# SAE Feature Intervention Mathematical Description

## Overview

The feature intervention operation modifies sparse auto-encoder (SAE) latent representations to amplify specific features while suppressing others. This technique enables causal analysis of feature importance in downstream tasks.

## Mathematical Formulation

### Input Variables

Let: - $\mathbf{z} \in \mathbb{R}^{n \times d}$ be the original latent representation from the SAE - $f \in \{1, 2, \ldots, d\}$ be the target feature index - $\alpha \geq 0$ be the activation factor - $\tau \geq 0$ be the minimum activation threshold

Where $n$ is the sequence length and $d$ is the latent dimension.

### Intervention Operation

The intervention operation consists of three sequential steps:

**1. Threshold Clamping** First, we ensure the target feature meets a minimum activation threshold:

$$\mathbf{z}_{:,f} = \max(\mathbf{z}_{:,f}, \tau)$$

Where $\mathbf{z}_{:,f}$ denotes all elements in column $f$ (the target feature across all sequence positions).

**2. Intervention Vector Construction** We construct an intervention vector $\mathbf{v} \in \mathbb{R}^{n \times d}$ with elements:

$$v_{i,j} = \begin{cases} \alpha & \text{if } j = f \\ \frac{1}{\alpha} & \text{if } j \neq f \text{ and } \alpha \neq 0 \\ 0 & \text{if } j \neq f \text{ and } \alpha = 0 \end{cases}$$

This creates a multiplicative mask that amplifies the target feature by factor $\alpha$ while suppressing all other features by factor $\frac{1}{\alpha}$.

**3. Element-wise Multiplication** The final modified latent representation is obtained via:

$$\mathbf{z}' = \mathbf{z} \odot \mathbf{v}$$

Where $\odot$ denotes element-wise (Hadamard) multiplication.

### Complete Transformation

Combining all steps, the complete intervention transformation can be written as:

$$\mathbf{z}'_{i,j} = \begin{cases} \max(z_{i,f}, \tau) \cdot \alpha & \text{if } j = f \\ z_{i,j} \cdot \frac{1}{\alpha} & \text{if } j \neq f \text{ and } \alpha \neq 0 \\ 0 & \text{if } j \neq f \text{ and } \alpha = 0 \end{cases}$$

## Properties and Interpretation

### Feature Amplification

When $\alpha > 1$, the target feature $f$ is amplified, making it more influential in downstream processing.

### Feature Suppression

When $\alpha > 1$, all non-target features are suppressed by factor $\frac{1}{\alpha}$, reducing their influence.

### Ablation Mode

When $\alpha = 0$, all non-target features are completely ablated (set to zero), creating a pure feature isolation experiment.

### Identity Transformation

When $\alpha = 1$, the operation reduces to identity after threshold clamping: $\mathbf{z}' = \mathbf{z}$ (except for the clamping effect on feature $f$).

## Implementation Notes

In the codebase implementation: - `latent` corresponds to $\mathbf{z}$ - `feature` corresponds to $f$ - `act_factor` corresponds to $\alpha$ - `feat_min` corresponds to $\tau$ - `intervention_vec` corresponds to $\mathbf{v}$ - `modified_latent` corresponds to $\mathbf{z}'$

## Applications

This intervention technique enables:

1. **Causal Feature Analysis**: Determining which SAE features causally influence model predictions
2. **Feature Importance Ranking**: Comparing the effect sizes of different features

3. **Mechanistic Interpretability**: Understanding how individual learned features contribute to model behavior
4. **Ablation Studies**: Isolating the contribution of specific features by suppressing others

## Example Parameter Values

Common parameter configurations used in the project:

- **Strong Amplification**: $\alpha = 10.0$, $\tau = 0.1$
- **Feature Isolation**: $\alpha = 0.0$, $\tau = 10.0$
- **Mild Enhancement**: $\alpha = 2.0$, $\tau = 0.1$

These configurations allow researchers to probe different aspects of feature functionality and importance in the genomic language model's learned representations.