Interpretable Company Similarity with Sparse Autoencoders[*]

Marco Molinari[†, 1] and **Victor Shao**[†, 1] and **Luca Imeneo**[2]
and **Mateusz Mikolajczak**[1] and **Vladimir Tregubiak**[1]
and **Abhimanyu Pandey**[1] and **Sebastião Kuznetsov Ryder Torres Pereira**[1]

[1] LSE.AI, London School of Economics
[2] Tower Research Capital

Correspondence: m.molinari1@lse.ac.uk[*]
[†] Equal contribution

Abstract

Determining company similarity is a vital task in finance, underpinning risk management, hedging, and portfolio diversification. Practitioners often rely on sector and industry classifications such as SIC and GICS codes to gauge similarity, the former is used by the U.S. Securities and Exchange Commission (SEC), and the latter widely used by the investment community. Since these classifications lack granularity and need regular updating, using clusters of embeddings of company descriptions has been proposed as a potential alternative, but the lack of interpretability in token embeddings poses a significant barrier to adoption in high-stakes contexts. Sparse Autoencoders (SAEs) have shown promise in enhancing the interpretability of Large Language Models (LLMs) by decomposing Large Language Model (LLM) activations into interpretable features. Moreover, SAEs capture an LLM's internal representation of a company description, as opposed to semantic similarity alone, as is the case with embeddings. We apply SAEs to company descriptions, and obtain meaningful clusters of equities. We benchmark SAE features against SIC-codes, Industry codes, and Embeddings. Our results demonstrate that SAE features surpass sector classifications and embeddings in capturing fundamental company characteristics. This is evidenced by their superior performance in correlating logged monthly returns – a proxy for similarity – and generating higher Sharpe ratios in co-integration trading strategies, which underscores deeper fundamental similarities among companies. Finally, we verify the interpretability of our clusters, and demonstrate that sparse features form simple and interpretable explanations for our clusters.

1 Introduction

Accurately assessing the similarity of companies is an integral task in finance, key to risk management, portfolio diversification and more (Delphini et al., 2019; Katselas et al., 2017). Hedging, a practice that relies on converse investments in related assets, is a prominent example of a financial strategy that requires a detailed understanding of the similarity between two companies.

Traditionally, company comparisons rely on (1) relative returns and (2) discrete classifications, or a combination of both[1]. For the former, relying on relative return spreads can be effective but is not foolproof, as market volatility, economic changes, fundamental changes in business, and temporal factors can alter them (Loretan and English, 2000). For the latter, discrete classification systems such as GICS[1] are limited, as the restricted granularity of a discrete classification system limits dynamic interpretations of companies' operations, in that they fail to account for the duality of certain companies [2] (Winton, 2018).

This is particularly important for pairs trading, a market-neutral strategy based on mean-reverting return spreads (Ehrman, 2012). Employing a pair-trading strategy with fundamentally similar companies whose returns are co-integrated[3] could reduce the risk of deviation from historical trends (Raghava and Bharadwaj, 2014).

Clustering embeddings of company descriptions has been proposed as a measure of similarity (Vamvourellis et al., 2023; Buchner et al., 2024), but token embeddings are not interpretable, and

---
[1]E.g. SIC-codes (U.S. Occupational Safety and Health Administration, 2001), and the Global Industry Classification System (GICS), which categorizes companies into 11 sectors and 163 sub-industries (MSCI, 2020).

[2]Emerging industries disproportionally exhibit this.

[3]Co-integration refers to a statistical property where two or more non-stationary time series variables, despite individual trends, exhibit a stationary linear combination, indicating a long-term equilibrium relationship (Engle and Granger, 1987).

1027

this leads to uncertainty, which is undesirable in the financial sector.

SAEs have the potential to provide an efficient measure of company similarity by decomposing large amounts of financial data into interpretable features (Chen et al., 2020). SAEs have recently been applied to LLMs resulting in interpretable decompositions of neural activations (Huben et al., 2024). Furthermore, SAEs can be applied at a Language Model (LM)'s deeper layers, and hence decompose a LM's internal representation of a company description, which means Sparse Autoencoder (SAE) features capture more abstract and cross-token concepts than raw embeddings (Templeton et al., 2024). This motivates their application to textual company descriptions.

To the best of our knowledge, we are the first to compute company similarity using SAEs on SEC[4] filings, and to show that SAEs can surpass existing alternatives on identifying similar companies despite the sparsity (interpretability) constraint. This is relevant since the competitiveness of SAEs has been called into question (Kantamneni et al., 2025) when compared with existing benchmarks of downstream performances.

Our contributions can be summarized as follows:

- We apply an open source SAE (EleutherAI, 2024) to Llama 3.1 8B (Grattafiori et al., 2024), and release a dataset containing company descriptions, extracted features, and returns, to support further research.[*]

- We demonstrate that clustering using sparse features outperforms embeddings and SIC/GISC codes (MSCI, 2020) in terms of intra-cluster pairwise correlations.

- We confirm the interpretability of our clusters by verifying that our explanations use a small number of highly interpretable features.

2 Related Works

2.1 Sparse autoencoders

The Linear Representation Hypothesis posits that LLMs linearly represent concepts in neuron activations (Park et al., 2024). However, as neuron activations are notoriously superpositioned (Elhage et al., 2022), SAEs enhance the interpretability of LLMs by writing neuron activations as a linear combination of sparse features (Bricken et al.,

2023). This reduces superposition and restores interpretability (Huben et al., 2024). SAEs have recently been applied both in the mechanistic interpretability of LLMs (Nanda et al., 2023; Conmy et al., 2023; Marks et al., 2024), and in deep learning more broadly (Chen and Guo, 2023). SAEs have been scaled to medium and large Language Models (LMs), such as GPT4 (Templeton et al., 2024; Gao et al., 2024).

SAEs learn a reconstruction $\hat{\mathbf{x}}$ as a sparse linear combination of features $\mathbf{y}_i \in \mathbb{R}^{d_s}$ for a given input activation $\mathbf{x} \in \mathbb{R}^{d_m}$ where d_m is the LLM's hidden size and:

$$d_s = k\, d_m, \quad \text{with } k \in \{2^n \mid n \in \mathbb{N}_+\}. \quad (1)$$

The decoder element of the SAE is given as:

$$(\hat{\mathbf{x}} \circ \mathbf{f})(\mathbf{x}) = \mathbf{b}_d + \mathbf{W}_d \mathbf{f}(\mathbf{x}) \quad (2)$$

where $\mathbf{b}_d \in \mathbb{R}^{d_m}$ is the bias term of the decoder, $\mathbf{W}_d$ is the decoder matrix with columns $\mathbf{v}_i \in \mathbb{R}^{d_m}$, and $\mathbf{f}(\mathbf{x})$ denotes the feature activations, which are described by:

$$\mathbf{f}(\mathbf{x}) = \text{TopK}(\mathbf{W}_e(\mathbf{x} - \mathbf{b}_d) + \mathbf{b}_e) \quad (3)$$

where $\mathbf{b}_e \in \mathbb{R}^{d_s}$ is the bias term of the encoder, $\mathbf{W}_e$ is the encoder matrix with columns $\mathbf{w}_i \in \mathbb{R}^{d_s}$, and the TopK activation function enforces sparsity following Gao et al. (2024).The loss function is the output's mean-squared error (MSE):

$$\mathcal{L} = \left\| \mathbf{x} - \hat{\mathbf{x}}_2^2 \right\| \quad (4)$$

2.1.1 Embedders

As a baseline, we replicate the embedding methodology of Vamvourellis et al. (2023), and obtain embeddings for company descriptions. In particular, we use their three best performing embedders for our evaluations and downstream tasks:

1. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2019).

2. Sentence-BERT (SBERT): Building on BERT, SBERT improves latency substantially (Reimers and Gurevych, 2019) and encodes meaning on the more abstract sentence level.

3. PaLM-gecko: Pathways Language Model (PaLM) (Chowdhery, 2022).

---

[4]Securities and Exchange Commission

3 Methodology

3.1 Dataset

Publicly listed companies in the US submit annual reports to the SEC, which include information on a company's operations, such as product specifications, subsidiaries, competition, and other financial details (SEC, 2023). Due to the closed-source nature of GICS classifications, we use SIC-codes and the industry/major division categorization[5] (BISC). Next, we tokenize company descriptions and preprocess them (Appendix A), resulting in a final dataset of 27,888 reports from 1996 to 2020.

3.2 Feature summing

In this work, we face the challenge of comparing sparse feature sequences of arbitrary lengths, where best practices are not well-established, though maxpooling has been proposed as a baseline for feature aggregation (Bricken et al., 2024). However, motivated by the specific demands of financial sequence modeling, we propose an alternative, employing sparse feature summing across tokens. This method provides a magnitude-scaled count of the frequency with which a feature appears within a sequence, reflecting both the number of tokens on which a feature is active and its intensity (Lan et al., 2024).

Our approach is inspired by analogous methodologies in literature. For example, Loughran et al. (2009) highlight the value of summing word counts in financial text analysis to derive domain insights.

We sum sparse features, across tokens, from an SAE (EleutherAI, 2024) applied to layer 30 (occurring at 90% of model depth). At this layer, we capture relevant features from preceding layers via the skip connection (Vaswani et al., 2017), but not the logit-related features that tend to occur at the very last layers (Ghilardi et al., 2024).

The skip connection ensures that a single SAE captures the entire residual stream (Longon, 2024), inherently including information from all preceding layers, thus ensuring that the summed sparse features represent a comprehensive aggregation of the model's internal representation of a company description. We analyze summed sparse features, and observe an interesting exponential decay pattern in feature activation frequencies (Figure 1).

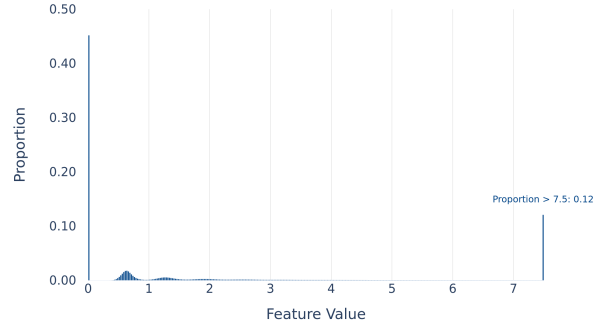Figure 1 highlights the sparsity of LLM latent

Figure 1: Distribution of summed feature activations.

features – even when these are summed across thousands of tokens – motivating feature summing as an approach. In this context, a single active feature has, on average, before summing, a value of ≈ 0.7 (the first bulge).

This method also addresses a limitation in using embeddings (Vamvourellis et al., 2023), which require equal-length sequences for comparison. By focusing on cumulative feature occurrences, summed sparse features enable comparisons between sequences of arbitrary lengths, offering greater flexibility for analyzing variable-length financial datasets.

3.3 Clustering

We benchmark our sparse features against embeddings and SIC/BISC-codes, where each SIC/BISC-code is its own cluster.

Each clustering method group G_k represents a distinct grouping methodology (i.e. G_{CD} uses the cosine distance metric in our Sparse Features, while G_{BERT} is based on the BERT embedders).

Within each model group G_k, clusters are generated independently for each year from 1996 to 2020. Thus, G_k is formally structured as a set of yearly clustering outcomes:

$$G_k = \left\{ G_k^{(y)} \mid y \in \{1996, 1997, \ldots, 2020\} \right\},$$

where $G_k^{(y)}$ is the set of clusters formed in year y:

$$G_k^{(y)} = \{C_1^{(y)}, C_2^{(y)}, \ldots, C_n^{(y)}\},$$

where $C_i^{(y)} \subseteq \{$Companies in year $y\}$. Each cluster $C_i^{(y)}$ contains a unique subset of companies active in year y, ensuring that clusters are independent across different years.

To evaluate each clustering model, we compute the mean intra-cluster correlation $MC(G_k^{(y)})$:

---

[5]The first 3 digits of the SIC code splits companies into 12 industry/major-divisions, referred to hereafter as BISC (Broader Industry Sector Code) (U.S. Occupational Safety and Health Administration, 2001).

$$\text{MC}(G_k^{(y)}) = \frac{1}{|G_k^{(y)}|} \sum_{C_i^{(y)} \in G_k^{(y)}} \frac{1}{|C_i^{(y)}|} \sum_{(a,b) \in C_i^{(y)}} \rho(a,b),$$

where $\rho(a,b)$ denotes the Pearson correlation of the logged monthly returns for companies a and b for the given year y. This metric quantifies the coherence of stock returns within clusters, providing a measure of how meaningful the cluster is.

We define the **overall mean correlation** (our main evaluation metric) of cluster groups G_k across years as:

$\text{MC}(G_k) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \text{MC}(G_k^{(y)})$, where $\mathcal{Y} = \{1996, \ldots, 2020\}$..

3.3.1 Clustering sparse features

Sparse features lack the locality and smoothness of embeddings (Kiros et al., 2015; Bischke et al., 2019) to define reliable similarity metrics. For instance, the TopK activation function (Gao et al., 2024) introduces sparsity, but with a strong discontinuity (truncates all features not in the top 128).

To overcome these limitations, we apply Principal Component Analysis (PCA) to the raw features[6]. PCA mitigates the impact of non-activating features by reducing dimensionality, and retains only the most informative feature directions. Furthermore, PCA expedites our computations.

To cluster the PCA-transformed sparse features, we adopt the graph-theoretic framework of Bonanno et al. (2004), employing Minimum Spanning Trees (MSTs) to extract hierarchical structures from financial data. A fully connected graph is constructed with edge weights representing a particular distance metric. The MST encodes a subdominant ultrametric, with ultrametric distance defined by the maximum edge weight on the unique path between two nodes[7]. We remove edges above a specified weight level, defining this as the "cut-off threshold" (θ), generating clusters directly from the MST. This eliminates the need for additional clustering steps, ensuring stable and interpretable results consistent with Bonanno et al. (2004).

Cosine Distance: We define the normalized cosine distance between our PCA-transformed sparse features as CD, which we use for clustering. The

resulting clusters are denoted as G_{CD}. This metric measures dissimilarity, which captures angular separation rather than absolute magnitude differences (Zafarani-Moattar et al., 2021). For each pair of companies i and j such that both companies belong to the same year[8], the cosine similarity is computed as:

$$S_{i,j} = \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_i\|\|\mathbf{g}_j\|}$$

where $\mathbf{g}_i$ and $\mathbf{g}_j$ are the PCA-transformed feature vectors, $\mathbf{g}_i \cdot \mathbf{g}_j$ denotes the dot product, and $\|\mathbf{g}_i\|$ represents the Euclidean norm (L^2-norm).

The cosine distance is then given by:

$$d_{\cos}(i,j) = 1 - S_{i,j}$$

We then normalize the cosine distance[9], defining the normalized distance function as CD. CD is used to determine the edge weights of the Minimum Spanning Tree (MST), and we apply a cut-off threshold θ to prune high-weight edges. The resulting connected components define the clusters G_{CD}[10].

Cut-off θ calibration: To determine the MST cut-off threshold θ for G_{CD}, we initially apply a two-fold temporal cross-validation scheme: θ is chosen to maximize the average intra-cluster correlation across two time periods covering 25% and 50% of our dataset. We define this as G_{CD} [11].

We ablate this choice by introducing a rolling variant. A separate $\theta_y^\star$ is chosen for each year y, based only on a five-year rolling lookback window:

$$\theta_y^\star = \arg \max_{\theta \in \{-4.5, -4.4, \ldots, -1.0\}} \frac{1}{5} \sum_{s=y-5}^{y-1} \text{MC}^{(s)}(\theta),$$

We rebuild $G_{\text{CD}}^{(y)}$ with $\theta_y^\star$, and report $\text{MC}^{(y)}(\theta_y^\star)$ as the yearly mean correlation statistic for each year $y = 2001, \ldots, 2020$; earlier years serve only as the look-back window. We define this rolling setup as G_{CDR}, for results see Appendix D, which confirms the robustness of our sparse-feature clusters under strict out-of-sample evaluation.

3.3.2 Clustering embeddings

Following Vamvourellis et al. (2023), each of the embedders discussed above is employed to define a unique clustering method group: (a) G_{BERT};

---

[6] We fit PCA globally across 1996–2020 for consistent eigenvectors, $n_{\text{components}} = 4000$ captures 89.92% variance.

[7] To enforce the ultrametric property, we employ single-linkage hierarchical clustering, which groups nodes by iteratively merging the pair of clusters with the smallest maximum distance between any two points. This process satisfies the ultrametric inequality ($d_{ij} \leq \max(d_{ik}, d_{kj})$) by construction.

[8] Note that we define pairs (i,j), ensuring that company i and company j are only compared within the same year.

[9] Normalizing cosine-based distances can enhance the performance of clustering algorithms (Uykan, 2021).

[10] We also refer to G_{CD} as $G_{\text{Sparse\_Features}}$ in our paper.

[11] See Appendix C for the optimization of G_{CD}'s cutoff.

(b) G_{SBERT}; and (c) $G_{\text{PaLM-gecko}}$[12] (details in Appendix B).

The SIC/BISC families are clusters by definition, and hence don't require further calibration.

3.4 Pairs trading

Our downstream task is pairs trading – a type of statistical arbitrage strategy that typically assumes a long-run equilibrium relationship between two stocks (Fallahpour et al., 2016). We begin by splitting the dataset into an in-sample period (Jan 2002–Dec 2013) and an out-of-sample period (Jan 2014–Dec 2020), with clusters G_k such that $k \in \{\text{Embedders, Sparse\_Features, SIC, BISC}\}$.

The pairs trading strategy consists of:

1. **Pre-selection:** For each cluster $C_i \in G_k$, stock pairs are filtered if the Pearson correlation of their monthly logged returns exceeds 0.95 during the in-sample period.

2. **Co-integration Testing:** An Engle-Granger co-integration test is conducted on stock prices (Jan 2002–Dec 2013) of pre-selected pairs using the Augmented Dickey-Fuller (ADF) statistic to assess the stationarity of the residual spread. Pairs with a p-value below 0.01 are considered co-integrated.

3. **Trading:** The identified co-integrated pairs for each G_k are evaluated out-of-sample[13] (Table 1). We assess co-integration effectiveness within each method group G_k via the entire portfolio's Sharpe ratio[14].

3.5 Interpretability

We show interpretability over a sample of 1000 features across 300 clusters. Clusters are formed using cosine distance, which can be interpreted as parallelism between the feature vectors (feature proportionality). There is no linear mapping between features and cosine distance (Appendix H), hence, we adopt an activation patching framework (Zhang and Nanda, 2024) with respect to cosine distance. This means that we obtain an interpretation of a cluster using the features that have the largest impact on cosine distance across the cluster when they are zeroed out (set to 0).

We define the importance of feature i as the total absolute variation in cosine distance across the cluster when feature i is zeroed out. Let g_i, g_j be PCA-transformed feature vectors i, j. Moreover, let g_i^z, g_j^z be the same vectors with feature z set to 0 before applying the PCA. We define the absolute impact on the cosine distance of feature z:

$$imp(z) = \sum_{i,j}^{cluster} |CD(g_i, g_j) - CD(g_i^z, g_j^z)|.$$

There are 2 necessary conditions for an interpretation of a cluster to be valid:

1. **Sparsity** There are n = 131,072 features, and we need to interpret a cluster using only a small subset of $k << n$ *important* features.

2. **Interpretability:** The sparse features that we use need to be interpretable on our dataset.

To obtain the set of *important* sparse features that constitutes the interpretation of a cluster, let F be the full set of n characteristics and define *impact* of a subset of features $S \subseteq F$ as follows:

$$\text{IMP}(S) = \sum_{z \in S} \text{imp}(z).$$

Then the set of *important* features, S^*, is given by

$$S^* = \arg\min_{S \subseteq F} |S| \quad \text{subject to} \quad \text{IMP}(S) \geq \text{IMP}(F \setminus S).$$

S^* is the smallest subset of features whose total impact on cosine distance in the cluster equals or exceeds that of the remaining features. We populate S^* by adding the most important feature in $F \setminus S$ to S until $IMP(S) \geq IMP(F \setminus S)$.

We interpret our *important* features using an auto-interpretability pipeline. First, the Gemini 2 Flash language model is prompted to explain a feature given examples of when the feature activates and when it does not. Then, the model predicts latent activations for new sentences based on its prior explanations (*fuzzing*). Interpretability is measured as the success rate in fuzzing.

While there is no benchmark for the interpretability of Llama 3.1 8B sparse features, we compare with the closest benchmark in the literature: Gemma 2 9B on the "Red Pajama" and "The Pile" datasets (Paulo et al., 2024).
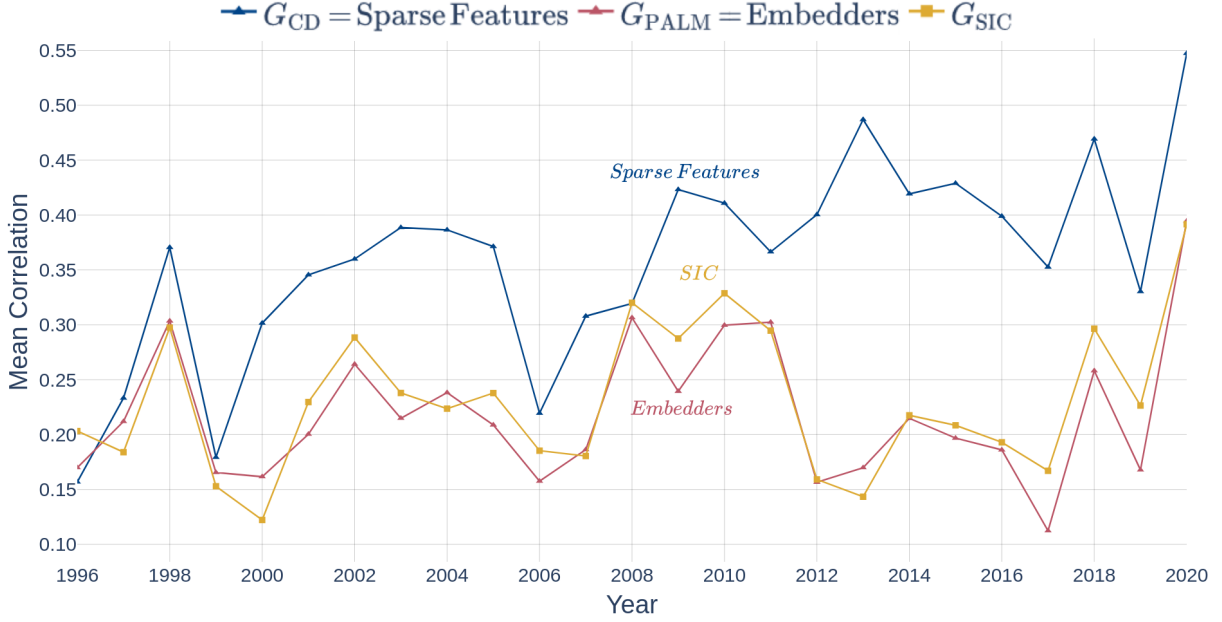
---

[12]We collectively refer to G_{BERT}, G_{SBERT}, and $G_{\text{PaLM-gecko}}$ as $G_{\text{Embedders}}$ for simplicity and to streamline discussion.

[13]See Appendix E for trading logic details

[14]The Sharpe Ratio quantifies risk-adjusted returns, measuring excess return per unit of risk (Guasoni and Mayerhofer, 2018; Peters, 2011).

Figure 2: Overall Mean Correlation ($\text{MC}(G_k)$) of G_{CD} (Normalized Cosine Distance Cluster Group) vs PaLM vs SIC Benchmarks between 1996-2020. Note that we use PaLM and SIC-codes for comparison, as they have the highest $\text{MC}(G_k)$ among the embedding-based and traditional benchmark groups, respectively.

4 Results

4.1 Clustering results

For each clustering method group G_k, we evaluate their $\text{MC}(G_k)$, and Sharpe Ratios (see Table 1 and Figure 2). The results demonstrate that clusters derived from our Sparse Features significantly outperform Embeddings, SIC-codes and BISC in terms of clustering similar companies.

Clustering Group ($\mathbf{G_k}$)	$\mathbf{MC(G_k)}$	Sharpe Ratio
Our Contribution		
$\mathbf{G_{\text{CD}}}$	**0.359**	**12.18**
$\mathbf{G_{\text{CDR}}}$	**0.385**	**9.69**
Embedding Benchmark		
G_{BERT}	0.198	7.58
G_{SBERT}	0.219	7.69
$G_{\text{PaLM-gecko}}$	0.219	10.57
Traditional Benchmark Cluster Groups		
G_{SIC}	0.231	9.70
G_{BISC}	0.187	7.58
Population[15]	0.161	—

Table 1: Performance comparison between different clustering groups (averaged across 1996-2020).

---

[15]Population group represents $\text{MC}(G_k)$ on the full dataset.

4.2 Pairs trading results

Sharpe ratios (risk-adjusted profits) were recorded for evaluation in backtesting. Within pairs trading, Hong and Hwang (2023) find pairs with higher fundamental similarity outperform those with weaker economic ties by reducing non-convergence risk. In line with these findings, our clustering approach can outperform Embedders and Traditional Classifications in Sharpe Ratio (Table 1), suggesting it may capture more fundamental company similarities.

4.3 Interpretability results

Interpretability	
Our Contribution	
Top 1% Features (G_{CD}) [16]	**80%**
Top 1% Features (G_{CDR})	**77%**
Average Feature	62%
Interpretability Benchmarks (Gemma 2 9B)	
The Pile	76%
Red Pajama	76%
Random Interpretation Baseline	
Fuzzing Score	51%

Table 2: Interpretability of SAE Features.

With regards to our first interpretability requirement, sparsity, we measure what percentage of features are *important* per cluster (Appendix G), and

find that the median cluster is very sparse with only 5% of *important* features.

In terms of interpretability, we observe that most features are interpretable (Table 2). Moreover, features that are *important* across multiple clusters, those we most want to interpret, also tend to be more interpretable (Figure 3). In particular, *top 1% features* (features in the first percentile for the amount of clusters they are important for) are 80% interpretable.
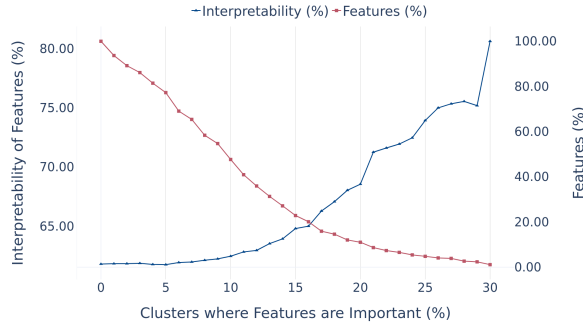

Figure 3: Interpretability Score of Features by Percentage of Clusters (G_{CD}) where Features are Important. Data selected between 100% (all features) and 1%.

Finally, we run the same experiments on the clusters constructed using the rolling cutoff (i.e. G_{CDR}), and our experiments yield similar results: *top 1% features* are 77% interpretable. In terms of sparsity, the median cluster is very sparse with only 1% of *important* features (Appendix G). The trend where more *important* features are more interpretable also holds (see Figure 4).
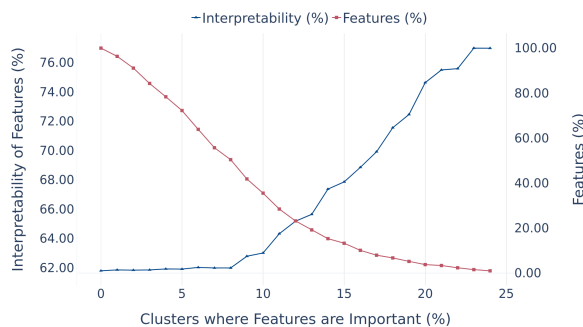

Figure 4: Interpretability Score of Features by Percentage of Clusters (G_{CDR}) where Features are Important.

4.4 Limitations

We do not fine tune embedders, SAEs, or LLMs. These could be exciting directions for future work.

---

[16]Top 1% features are important for more clusters than the remaining 99%, they are not the top 1% for interpretability.

Reported Sharpe ratios should be interpreted cautiously as they may be sensitive to the choice of θ, slippage, regime shifts, and finite-sample bias (Lo, 2003; Bailey and López de Prado, 2012).

5 Conclusions

We find that using SAE features is an effective and interpretable method for computing company similarity. Future work might explore applications in portfolio diversification and hedging strategies; optimizing trading strategies through fine-tuning θ and modeling shifts in economic regimes; extending the framework to other domains such as healthcare; or ablation studies such as replacing MST clustering with K-means.

6 Acknowledgments

References

David H. Bailey and Marcos López de Prado. 2012. The sharpe ratio efficient frontier. *Journal of Risk*, 15(2):3–44. Available at SSRN: https://ssrn.com/abstract=1821643.

Benjamin Bischke, Patrick Helber, Damian Borth, and Andreas Dengel. 2019. Multi-task learning for disaster image classification. In *2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 224–227. IEEE.

G. Bonanno, G. Caldarelli, F. Lillo, S. Miccicche, N. Vandewalle, and R. N. Mantegna. 2004. Networks of equities in financial markets. *The European Physical Journal B - Condensed Matter*, 38(2):363–371.

Trenton Bricken, Jonathan Marcus, Siddharth Mishra-Sharma, Meg Tong, Ethan Perez, Mrinank Sharma, Kelley Rivoire, and Thomas Henighan. 2024. Using dictionary learning features as classifiers. Technical report, Anthropic.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, and Adam Jermyn. 2023. Towards monosemanticity: Decomposing language models with dictionary learning.

Valentin Buchner, Lele Cao, Jan-Christoph Kalo, and Vilhelm Von Ehrenheim. 2024. Prompt tuned embedding classification for industry sector allocation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 108–118, Mexico City, Mexico. Association for Computational Linguistics.

Shuangshuang Chen and Wei Guo. 2023. Autoencoders in deep learning—a review with new perspectives. *Mathematics*, 11(8).

Wanghu Chen, Huijun Li, Jing Li, and Ali Arshad. 2020. Autoencoder-based outlier detection for sparse, high dimensional data. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2735–2742.

Chowdhery. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352.

Danile Delphini, Stefano Battiston, Guido Caldarelli, and Massimo Raccaboni. 2019. Systemic risk from investment similarities. *PLOS ONE*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Douglas S. Ehrman. 2012. *The Handbook of Pairs Trading*. Wiley Trading.

EleutherAI. 2024. Sae-llama-3-8b-32x. Hugging Face. Model card: "This is a set of sparse autoencoders (SAEs) trained on the residual stream of Llama 3 8B using the RedPajama corpus. The SAEs are organized by layer, and can be loaded using the EleutherAI sae library." Retrieved from https://huggingface.co/EleutherAI/sae-llama-3-8b-32x.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Preprint*, arXiv:2209.10652.

Robert F. Engle and C. W. J. Granger. 1987. Cointegration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276.

Saeid Fallahpour, Hasan Hakimian, Khalil Taheri, and Ehsan Ramezanifar. 2016. Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach. *Soft Computing*, 20(12):5051–5066. Intraday US stocks' price data

Data is obtained from the FactSet Research Systems, Inc. (FactSet) The sample period is from June 2015 to January 2016.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *Preprint*, arXiv:2406.04093.

Davide Ghilardi, Federico Belotti, Marco Molinari, and Jaehyuk Lim. 2024. Accelerating sparse autoencoder training via layer-wise transfer learning in large language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 530–550, Miami, Florida, US. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Paolo Guasoni and Eberhard Mayerhofer. 2018. The limits of leverage. *Mathematical Finance*, 29(1):249–284.

Sungju Hong and Soosung Hwang. 2023. In search of pairs using firm fundamentals: is pairs trading profitable? *The European Journal of Finance*, 29(5):508–526.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. 2025. Are sparse autoencoders useful? a case study in sparse probing. *Preprint*, arXiv:2502.16681.

Dean Katselas, Baljit K. Sidhu, and Chuan Yu. 2017. Know your industry: the implications of using static gics classifications in financial research. *Accounting and Finance*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 329–339.

Michael Lan, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez. 2024. Sparse autoencoders reveal universal feature spaces across large language models. *arXiv preprint*, 2410.06981v1. Work done during the ERA-Krueger AI Safety Lab internship.

Andrew Lo. 2003. The statistics of sharpe ratios. *Financial Analysts Journal*, 58.

André Longon. 2024. Interpreting the residual stream of resnet18. *arXiv preprint arXiv:2407.05340*.

Mico Loretan and William B. English. 2000. Evaluating changes in correlations during periods of high market volatility. *BIS Quarterly Review*.

Tim Loughran, Bill McDonald, and Hayong Yun. 2009. A wolf in sheep?s clothing: The use of ethics-related terms in 10-k reports. *Journal of Business Ethics*, 89(S1):39–49.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arXiv:1802.03426.

George J. Miao. 2014. High frequency and dynamic pairs trading based on statistical arbitrage using a two-stage correlation and cointegration approach. *International Journal of Economics and Finance*, 6(3).

MSCI. 2020. Gics methodology 2020.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. *Preprint*, arXiv:2311.03658.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. 2024. Automatically interpreting millions of features in large language models. *Preprint*, arXiv:2410.13928.

Ole Peters. 2011. Optimal leverage from non-ergodicity. *Quantitative Finance*, 11(11):1593–1602.

Manda Raghava and Santosh Bharadwaj. 2014. Pairs trading using cointegration in pairs of stocks. Master of finance research project, Saint Mary's University, Halifax, Nova Scotia, September. Submitted for MFIN 6692 under the direction of Dr. J. Colin Dodds and approved by Dr. Francis Boabang, MFIN Director.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

U.S. Occupational Safety and Health Administration. 2001. Standard industrial classification (sic) manual. Accessed: 2024-11-08.

U.S. Securities and Exchange Commission. 2023. Form 10-k: Annual report pursuant to section 13 or 15(d) of the securities exchange act of 1934. Accessed: 2024-12-02.

U.S. Securities and Exchange Commission. n.d. Cik lookup. https://www.sec.gov/search-filings/cik-lookup. Accessed: 2025-03-20.

Zekeriya Uykan. 2021. On the effect of data centering on spectral clustering with cosine similarity. In *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, pages 326–331.

Dimitrios Vamvourellis, Michael Toth, Shubham Bhagat, Dhairya Desai, Dhruv Mehta, and Sara Pasquali. 2023. Company similarity using large language models. *arXiv preprint arXiv:2308.08031*. [Online; accessed 2-Dec-2024].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc.

Winton. 2018. Systematic methods for classifying equities. Technical report, Winton Capital Management Limited ("WCM").

Yahoo Finance. 2024. Yahoo finance.

Elnaz Zafarani-Moattar, Mohammad Reza Kangavari, and Amir Masoud Rahmani. 2021. A comparative study on transfer learning and distance metrics in semantic clustering over the covid-19 tweets. *arXiv preprint*, arXiv:2111.08658.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *Proceedings of the Twelfth International Conference on Learning Representations*.

A Data Preprocessing

We consider 220,275 annual SEC reports from 1993 to 2020, ignoring any de-lists, accompanied

by related meta-data on Company Name, Year, SIC-code, and CIK number (a unique SEC corporation identifier) (U.S. Securities and Exchange Commission, n.d.). CIK numbers are mapped to their corresponding publicly traded ticker symbol, from which the monthly logged returns are retrieved via Yahoo Finance (2024). We remove entries with missing or very short: company descriptions, ticker information, or monthly returns. This leaves us with 27,888 reports. We tokenize using Meta's Llama 3 8B Tokenizer (Grattafiori et al., 2024). We only retain companies that are consistently available for at least five years. In our analysis, we ignore pre-1996 data as the sample size is too small. To refine the dataset further, we retain only annual reports with token counts within the context window.

B   Clustering Embeddings

For BERT, we used `bert-base-uncased` from the transformers library. For SBERT, we used `all-MiniLM-L6-v2` from the `sentence_transformers` library. For PaLM-gecko, we used `textembedding-gecko@003` from the `vertexai` library.

Chunking: In our methodology, for both BERT and SBERT, we followed Vamvourellis et al. (2023) and implemented a chunking mechanism to accommodate the models' maximum token limit of 512. Specifically, company descriptions exceeding this limit were split into overlapping chunks of 512 tokens. The `[CLS]` embeddings of these chunks were averaged to generate a single document embedding of 1536 tokens. For PaLM-Gecko, we leveraged its extended context window of 3072 tokens and directly processed the descriptions without chunking.

The pipeline below is optimised through Optuna's Tree-structured Parzen Estimator (TPE) sampler for Bayesian hyperparameter optimization. The objective function maximizes $\mathrm{MC}(G_k)$. This search is constrained to 150 trials and a maximum timeout of 9 hours to balance thoroughness and resource usage:

Dimensionality Reduction with UMAP: Given the high dimensionality of the input embeddings (768-dimensional vectors derived from a BERT model), we first employ Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2020) to reduce these high-dimensional textual embeddings to a lower-dimensional space, preserving both local and global data structures. We optimize three UMAP parameters to improve the quality of the downstream clustering: (a) `n_components` (target dimensionality); (b) `n_neighbors`; and (c) `min_dist`. All embeddings are standardized and casted to `float32` to ensure computational efficiency.

Clustering with Spectral Clustering: After reducing dimensionality, we perform clustering using Spectral Clustering, which is capable of handling noise and complex cluster shapes, following Vamvourellis et al. (2023). We first construct an affinity matrix from a k-nearest neighbors (KNN) graph of the UMAP outputs. Spectral Clustering then operates on this graph's eigenstructure to form clusters. The number of clusters (`n_clusters`) is tuned via Optuna, while the neighborhood size (k) is set to a constant of 5, following Vamvourellis et al. (2023).

Temporal Cross-Validation: To evaluate the stability and temporal generalization of the resulting clusters, we employ temporal cross-validation. The dataset is split into chronological folds. This setup reduces temporal bias and assesses whether the identified cluster structure remains consistent over time. We used parallel processing to evaluate each fold.

Embedder Cluster Group (G_{embedder})	UMAP $n_{\text{components}}$	UMAP $n_{\text{neighbors}}$	UMAP min_dist
G_{BERT}	7	119	0.109
G_{SBERT}	7	79	0.012
$G_{\text{PaLM-gecko}}$	6	40	0.120

Table 3: Optimized UMAP Thresholds for Embedders

Embedder Cluster Group (G_{embedder})	Spectral n_{clusters}	Spectral $n_{\text{neighbors}}$
G_{BERT}	10	5
G_{SBERT}	49	5
$G_{\text{PaLM-gecko}}$	27	5

Table 4: Optimized Spectral Clustering Thresholds for Embedders
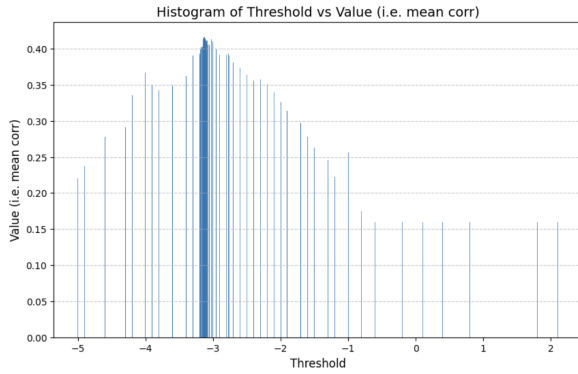
C Clustering Sparse Features

Figure 5: Optuna Study – Histogram of Sparse Features' MST cutoff thresholds. Maximizing Threshold = -3.130.

Figure 5 plots the distribution of candidate MST cut-off values θ (x-axis) against their corresponding mean intra-cluster correlations (y-axis). The long right tail approaches the overall population mean correlation (≈ 0.161) as θ loosens, while bulk of high MeanCorr values sits to the left (lower θ), reflecting tighter distance threshold groups similar firms.

D Clustering Sparse Features OOS with Rolling Frame

In terms of results, the forward rolling variant achieves a higher overall mean correlation of $\text{MC}(G_{\text{CDR}}) = 0.391$, compared to the temporal fold result of $\text{MC}(G_{\text{CD}}) = 0.359$. As shown in Figure 6, the optimal cut-off $\theta_y^\star$ evolves smoothly over time, while the out-of-sample mean intra-cluster correlation remains between 0.30 and 0.46 in most years—peaking in 2020 when market-wide correlations surged during the COVID-19 crisis.
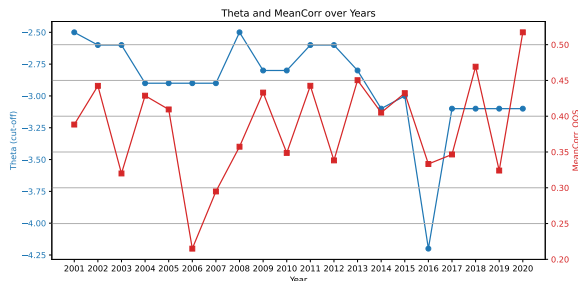

Figure 6: Walk-forward tuning results for the sparse-feature (G_{CDR}). Blue (left axis): optimal MST edge-weight cut-off $\theta_y^\star$ obtained from the preceding five-year rolling window. Red (right axis): resulting out-of-sample per-year mean intra-cluster correlation MC_y^{OOS}.

These findings confirm the robustness of our sparse-feature clusters under forward-looking evaluation.

E Trading Details

For each clustering-based strategy G_k, we simulate pair trades over the out-of-sample period 2014–2020 and record, for each business day t, the total portfolio value $V_{k,t}$. This series acts as the portfolio trajectory and is constructed as follows: (1) On each business day t, add realized PnL from any closed trades to cash. (2) Mark open positions to market and compute unrealized P&L. (3) Set $V_{k,t} = \text{cash}_t + \text{unrealized\_PnL}_t$ and append it to the portfolio trajectory series, which was subsequently used for Sharpe ratio calculations.

Following Miao (2014), we assumed zero transaction costs, opening positions when the residual spread deviated beyond $\pm 1\sigma$ its mean, and closing when the spread reverted to the mean. A stop-loss mechanism is triggered if the spread exceeds $\pm 2\sigma$. We obtained stock price data via finance.
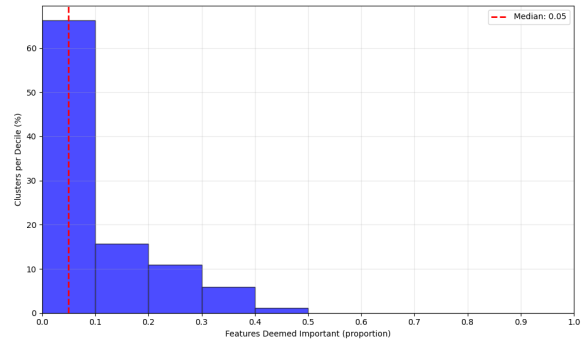
F Feature Sparsity Analysis

Figure 7: Distribution of the proportion of important features over clusters (G_{CD}).
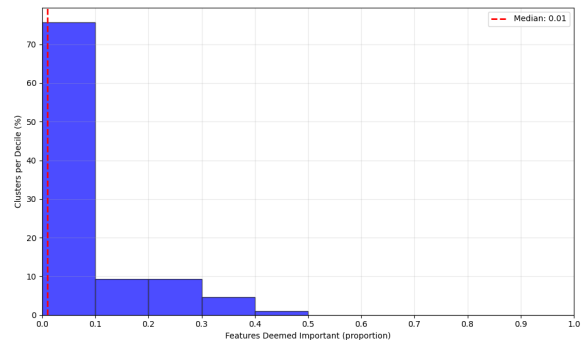
G Feature Sparsity Analysis

Figure 8: Distribution of the proportion of important features over clusters (G_{CDR}).

H Why a Linear Distance Must Be Trivial

Claim. If a function $d(\cdot, \cdot)$ on a vector space is both a *distance function* (metric) and *linear* in its arguments (plus symmetry), then $d(x, y) = 0$ for all x, y.

Proof. By the metric property, $d(z, z) = 0$ for any z. Pick arbitrary vectors x and y, and let $z = x + y$. Then

$$0 = d(z, z) = d(x + y, \ x + y).$$

Assume d is linear in the first argument and symmetric. By linearity on the first argument,

$$d(x + y, \ x + y) \ = \ d(x, \ x + y) \ + \ d(y, \ x + y).$$

By symmetry, $d(x, \ x + y) = d(x + y, \ x)$. Applying linearity in the first argument again,

$$d(x + y, \ x) = d(x, \ x) + d(y, \ x) = 0 + d(y, \ x),$$

because $d(x, x) = 0$ from the metric property. Symmetry again gives $d(y, x) = d(x, y)$. Hence

$$d(x, \ x + y) = d(x, y).$$

Similarly, $d(y, \ x + y) = d(x, y)$. Therefore,

$$d(x + y, \ x + y) = 2\, d(x, y).$$

But $d(x + y, \ x + y) = 0$, so $2\, d(x, y) = 0 \implies d(x, y) = 0$ for all x, y. Thus, if a distance were to be linear, it would be zero for all elements x,y, contradicting the usual requirement $d(x, y) = 0 \iff x = y$ unless the entire space is collapsed. $\square$