Activation Projection Explainer

How to read the 2D scatter plots in the activation viewer — from raw residual stream to the images on screen.

1. What comes out of the model

The dataset contains N decision-theoretic prompts, each with a **pre-written EDT completion** and a **pre-written CDT completion**. The model never answers freely — instead, we run Qwen3-32B (4-bit) on each prompt **twice**, once with the EDT completion appended as if the assistant said it, and once with the CDT completion. This is a **contrastive activation** approach (sometimes called representation engineering): by forcing matched pairs of completions that differ only in their decision-theoretic stance, we isolate the model's internal representation of "EDT-style reasoning" vs "CDT-style reasoning," controlling for confounds like answer length or topic.

At the last token position of each forced completion, we extract the **residual stream vector** from selected layers.

Each residual stream vector lives in $\mathbb{R}^{5120}$ (the model's hidden dimension). So for N contrastive prompts, we get two matrices per layer:

$$\mathbf{E} \in \mathbb{R}^{N \times 5120} \quad \text{(EDT activations)} \mathbf{C} \in \mathbb{R}^{N \times 5120} \quad \text{(CDT activations)}$$

The subscript notation: $\mathbf{e}_i$ is the 5120-dim activation for prompt i with the EDT completion, $\mathbf{c}_i$ for CDT.

---

2. Projection Mode 1: Standard PCA

What it shows: The two directions of greatest overall variance in the activation space, agnostic to the EDT/CDT labels.

Steps

1. **Stack** all activations into one matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{E} \\ \mathbf{C} \end{bmatrix} \in \mathbb{R}^{2N \times 5120}$$

2. **Center** by subtracting the column mean:

$$\bar{\mathbf{x}} = \frac{1}{2N} \sum_{i=1}^{2N} \mathbf{x}_i, \qquad \tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^\top$$

3. **Compute the covariance matrix** (conceptually — sklearn does this via SVD):

$$\mathbf{\Sigma} = \frac{1}{2N-1} \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \in \mathbb{R}^{5120 \times 5120}$$

4. **Extract eigenvectors.** The top two eigenvectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{5120}$ (sorted by eigenvalue $\lambda_1 \geq \lambda_2$) define PC1 and PC2.

5. **Project** each point:

$$\text{PC1}_i = \tilde{\mathbf{x}}_i \cdot \mathbf{v}_1, \qquad \text{PC2}_i = \tilde{\mathbf{x}}_i \cdot \mathbf{v}_2$$

These are the (x, y) coordinates you see on the plot.

How to read it

- **Variance explained** (shown in the UI) tells you what fraction of total variance each axis captures. If PC1 explains 15% and PC2 explains 8%, together they show 23% of the 5120-dimensional structure — the rest is lost in projection.
- **Cluster separation** between EDT (blue) and CDT (red) points means the EDT/CDT distinction aligns with the dominant variance directions. If the two clouds overlap completely, the EDT/CDT signal is orthogonal to the main variance axes (it exists but PCA can't see it).
- **Spread along an axis** = high variance in that direction. Tight clusters mean most prompts land in similar regions of activation space.

Orientation convention

The code flips PC1 so the EDT centroid has positive x-coordinate. This is cosmetic — PCA eigenvectors are sign-ambiguous (if $\mathbf{v}$ is an eigenvector, so is $-\mathbf{v}$).

Key limitation

PCA finds directions of **maximum total variance**, not maximum **class separation**. The biggest source of variance might be prompt length, scenario topic, or token identity — not EDT vs CDT. If the EDT/CDT signal is small relative to other variance sources, PCA will miss it entirely, even if a linear probe can find it easily. This is why we also have Mode 2.

---

3. Projection Mode 2: Probe (Contrastive Direction)

What it shows: The x-axis is explicitly the EDT-vs-CDT separation direction. The y-axis is the next most interesting direction *after removing* that signal.

Steps

1. **Compute the contrastive direction** (the "steering vector"):

$$\boldsymbol{\delta} = \bar{\mathbf{e}} - \bar{\mathbf{c}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{e}_i - \frac{1}{N}\sum_{i=1}^{N} \mathbf{c}_i$$

Then normalize:

$$\hat{\boldsymbol{\delta}} = \frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|}$$

This is a single unit vector in $\mathbb{R}^{5120}$ pointing from the CDT centroid toward the EDT centroid.

2. **Center the data** (same as PCA):

$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^\top$$

3. **Project onto the contrastive direction** (this becomes the x-axis):

$$p_i = \tilde{\mathbf{x}}_i \cdot \hat{\boldsymbol{\delta}}$$

A positive p_i means that point is on the EDT side of the mean; negative means CDT side.

4. **Remove the contrastive component** to get the residual:

$$\mathbf{r}_i = \tilde{\mathbf{x}}_i - p_i \, \hat{\boldsymbol{\delta}}$$

This strips out the EDT/CDT axis entirely. What's left is the 5119-dimensional subspace orthogonal to $\hat{\boldsymbol{\delta}}$.

5. **PCA on the residual** — fit PCA with 1 component on the residual matrix $\mathbf{R}$:

$$q_i = \mathbf{r}_i \cdot \mathbf{u}_1$$

where $\mathbf{u}_1$ is the top eigenvector of the residual's covariance matrix.

6. **Plot** (p_i, q_i) for each point.

How to read it

- **X-axis (EDT–CDT direction):** This is the axis that *maximally separates the two classes by their means*. If you see clean left/right separation, the model's internal representations genuinely differ between EDT and CDT completions at this layer.
- **Y-axis (1st orthogonal PC):** This is the biggest remaining structure *after* you subtract the EDT/CDT signal. It often captures prompt-level variation (different scenarios landing in different places), or it might reveal sub-clusters within EDT or CDT.
- **Centroid separation** ($\|\boldsymbol{\delta}\|$, shown in the metadata): The L2 distance between EDT and CDT means in the full 5120-dim space. Larger = the model represents EDT and CDT completions more differently at this layer. This number naturally varies across layers.
- **Variance explained:** The two numbers show what fraction of total variance each axis captures. The contrastive direction might explain very little total variance (e.g., 0.5%) but still perfectly separate the classes — because class separation and variance are different things.

Why this is more informative than PCA

PCA asks: "where is the most variance?" Probe mode asks: "where is the EDT/CDT distinction?" These are different questions. A direction can perfectly separate EDT from CDT while explaining $<$ 1% of total variance — PCA would never surface it.

Formally: PCA maximizes $\mathrm{Var}(\mathbf{Xv})$ over unit vectors $\mathbf{v}$. The contrastive direction maximizes $|\bar{\mathbf{e}} \cdot \mathbf{v} - \bar{\mathbf{c}} \cdot \mathbf{v}|$. These objectives only align when the class difference *is* the dominant variance source.

---

4. The distance coloring (opacity)

Each point also carries a **distance-from-centroid** value used for opacity:

$$d_i^{\text{EDT}} = \|\mathbf{e}_i - \bar{\mathbf{e}}\|_2, \qquad d_i^{\text{CDT}} = \|\mathbf{c}_i - \bar{\mathbf{c}}\|_2$$

These are computed in the **full 5120-dim space** (not in the projected 2D), then normalized by dividing by the max distance across all points. Points far from their class centroid in high-dimensional space appear more transparent — they're "outlier" activations. A point can look close to the centroid in 2D but be far in the full space, or vice versa.

---

5. Layer-by-layer interpretation

The viewer lets you scrub through layers: **0, 8, 16, 24, 32, 40, 48, 56, 63, post_norm**.

What changes across layers

In a transformer, each layer's residual stream is the cumulative sum of all previous attention and MLP outputs:

$$\mathbf{h}^{(\ell)} = \mathbf{h}^{(0)} + \sum_{k=1}^{\ell} \Big(\text{Attn}_k(\mathbf{h}^{(k-1)}) + \text{MLP}_k(\ldots) \Big)$$

So layer 0 is just the token embedding — it carries no contextual information about EDT vs CDT reasoning. As you move through layers:

- **Early layers (0–16):** You might expect heavy overlap, but in practice the probe can already classify well above chance even at layer 0. This is a **surface-level artifact**: the EDT and CDT completions are different text, so the last token is often a different token entirely. The layer-0 probe is learning "which tokens tend to end EDT completions vs CDT completions," not anything about reasoning. The tell: centroid separation ($\|\boldsymbol{\delta}\|$) is tiny at layer 0 (e.g., 0.4) despite above-chance probe accuracy, and the val/test accuracy gap is large (overfitting to token coincidences).
- **Middle layers (24–40):** The model is building up abstract representations. Watch for centroid separation growing — this indicates the model is constructing a robust EDT/CDT distinction beyond token-level artifacts.
- **Late layers (48–63):** These representations are closest to the output. Centroid separation peaks here (e.g., 144.7 at layer 63), indicating the model has built a strong, high-magnitude distinction between EDT and CDT completions. Separation here may reflect the model preparing different next-token predictions for EDT vs CDT completions.
- **Post-norm:** The residual stream after the final RMSNorm, which is the last representation before the unembedding matrix converts it to logits. RMSNorm rescales each vector:

$$\text{RMSNorm}(\mathbf{x}) = \frac{\mathbf{x}}{\sqrt{\frac{1}{d}\sum x_i^2}} \odot \boldsymbol{\gamma}$$

This forces all vectors to approximately the same magnitude, **destroying magnitude information while preserving directional information**. As a result, centroid separation drops dramatically (e.g., 144.7 at layer 63 to 9.1 at post-norm) and the centroids collapse into the point cloud, but probe accuracy remains high because the EDT and CDT points still *point in different directions*. This makes functional sense: the unembedding matrix maps directions to token probabilities, so RMSNorm strips out magnitude to give the unembedding clean directional signal.

What to look for

Pattern	Interpretation
Probe accuracy above chance at layer 0 but low centroid separation	Surface artifact: probe is classifying based on token identity, not reasoning
Centroid separation grows across layers	The model is building an increasingly robust EDT/CDT representation
Separation appears in PCA mode too	The EDT/CDT distinction is a *dominant* feature of the representation, not just a subtle linear direction
Separation in probe mode but NOT in PCA	The EDT/CDT signal exists but is a small-variance direction — it's real but subtle
Centroid separation peaks at late layers then drops at post-norm	Normal: RMSNorm compresses magnitudes. Check that probe accuracy stays high — if so, directional signal is preserved
Centroids collapse into the point cloud at post-norm	RMSNorm has pulled all vectors to similar magnitudes, shrinking all pairwise distances including centroid-to-point distances

---

6. Relationship to probe accuracy

The probe results (shown in the UI per layer) come from training a **logistic regression** on the full 5120-dim activations:

$$P(\text{EDT} \mid \mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

where σ is the sigmoid and $\mathbf{w} \in \mathbb{R}^{5120}$ is learned. The probe has access to all 5120 dimensions, while the scatter plot only shows 2. So:

- **Probe accuracy $\approx$ 50%:** no linear separation exists at this layer (in any direction)
- **Probe accuracy high but 2D plot overlaps:** separation exists but in directions not captured by the 2 axes shown (especially possible in PCA mode)
- **Probe accuracy high AND 2D separation visible:** the displayed axes capture the separating structure well

---

7. Summary of the full pipeline

$$\boxed{\begin{aligned}&\text{Prompt + EDT completion} \xrightarrow{\text{forward pass}} \mathbf{h}_{\text{EDT}} \in \mathbb{R}^{5120} \\ &\text{Prompt + CDT completion} \xrightarrow{\text{forward pass}} \mathbf{h}_{\text{CDT}} \in \mathbb{R}^{5120}\end{aligned}}$$

PCA mode: Stack $[\mathbf{E}; \mathbf{C}]$, center, SVD to get $\mathbf{v}_1, \mathbf{v}_2$, project to get (PC1, PC2) coordinates.

Probe mode: Compute $\boldsymbol{\delta} = \text{mean}(\mathbf{E}) - \text{mean}(\mathbf{C})$, normalize to $\hat{\boldsymbol{\delta}}$, project $\mathbf{x}$ onto $\hat{\boldsymbol{\delta}}$ for the x-axis (p), subtract to get residual $\mathbf{r} = \mathbf{x} - p\hat{\boldsymbol{\delta}}$, PCA on $\mathbf{r}$ for the y-axis (q).