material, you will also need to supply it with a water bottle and a food dish.
Answer format: solution1 or solution2

HellaSwag. The HellaSwag (Zellers et al., 2019) dataset is a benchmark designed to evaluate the commonsense reasoning capabilities of natural language understanding models. Introduced in 2019, it consists of multiple-choice questions that require models to predict the most plausible continuation of a given scenario, drawing on everyday knowledge and contextual understanding. Unlike many other datasets, HellaSwag emphasizes real-world situations and nuanced reasoning, making it particularly challenging for state-of-the-art models. The dataset was created to address the limitations of previous benchmarks, which often relied on superficial patterns or biases in the data. By focusing on scenarios that require deeper comprehension and inference, HellaSwag has become a valuable tool for advancing research in artificial intelligence and improving the robustness of language models.

An example of data from the HellaSwag dataset

Instructions: Please choose the correct ending to complete the given sentence: Roof shingle removal: A man is sitting on a roof. he
Ending1: is using wrap to wrap a pair of skis.
Ending2: is ripping level tiles off.
Ending3: is holding a rubik's cube.
Ending4: starts pulling up roofing on a roof.
Answer format: ending1/ending2/ending3/ending4.

WinoGrande. The WinoGrande (Sakaguchi et al., 2020) dataset is a large-scale collection of natural language inference problems designed to evaluate the reasoning capabilities of artificial intelligence systems, particularly in the context of

commonsense reasoning. Introduced as a more challenging successor to the Winograd Schema Challenge, WinoGrande contains over 44,000 carefully crafted pronoun resolution problems that require understanding context, world knowledge, and subtle linguistic cues. Each problem presents a short passage with an ambiguous pronoun, and the task is to determine the correct referent from two possible options. To address biases and ensure robustness, the dataset was created using a crowdsourcing approach followed by a systematic adversarial filtering process. WinoGrande has become a benchmark for testing the limits of machine learning models in handling complex reasoning tasks, pushing the boundaries of AI systems toward more human-like comprehension and decision-making.

An example of data from the WinoGrande dataset

Instructions: Please choose the correct answer to fill in the blank to complete the given sentence: Sarah was a much better surgeon than Maria so always got the easier cases.
Option1: Sarah
Option2: Maria
Answer format: option1/option2

ARC-e. The ARC-e (AI2 Reasoning Challenge - Easy) (Clark et al., 2018) dataset is a collection of elementary-level science questions designed to evaluate the reasoning and comprehension capabilities of artificial intelligence systems. Developed by the Allen Institute for AI, ARC-e focuses on multiple-choice questions that require a fundamental understanding of scientific concepts, making it an accessible yet challenging benchmark for AI models. Unlike its more advanced counterpart, ARC (AI2 Reasoning Challenge), ARC-e is tailored to assess basic knowledge and straightforward reasoning, often drawing from topics taught in early education. By providing a simplified yet diverse set of questions, ARC-e serves as a valuable tool for testing the foundational abilities of AI systems in processing and answering science-related queries, paving the way for more complex reasoning tasks.

An example of data from the ARC-e dataset

Instructions: Please choose the correct answer to the question: Which statement best explains why photosynthesis is the foundation of most food webs?
Answer1: Sunlight is the source of energy for nearly all ecosystems.
Answer2: Most ecosystems are found on land instead of in water.
Answer3: Carbon dioxide is more available than other gases.
Answer4: The producers in all ecosystems are plants.
Answer format: answer1/answer2/answer3/answer4

ARC-c. The ARC-c (Clark et al., 2018) dataset, part of the AI2 Reasoning Challenge (ARC), is a comprehensive collection of science questions designed to evaluate the reasoning and comprehension capabilities of artificial intelligence systems. Comprising multiple-choice questions from various grade levels, the dataset emphasizes complex reasoning, requiring models to go beyond simple retrieval and engage in deeper understanding and inference. The questions are drawn from diverse scientific domains, including biology, chemistry, physics, and earth science, making it a robust benchmark for assessing the generalization and problem-solving skills of AI. By focusing on challenging, curriculum-aligned content, the ARC-c dataset serves as a critical tool for advancing the development of AI systems capable of nuanced and context-aware reasoning.

An example of data from the ARC-c dataset

Instructions: Please choose the correct answer to the question: A group of engineers wanted to know how different building designs would respond during an earthquake. They made several models of buildings and tested each for its ability to withstand earthquake conditions. Which will most likely result from testing different building designs?
Answer1: buildings will be built faster
Answer2: buildings will be made safer
Answer3: building designs will look nicer
Answer4: building materials will be cheaper
Answer format: answer1/answer2/answer3/answer4

OBQA. The Open Book Question Answering (OBQA) (Mihaylov et al., 2018) dataset is a benchmark designed to evaluate the ability of machine learning models to answer science-based questions by combining open-book fact retrieval with reasoning skills. Unlike traditional QA datasets, OBQA requires systems to not only retrieve relevant information from a provided knowledge source but also apply logical reasoning to infer the correct answer. The dataset consists of multiple-choice questions that span a wide range of scientific topics, challenging models to demonstrate both comprehension and analytical capabilities. By emphasizing the integration of external knowledge and reasoning, OBQA serves as a valuable tool for advancing research in question answering, knowledge representation, and artificial intelligence.

An example of data from the OBQA dataset

Instructions: Please choose the correct answer to the question: A person wants to start saving money so that they can afford a nice vacation at the end of the year. After looking over their budget and expenses, they decide the best way to save money is to
Answer1: make more phone calls
Answer2: quit eating lunch out
Answer3: buy less with monopoly money
Answer4: have lunch with friends
Answer format: answer1/answer2/answer3/answer4

C.2. Arithmetic Reasoning

AQuA. The AQuA (Algebraic Question Answering) (Ling et al., 2017) dataset is a specialized resource designed to advance research in the field of mathematical reasoning and natural language processing. It consists of a collection of algebra word problems, each paired with multiple-choice answers and detailed step-by-step solutions. The dataset is particularly valuable for training and evaluating models that aim to understand and solve complex mathematical problems expressed in natural language. By providing both the problem statements and their corresponding solutions, AQuA enables researchers to develop and test algorithms that can not only generate correct answers but also explain the reasoning process behind them. This makes AQuA an essential tool for exploring the intersection of mathematics and artificial intelligence, fostering advancements in automated problem-solving and educational technologies.

An example of data from the AQuA dataset

Instructions: A rectangular piece of 150 sq m has a length which is 1m more than the 4 times the breadth. What is the perimeter of the piece? Answer Choices: (A) 60 m (B) 61 m (C) 62 m (D) 63 m (E) 64 m
Let's assume the breadth of the rectangle to be x. According to the problem, the length of the rectangle is 1 m more than 4 times the breadth. Therefore, the length of the rectangle is $4x + 1$.
We know that the area of the rectangle is 150 sq m. Therefore, we can write:

$$\text{Area} = \text{Length} \times \text{Breadth}$$

$$150 = (4x + 1) \times x$$

Simplifying this equation, we get:

$$150 = 4x^2 + x$$

$$150 - x = 4x^2$$

$$x^2 - \frac{1}{4}x - \frac{15}{4} = 0$$

Solving for x, we get:

$$x = 1.732 \quad \text{or} \quad x = -0.732$$

Since the breadth cannot be negative, we can ignore the negative value of x. Therefore, the breadth of the rectangle is 1.732 m.
Now, we can find the length of the rectangle:

$$\text{Length} = 4x + 1 = 4(1.732) + 1 = 7.148\,\text{m}$$

The perimeter of the rectangle is the sum of the length of all four sides. Therefore, the perimeter is:

$$\text{Perimeter} = 2(\text{Length} + \text{Breadth}) = 2(7.148 + 1.732) = 16.928\,\text{m}$$

Rounding off to the nearest integer, we get the answer as 17. Therefore, the answer is $\boxed{(E)\,64\,\text{m}}$.

AddSub. The AddSub (Hosseini et al., 2014) dataset is a widely-used benchmark in NLP designed to evaluate the ability of models to solve arithmetic word problems. It consists of pairs of questions and answers, where each question is a textual description of a mathematical problem involving addition or subtraction, and the corresponding answer is the numerical result. This dataset challenges models to not only understand the linguistic nuances of the problem but also to perform the necessary calculations accurately. By focusing on basic arithmetic operations, AddSub serves as a fundamental testbed for assessing the reasoning and comprehension capabilities of NLP systems, making it a valuable resource for research in machine learning and artificial intelligence.

An example of data from the AddSub dataset

Instructions: There are 7 crayons in the drawer . Mary took 3 crayons out of the drawer. How many crayons are there now?
Step 1: Start with the total number of crayons in the drawer: 7
Step 2: Subtract the number of crayons Mary took out: 3
Step 3: Perform the subtraction: 7 - 3 = 4
Answer: There are now 4 crayons in the drawer.

GSM8K. The GSM8K (Cobbe et al., 2021) dataset is a collection of 8.5K high-quality, linguistically diverse grade school math word problems designed to benchmark the problem-solving abilities of AI systems. Each problem in the dataset requires multiple reasoning steps to arrive at the correct solution, making it a valuable resource for evaluating the mathematical and logical reasoning capabilities of language models. The problems are crafted to reflect real-world scenarios, ensuring that they are both challenging and accessible to students at the elementary level. By providing a diverse set of

problems that span various mathematical concepts, GSM8K serves as a robust testbed for advancing the development of AI systems capable of understanding and solving complex, multi-step problems.

An example of data from the GSM8K dataset

Instructions: Mr Boarden is remodeling his bathroom. For every square foot, he needs 24 mosaic tiles. How many mosaic tiles would Mr Boarden need to cover two thirds of his 36 sq ft bathroom?
Step 1: Find the total area of the bathroom: 36 sq ft
Step 2: Calculate the area to be covered by the mosaic tiles
36 sq ft x 2/3 = 24 sq ft
Step 3: Calculate the number of mosaic tiles needed
24 sq ft / 1 sq ft per 24 tiles = 1 tile
Therefore, Mr Boarden would need 1 mosaic tile to cover two thirds of his 36 sq ft bathroom.
The answer in Arabic numerals is 1.

MAWPS. The MAWPS (Math Word Problem Solving) (Koncel-Kedziorski et al., 2016), is a comprehensive collection of mathematical word problems designed to facilitate research in NLP. This dataset encompasses a wide range of problem types, including arithmetic, algebra, and geometry, presented in natural language form, making it a valuable resource for developing and evaluating algorithms that aim to solve math problems through automated reasoning and language understanding. MAWPS is particularly useful for training and testing models that require both linguistic comprehension and mathematical problem-solving skills, bridging the gap between textual data and quantitative reasoning. By providing a diverse set of problems with varying levels of complexity, MAWPS serves as a benchmark for advancing the capabilities of AI systems in educational applications and beyond.

An example of data from the MAWPS dataset

Instructions: Nicole found an orange caterpillar and a green caterpillar in her backyard . The green caterpillar was 3 inches long and the orange caterpillar was 1.1666666666666667 inches long . How much longer was the green caterpillar than the orange caterpillar ?
To find the difference in length between the green and orange caterpillars, we need to subtract the length of the orange caterpillar from the length of the green caterpillar.
Length of green caterpillar - Length of orange caterpillar = Difference in length
3 inches - 1.1666666666666667 inches = 1.8333333333333333 inches
Therefore, the green caterpillar was 1.8333333333333333 inches longer than the orange caterpillar.
The answer in Arabic numerals is 1.8333333333333333

MultiArith. The MultiArith (Roy & Roth, 2015) dataset is a benchmark collection designed to evaluate the performance of machine learning models, particularly in solving arithmetic word problems that require multiple steps of reasoning. This dataset consists of a variety of mathematical problems presented in natural language, challenging models to interpret the text, extract relevant numerical information, and perform sequential arithmetic operations to arrive at the correct solution. By encompassing problems that involve addition, subtraction, multiplication, and division, often in combination, MultiArith tests both the linguistic comprehension and mathematical reasoning capabilities of AI systems. It serves as a valuable resource for advancing research in NLP, particularly in bridging the gap between language understanding and quantitative problem-solving.

An example of data from the MultiArith dataset

Instructions: At the schools book fair Sam bought 13 adventure books and 17 mystery books. If 15 of the books were used, how many new books did he buy?
To solve the problem, we need to find the total number of books Sam bought and then subtract the number of used books from that total.
1. Find the total number of books Sam bought:
13 adventure books + 17 mystery books = 30 books
2. Subtract the number of used books from the total:
30 books - 15 used books = 15 new books
Therefore, Sam bought 15 new books.
The answer in Arabic numerals is 15.

SingleEq. The SingleEq (Koncel-Kedziorski et al., 2015) dataset is a specialized resource designed to support research and development in the field of mathematical problem-solving, particularly focusing on single-variable linear equations. This dataset comprises a collection of problems, each involving the formulation and solution of linear equations with one unknown, making it an invaluable tool for training and evaluating machine learning models, educational software, and automated tutoring systems. By providing a structured and diverse set of equations, the SingleEq dataset enables researchers to explore various computational approaches, from symbolic reasoning to neural network-based methods, aiming to enhance the accuracy and efficiency of automated equation-solving technologies. Its comprehensive nature ensures that models trained on this dataset can handle a wide range of equation types, promoting robustness and generalizability in mathematical problem-solving applications.

An example of data from the SingleEq dataset

Instructions: Oceanside Bike Rental Shop charges 17 dollars plus 7 dollars an hour for renting a bike. Tom paid 80 dollars to rent a bike. How many hours did he pay to have the bike checked out ?
Let's assume that Tom had the bike checked out for ẍ̈hours.
According to the problem, the cost of renting a bike is 17 dollars plus 7 dollars per hour. So, the total cost for x hours can be represented as:
17 + 7x
We know that Tom paid 80 dollars to rent the bike, so we can set up an equation:
17 + 7x = 80
Subtracting 17 from both sides, we get:
7x = 63
Dividing both sides by 7, we get:
x = 9
Therefore, Tom paid to have the bike checked out for 9 hours.

SVAMP. The SVAMP (Patel et al., 2021) (Simple Variations in Arithmetic Word Problems) dataset is a carefully curated collection of arithmetic word problems designed to evaluate and enhance the problem-solving capabilities of machine learning models. Unlike traditional datasets, SVAMP introduces variations in problem structure, wording, and complexity to test the robustness and generalization of models across diverse scenarios. Each problem is crafted to require a combination of mathematical reasoning and natural language understanding, making it a valuable benchmark for assessing the performance of AI systems in real-world applications. By incorporating a wide range of problem types, SVAMP aims to bridge the gap between simple arithmetic tasks and more complex, context-rich challenges, providing a comprehensive tool for advancing research in mathematical reasoning and natural language processing.

An example of data from the SVAMP dataset

Instructions: Matthew gave equal numbers of crackers and cakes to his 4 friends. If he had 32 crackers and 98 cakes initially. How many crackers did each person eat?
To solve the problem, we need to divide the total number of crackers by the number of friends.
Step 1: Find the total number of crackers by adding them up.
32 crackers
Step 2: Divide the total number of crackers by the number of friends.
32 crackers ÷ 4 friends = 8 crackers per person
Therefore, each person ate 8 crackers.
Answer: 8