Representation Alignment Rests on Linear Structure

Kiril Bangachev[*†] kirilb@mit.edu

Guy Bresler[*‡] guy@mit.edu

Yury Polyanskiy[*§] yp@mit.edu

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA, 02139

Abstract

We investigate the Platonic Representation Hypothesis (PRH) through a tripartite statistical framework of representations: signal, bias, and noise. 1) Signal: We propose that Platonic alignment arises from the universal relationship between objects and attributes, which is encoded linearly in representations according to the Linear Representation Hypothesis (LRH). We provide evidence that LRH helps explain PRH by extracting linear object-attribute features with sparse autoencoders and showing that these sparse representations often exhibit stronger cross-modal alignment than their dense counterparts. 2) Bias: Models have different implicit biases due to the diverse architectures and training procedures used. We show that this difference can be partially mitigated. Centering and normalization consistently improve cross-model alignment. 3) Noise: Finite-sample training leads to noise in representations. We provide evidence that representational noise is driven by data scarcity by revealing a strong and consistent positive correlation between word frequency and alignment in LLMs and text embedding models. Synthesizing signal, bias, and noise, we propose a statistical model that refines the Linear Representation Hypothesis and explains further phenomena related to the alignment of representations emerging from diverse modern AI architectures. All code is available at github/L2PRH.

1 Introduction

1.1 The Platonic Representation Hypothesis

The Platonic Representation Hypothesis (PRH) [HCWI24] posits that modern machine learning models learn surprisingly similar representations of data, even when trained with different architectures, datasets, modalities, and objectives. A central claim of the hypothesis is a striking trend: *as models improve on downstream tasks, the representations they learn become increasingly similar* [HCWI24]. PRH is appealing because it hints at a universality in representation learning. Disparate models may be converging toward a shared, *ground truth* (Platonic) representation of the world. The potential value of such representations is enormous. If learned features reliably captured what an object is and is not, they could provide a foundation for discrimination, detection, synthesis, causal reasoning, and

[*] Author names appear in alphabetical order by last name.

[†] Supported by NAE Grand Challenge Vest Fellowship.
[‡] Supported by NSF award 2428619.
[§] Supported by NSF Grant CCF-21-12665 and by generous gifts from Jane Street and Google Research.

generation. Put more boldly, if "ground truth" representations of objects exist at all, access to them would seem difficult to separate from general intelligence. From this perspective, alignment toward a Platonic representation is not merely an emergent curiosity but a learning objective in its own right.

Puzzles About PRH. Pursuing this objective, however, requires a formal learning-theoretic framework that yields consistent, robust, and efficient algorithms for finding Platonic representations. To the best of our knowledge, the current PRH literature does not yet provide such a model. This gap persists in part due to several puzzles, central to the hypothesis:

Question 1. Representations align, but toward what? Empirical evidence [HCWI24, LIMB+25, SAC25, RSV+25, GWB26] is already overwhelming for the convergence of representations of data between machine learning models (and even the human brain!) across modalities and architectures. However, there is no consensus yet on what "Platonic object" the representations converge to.

Question 2. How do model specifics bias representations? Similarity of representations is shown to be strongly correlated with the performance of models on downstream tasks [HCWI24]. This is a compelling hypothesis that also carries a lot of philosophical weight – greater understanding and capabilities (of the model) lead to greater alignment with the (Platonic) ground truth. Unfortunately, this trend of correlating performance and representation similarity is limited to the KNN overlap metric [HCWI24]. For other metrics such as CKA, the authors observe that "alignment with CKA revealed a very weak trend of alignment between models, even when comparing models within their own modality". The recent work [GWB26] shows, in fact, that when appropriately controlling for width and depth, the trend in certain metrics, including CKA, altogether disappears. The *inconsistency across similarity metrics* raises a central question: how do model-specific factors such as depth, width, and modality bias representation similarity?

Question 3. How does inference data impact representation similarity? Fully explaining the alignment of representations only through the models and their capacity is quite unlikely since a representation $f_\theta(X)$ is a combination of the trained model f_θ and the inference data X. Missing from [HCWI24] and subsequent works is the basic question: what role does X have? This role is, of course, non-trivial. In an extreme setting where X is far out of distribution, we can hardly expect any non-trivial alignment. In fact, this idea is used to construct multimodal OOD detectors [KH25].

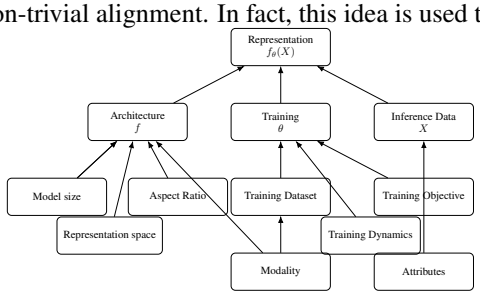


Figure 1: Constituent Parts of Learned Representations. Most important to this work are the first and second levels. Further levels are omitted.

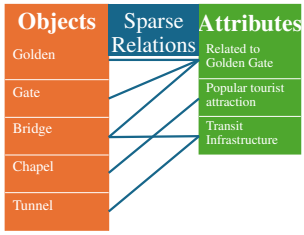


Figure 2: Sparse relationships between objects and attributes under LRH following an example in [TCM+24]. Encoding attribute-object relations in a bipartite graph is related to Formal Concept Analysis [GSW03, Xio26].

1.2 Our Framework: Signal, Bias, and Noise in Representations

Methodology: Our aim is to put forward a statistical model of learned representations that elucidates Questions 1 - 3. To do so, we decompose data representations into their constituent parts (Figure 1). This decomposition is our guide toward identifying *signal, bias,* and *noise* in representations.

1. Signal: The Linear Representation Hypothesis. Among f, θ, X, the only invariant between models is the inference data X. But what about inference could be Platonic? We suggest that the Platonic signal is a rather old and well-known geometric structure of learned representations, the Linear Representation Hypothesis (LRH). As stated in [FWL+25, GKP26], it posits that each representation is approximately a linear combination of representations of its constituent attributes (see Section 2.2 for more background on LRH). More formally, representations linearly encode a ground truth sparse relationship between objects and their attributes as in Figure 2.

We confirm this hypothesis by training top-k sparse autoencoders [MF14] on features produced by a variety of different models. Our first contribution is to ***demonstrate that sparse features are similar across models. In particular, the alignment between sparse features is typically higher than the alignment between the dense features from which they are extracted (Figure 5).*** We propose that the Platonic signal is the sparse object-attribute structure posited by LRH: each object activates a small set of attributes, and learned representations encode these attributes approximately linearly. ***This is the first link between the two hypotheses on representation geometries.***

2. Bias: Architecture and Training. Unlike inference data – which is common across representations – architecture and weights are different. As a result, one may expect an *approximation error* in the learned representations due to the concept class described by the architecture f. Likewise, there should be an *optimization error* due to the training procedure resulting in weights θ. Hence, a statistical procedure for removing the bias of models should increase the alignment of representations. We confirm this hypothesis in Section 3.2 via a simple debiasing procedure that subtracts the population mean (over the entire inference dataset for the same model). Our second main contribution is to ***demonstrate that the simple debiasing procedure of subtracting the population mean and normalizing consistently increases alignment of representations.***

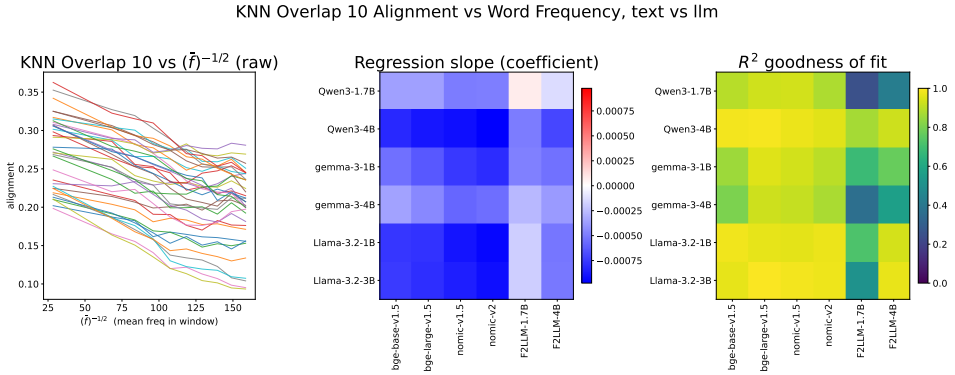


Figure 3: Experiment on wordfreq dataset. Each line corresponds to the alignment of one pair of text embedding and LLM models. On the y-axis we have the alignment value and on the x-axis, the scaled relative frequency as $f^{-1/2}$ where f is the frequency (so, more frequent words are to the left).

3. Noise: Frequency of Inference Data. To address Question 3 and go beyond the extreme setting of lack of alignment in far-OOD samples, we recall Statistics-101. More samples lead to lower estimation errors and, hence, better signal-to-noise ratio. This yields the natural hypothesis that alignment is greater for more frequent data points. We test this hypothesis over a dataset where data frequency can be reliably and accurately measured – common English words. We ***demonstrate the trend that alignment is greater for more frequent words between LLMs and text embedding models.*** Our experiments furthermore suggest a noise rate proportional to inverse square root of data frequency consistent with classical central limit theorem behavior of estimators, see Figure 3.

1.3 Our Statistical Model of Representations and Implications

Having identified the different components – signal, bias, and noise – in learned representations, we can write down a statistical model as follows. It refines the Linear Representation Hypothesis by explicitly attributing different components of the model to different components of the representations:

$$f_\theta(X) = A(\theta, f) \times (Z(X) \odot M(X, \theta, f)) + \eta_{X,\theta,f}. \qquad \text{(Statistical Model of Representations)}$$

The different components are:

Platonic Signal: Sparse feature vector $Z(X) \in \{0,1\}^m$ which is a function purely of the inference data X. It encodes which attributes are relevant to object X.

Bias 1: Signal Magnitude $M(X, \theta, f) \in \mathbb{R}^m_{\geq 0}$ which encodes the importance of each attribute for the object X in the specific training dataset $\mathcal{D}$, learning objective of f, or architecture capability.

Bias 2: Dictionary of Attributes $A(\theta, f) \in \mathbb{R}^{d \times m}$ which depends on the architecture and training.

Estimation Error $\eta_{X,\theta,f}$ which is determined by how well X is learned during training θ with model f. Model architecture is relevant, for instance due to its capacity. Training and weights are relevant as they determine how data is utilized (e.g., learning rate schedule).

Assumptions on Representation Components. Our Statistical Model of Representations predicts several phenomena related to PRH observed in the literature under common assumptions on the different components of dictionary learning (see, for example, [AGMM15]).

P.1) *Data Normalization:* $\|f_\theta(X)\|_2 = 1$. This property is a non-restrictive assumption as far as cosine similarities are concerned since we can always add a normalization layer.

A.1) *Sparsity:* Each $Z(X) \in \mathbb{R}^m$ has sparsity $k = o(m)$. This common assumption can be interpreted as the fact that each object has a limited number of attributes. The small residuals of SAEs when fitting sparse autoencoders support A.1), see Appendix C.8.

A.2) *Dictionary Incoherence:* For every $i \in [m]$, the feature $A(\theta, f)_i$ has unit norm, $\|A(\theta, f)_i\|_2 = 1$, and for every $1 \le i < j \le m$, $|\langle A(\theta, f)_i, A(\theta, f)_j\rangle| \le \epsilon_{\text{dict}} = o(1/k)$. This assumption is common in theoretical works on sparse autoencoders such as [AGMM15, AGM14]. It is also closely related to the proposition that "causally separable" concepts are represented as orthogonal vectors [PCV24]. When $m > d$, an orthogonal frame does not exist, so the best one can do is a near-orthogonal frame with small ϵ_{dict}. See Figure 4, Appendix C.6 for empirical support of the small incoherence of SAE dictionaries.

A.3) *Estimation with Non-Trivial Signal*. Non-zero coordinates are well-expressed. Namely, for some $0 < \phi_{\text{sig}} < \Phi_{\text{sig}}$ and for any X, i such that $Z(X)_i = 1$, it holds that $\frac{\phi_{\text{sig}}}{\sqrt{k}} \le M(X, \theta, f)_i \le \frac{\Phi_{\text{sig}}}{\sqrt{k}}$. This assumption is common in theoretical works, see [AGMM15, Assumption (2) of model]. See also Appendix C.7 for empirical support.

MA.1) *Estimation with Non-Trivial Noise*. With high probability over $\eta_{X,\theta,f}$, it holds that $\|\eta_{X,\theta,f}\|_2 \le \epsilon_{\text{noise}}$ for some $\epsilon_{\text{noise}} \le 1$. This is our main assumption. It effectively states that the Linear Representation Hypothesis holds [AGMM15, AGM14, GKP26].

Representation Similarity. Let X_1, X_2 be two different objects and let f_θ be a model. For brevity, denote $A = A(\theta, f), M_i = M(X_i, \theta, f), Z_i = Z(X_i), \eta_i = \eta_{X_i,\theta,f}$. Under the assumptions above:

$$
\begin{aligned}
&\langle f_\theta(X_1), f_\theta(X_2)\rangle \\
&= \langle A(\theta, f) \times (Z(X_1) \odot M(X_1, \theta, f)) + \eta_{X_1,\theta,f}, A(\theta, f) \times (Z(X_2) \odot M(X_2, \theta, f)) + \eta_{X_2,\theta,f}\rangle \\
&\langle A(Z_1 \odot M_1), A(Z_2 \odot M_2)\rangle + \langle f_\theta(X_1), \eta_2\rangle + \langle f_\theta(X_2), \eta_1\rangle - \langle \eta_1, \eta_2\rangle \\
&= \Big\langle \sum_{i\,:\,(Z_1)_i=1} A_i(M_1)_i, \sum_{i\,:\,(Z_2)_i=1} A_i(M_2)_i \Big\rangle + \langle f_\theta(X_1), \eta_2\rangle + \langle f_\theta(X_2), \eta_1\rangle - \langle \eta_1, \eta_2\rangle
\end{aligned}
$$

Recalling the assumptions, we rewrite as follows (the bounds are derived in Appendix A.1):

$$
\begin{aligned}
&\sum_{i\,:\,(Z_1)_i=(Z_2)_i=1} (M_1)_i(M_2)_i && \boxed{\text{Signal} \in \Big[\frac{\phi_{\text{sig}}^2\langle Z_1, Z_2\rangle}{k}, \frac{\Phi_{\text{sig}}^2\langle Z_1, Z_2\rangle}{k}\Big]} \\
&+ \sum_{i\neq j\,:\,(Z_1)_i=1,(Z_2)_j=1} (M_1)_i(M_2)_j\langle A_i, A_j\rangle && \boxed{\text{Bias} \le k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2} \\
&+ \langle f_\theta(X_1), \eta_2\rangle + \langle f_\theta(X_2), \eta_1\rangle - \langle \eta_1, \eta_2\rangle && \boxed{\text{Noise} \le 3\epsilon_{\text{noise}}}
\end{aligned}
\tag{1}
$$

Aristotelian View: Sparse Signal and The KNN overlap metric. The original paper [HCWI24] suggests a strong positive correlation between model performance and alignment in the KNN overlap metric for small k. The work [GWB26] further suggests that other metrics such as CKA and SVCCA suffer from biases arising due to depth and width of models. [GWB26] goes further to propose that Platonic is only the local neighborhood structure, but not the global geometry.

Our Statistical Model of Representations explains why for strong models, similarity of local neighborhoods arises while global geometry may differ. We give a full proof in Appendix A.

Proposition 1. *Suppose that* $k\epsilon_{\text{dict}}\Phi_{sig}^2 + 3\epsilon_{\text{noise}} \le 1$. *Then, for any* $\gamma > 0$,

1. If $\langle Z(X_1), Z(X_2)\rangle \geq \frac{2k}{\phi_{\text{sig}}^2}(k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma/2)$, then $\langle f_\theta(X_1), f_\theta(X_2)\rangle \geq k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma$.

2. If $\langle f_\theta(X_1), f_\theta(X_2)\rangle \geq k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma$, then $\langle Z(X_1), Z(X_2)\rangle \geq \frac{k}{\Phi_{\text{sig}}^2}\gamma$.

In particular, for strong models (such that bias and noise are simultaneously small, $k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} \leq 1$), the local neighborhood of $f_\theta(X_1)$ is nearly determined as the vectors $f_\theta(X_2)$ for which the number of shared attributes – $\langle Z(X_1), Z(X_2)\rangle$ – exceeds a certain threshold.

While the local neighborhoods are shared between representations, global geometry might be lost due to differences in magnitudes $M_i(X, \theta, f)$, dictionaries $A(\theta, f)$, and random noise terms $\eta_{X,\theta,f}$.

Bias and Similarity of Training Objectives. [CLM+24] demonstrates that similarity of training objectives leads to increased alignment. This view is consistent with our Statistical Model of Representations. One may expect that distinct features are important for different objectives. As the sparse-feature weights $M(X, \theta, f)$ and the dictionary $A(\theta, f)$ depend on the objective, models with the same objective are more aligned.

Bias and Model Width. It has been empirically observed that models in higher dimension tend to have higher alignment in metrics such as CKA [GWB26]. The work [GWB26] explains this via the fact that in high dimension, even two independent matrices, each with independent entries, have high CKA alignment since the spectrum of the corresponding Gram matrices converges weakly to a limiting measure [Wac78]. While illuminating, the intuition from independent-entry ensembles may not carry over to representations in modern AI systems. The latter have a highly non-random structure that allows for geometric properties such as LRH (see Section 2.2) or success in multimodal retrieval (see [BBNP25]).

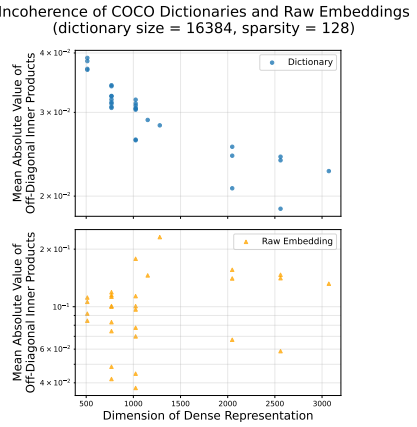


Figure 4: Higher dimensions consistently lead to smaller in magnitude inner products of dictionary elements for the 30 models from Table 1. However, this is not true for raw embeddings. In Appendix C.6, we test other datasets, sparsities, and dimensionalities.

Our Statistical Model of Representations explains the positive correlation between alignment and dimension even when correlations between the matrix entries due to common features of objects exist. Higher dimensionality enables better sphere packings of feature representations in the dictionary, thus driving ϵ_{dict} down (see Figure 4 for evidence that this happens in practice). A lower value of ϵ_{dict} in turn reduces the bias in (1) and makes representations more similar. Figure 4 also shows that the incoherence of raw representations does not decrease with dimension. This refutes an alternative explanation in the style of [GWB26] that the embeddings themselves are closer to orthogonal.

Noise and Strong Models. Stronger models are typically trained on larger or more carefully filtered datasets in order to decrease noise. Again, this leads to a smaller deviation of $\langle f_\theta(X_1), f_\theta(X_2)\rangle$ from the Platonic signal $\frac{\langle Z(X_1), Z(X_2)\rangle}{k}$, which explains the trend between alignment and model performance in [HCWI24]. Our experiment in Figure 7 additionally suggests that when both models have been trained on more data, alignment between them is greater. Likewise, the relationship between noise and strong models elucidates the observation that "linear representations of truth emerge with scale" from [MT24]. Representations from stronger models are less noisy, so are closer to perfectly satisfying the linear relationship between objects and attributes, i.e. $\eta_{X,\theta,f} = 0$.

2 Prior Work and Preliminaries

Modern AI systems represent various forms of data as vectors, either explicitly in representation models such as CLIP, DINO, or SigLIP, or implicitly via neuron activations in intermediate layers. We will abstract all of these as real-valued vectors $f_\theta(X)$ where f is the model architecture, θ the trained weights, and X the data to be represented. We scale all features to be unit norm.

2.1 Platonic Representation Hypothesis

Since the publication of [HCWI24], there has been extensive interest in the PRH including alignment beyond image-text models [LIMB+25, WLD+25, MJD25, GKJ+26], a learning-theoretic formalization [IHR25], applications [GSW+25, SMC+24, GWLB25, LLL+24, YYK25], possible explanations [LVAW25, ZC25, Lob25], refinements [GWB26], and even some objections [Tho25].

Comparing Representations. To make PRH formal, we need a formal notion of representation similarity. Let $(X_i, Y_i)_{i=1}^N$ be a dataset of paired objects. Using two models f_θ, g_ϕ, form the representations $F = (f_\theta(X_1), \ldots, f_\theta(X_N)) \in \mathbb{R}^{d_1 \times N}$, $G = (g_\phi(Y_1), \ldots, g_\phi(Y_N)) \in \mathbb{R}^{d_2 \times N}$. The comparison is over the two Gramians $F^T F, G^T G \in \mathbb{R}^{N \times N}$ which contain the cosine similarities of unit-norm representations. The metrics that are used to compare $F^T F, G^T G$ in [HCWI24] and related works fall into two categories. *Cardinal:* Important is the magnitude of distances; examples include CKA, Unbiased CKA, and SVCCA. *Ordinal:* Important is the relative order of distances; examples include overlap and edit distance of KNN neighborhoods. See [KMB08, RGYSD17, KNLH19, HCWI24, IHR25, GWB26] for definitions and interpretations of the metrics.

2.2 Linear Representation Hypothesis

LRH predicts the existence of sparse linear relations between data representations. Early works on the Linear Representation Hypothesis analyze relationships of the form $f_\theta(\mathsf{man}) - f_\theta(\mathsf{woman}) \approx f_\theta(\mathsf{king}) - f_\theta(\mathsf{queen})$, demonstrating linearity of concepts (man-to-woman in the specific case) in models such as Word2Vec [MSC+13, MCCD13, MYZ13] and GloVe [PSM14].

Sparse Dictionary Learning Formulation. Recent work on sparse linear relations between data representations has focused on a different but related formulation of the Linear Representation Hypothesis – each representation is a superposition of the representations of different attributes [PCV24, FWL+25, GKP26, VKJ+25]. This view can be formalized in the framework of sparse dictionary learning, where each feature $f_\theta(X)$ is a sparse linear combination of some representations of attributes $(D_a)_{a \in \mathcal{A}}$ for a set of global attributes $\mathcal{A}$. Namely, $f_\theta(X) = \sum_{b \in B} c_b D_b$ where B is a set of relevant attributes and the c_b are *non-negative* coefficients. This formulation bridges LRH with the literature on mechanistic interpretability, where sparse autoencoders are trained on top of dense representations and sparse interpretable features are extracted [BTB+23, TCM+24, GlTT+24, TSLL25, HCRS+24, PKB+25, XDF+25].

2.3 Comparison with Prior Work

Learning-Theoretic Perspective. [IHR25] also views representation alignment as a learning-theoretic task. They propose that there is a common reality Ξ and different models are different observations of it. In our work, we make both the common reality and observation models more explicit. First, we postulate that the reality Ξ is a sparse relationship between features and attributes. Second, we make our model fit into a standard formulation with signal, bias, and noise. The concreteness of our approach opens the door to empirical predictions which we verify experimentally. While illuminating, the abstract formulation of [IHR25] does not yield any concrete experimental predictions.

Alignment emerging from training. [ZC25] shows that representation similarity arises via the inductive bias of (stochastic) gradient descent on linear networks. While training dynamics certainly play a role (note that training θ is part of our Statistical Model of Representations), the linear set-up is perhaps too simplistic to explain the alignment of features of practical models on real datasets. For instance, one byproduct of their main theorem is that the features at each layer of the network are the same up to a rotation and scalar multiplication. In particular, the trained network is equivalent to a single-layer network and the complexities of depth and architecture are not captured. [GMM24] takes a related approach and explains PRH via the assumption that the weights of trained models are independent samples from the same distribution.

Alignment emerging from a shared PMI matrix. [HCWI24] outlines that one possible cause of the Platonic Representation Hypothesis is that the Gram matrices of representations converge to the same Pointwise Mutual Information (PMI) matrix $\mathsf{K}_{\mathsf{PMI}}$. Convergence to the $\mathsf{K}_{\mathsf{PMI}}$ matrix has also been used to explain the Linear Representation Hypothesis in the form $f_\theta(\mathsf{man}) - f_\theta(\mathsf{woman}) \approx f_\theta(\mathsf{king}) - f_\theta(\mathsf{queen})$ [LG14, PCV24, ALL+16, SPA+19, GKJ+26, JRR+24, KKBW25]. Concretely, the PMI

matrix is defined as follows. The entry corresponding to the pair of data points (X_1, X_2) is given by $\mathsf{K}_{\mathsf{PMI}}(X_1, X_2) = \log \frac{P(X_1,X_2)}{P(X_1)P(X_2)}$. Here, X_1, X_2 are two data points, corresponding to object-concept or object-object pairs. The key identity is that when a fully unconstrained representation model f_θ is minimized using the InfoNCE or sigmoid contrastive objectives, it holds that $\langle f_\theta(X_1), f_\theta(X_2)\rangle = \mathsf{K}_{\mathsf{PMI}}(X_1, X_2) + C$ where C is some constant independent of X_1, X_2 (in the multimodal or object-context setting, $\langle f^1_\theta(X_1), f^2_\theta(X_2)\rangle = \mathsf{K}_{\mathsf{PMI}}(X_1, X_2) + C$ for different encoders f^1, f^2).

There are, however, several challenges in explaining PRH as convergence to the $\mathsf{K}_{\mathsf{PMI}}$ matrix: *1. Distributional:* The PMI matrix may not be similar across training datasets and modalities. *2. Geometric:* In practice, representations are low-dimensional ($d \leq 4096$) while the $\mathsf{K}_{\mathsf{PMI}}$ matrix captures co-occurrences between billions of data-points. Thus, for $\langle f^1_\theta(X_1), f^2_\theta(X_2)\rangle = \mathsf{K}_{\mathsf{PMI}}(X_1, X_2) + C$ to hold even approximately, $\mathsf{K}_{\mathsf{PMI}}$ has to be approximately low-rank. We don't know of any evidence for this. In the object-object setting used to explain PRH in [HCWI24], $\langle f_\theta(X_1), f_\theta(X_2)\rangle = \mathsf{K}_{\mathsf{PMI}}(X_1, X_2) + C$ can only hold if $\mathsf{K}_{\mathsf{PMI}}$ is positive semi-definite in the space orthogonal to the all-ones vector, which is again unjustified. *3. Optimization-Related:* In practice, the (inverse) temperature parameter is also trainable for InfoNCE-based models like CLIP [RKH+21] and sigmoid-loss-based models like SigLIP [ZMKB23]. When the temperature parameter is trainable rather than fixed, the geometry of global minimizers changes and does not correspond any longer to $\langle f^1_\theta(X_1), f^2_\theta(X_2)\rangle = \mathsf{K}_{\mathsf{PMI}}(X_1, X_2) + C$, see [BBNP25]. *4. Generality:* While the PMI matrix has been shown to arise in certain contrastive learners, it is less clear what the relevance of the PMI matrix is to other models such as LLMs and image foundation models.

3 Experiments

Metrics. We focus on the KNN overlap metric with a small value of $k = 10$ as in [HCWI24]. This metric captures similarity of local neighborhoods of representations. The recent work [GWB26] suggests that other spectral and geometric metrics suffer from undesirable biases. This is especially important for us when considering sparse features produced by SAEs since they only have non-negative coordinates, which leads, for instance, to the strong bias that all representations belong to the same orthant. Nevertheless, in Appendix C, we reproduce all experiments with other metrics including CKA, Unbiased CKA, KNN overlap with $k = 100$, and KNN edit distance with $k = 10, 100$.

Models. We perform experiments with 2 models from 3 families from each of the following 4 categories: Text Embedding, Image Embedding, Multimodal Contrastive, and LLMs. This leads to 30 different representations of each object (the text and image representations for each multimodal model are different). The models as well as their characteristics are given in Appendix C.5.2. Unlike in text, image, and multimodal embedding models, LLMs do not explicitly produce text embeddings. Instead, on a text input of T tokens, we find the $T \times D$ state of the residual stream of the middle layer and mean-pool it into a vector of dimension D, which we then normalize to have unit norm. While not necessary, the choices of mean-pooling and mid-layer are consistent with prior mechanistic interpretability literature such as [TCM+24, KXCR20]. By choosing a concrete layer, we avoid the confounder of depth arising from optimizing over layers [GWB26].

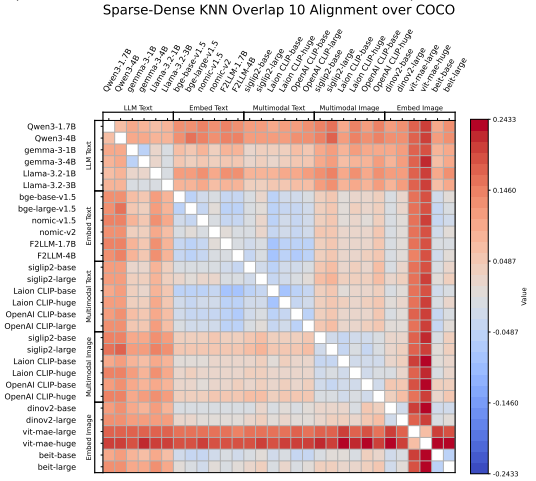


Figure 5: The difference of model alignment in the KNN-10 metric between sparse features and dense features over COCO. Alignment is higher (red) with few weak exceptions, which are predominantly for models over the same modality.

Datasets. The experiments in Sections 3.1 and 3.2 are over COCO [LMB+15]. In Appendix C, we reproduce the experiments over CC3M [SDGS18] and Visual Genome [KZG+17]. The experiments in Section 3.3 are over English words together with their frequencies in the wordfreq library [Spe22].

3.1 Signal: Sparse Features Increase Similarity

We show that the alignment between sparse features is typically higher than the alignment between the dense features from which they are extracted. This trend is especially prevalent for the alignment between two models with different objectives and modalities.

Sparse Features. To derive the sparse features, for each model f_θ resulting in an embedding matrix $F \in \mathbb{R}^{d \times N}$ we train top-k sparse autoencoders on F [MF14]. The concrete parameters we choose are $D = 16384$ and $k = 32$ for text models, $k = 64$ for multimodal models, $k = 128$ for image models (we use bigger k for models that more strongly rely on image data due to the intuitive fact that images contain more information than single-sentence captions). As is common, we additionally resample dead neurons (at every 2500 steps) to mitigate the issue of dead neurons [BTB+23]. Finally, we remove features that activate on more than 10% of the data points since they are unlikely to be monosemantic and on less than 0.001% since they are likely noise. We trained on a single H200 GPU. Each run took up to 1 hour per model per choice of dimension and sparsity.

Experiment 1: Alignment with Sparse Features. We averaged the KNN overlap similarity metric over 10 different uniformly random samples of size 1000 for the dense and sparse models.

Analysis. Alignment in the KNN-10 metric is higher for sparse features, especially in the case of models with different modalities. We explain this via the fact that dictionaries $A(\theta, f)$ which depend on model architecture and training (in particular, modality of training dataset) should have a similar bias for models with similar architecture and training. Thus, the fact that alignment is sometimes lower between multimodal image models when sparse features instead of dense ones are considered does not contradict our Statistical Model of Representations.

We note the special case of the ViT-MAE models. In the original PRH paper [HCWI24], the alignment of the MAE (Figure 9 in the arXiv version) with other models is surprisingly low compared to the alignment between other models. This is also true in our experiments as far as dense features are concerned. The low alignment of MAE models is largely mitigated when passing to sparse features.

Experiment 2: Correlation of Sparse Features. We additionally test the null hypothesis that sparse features between different models are independent. Concretely, suppose that the sparse features from two models $f^1_{\theta^1}$ and $f^2_{\theta^2}$ over paired data $(X_i, Y_i)_{i=1}^N$ are $Z_1, Z_2 \in \mathbb{R}^{N \times D}_{\geq 0}$. Since the sparse features are undefined up to a global permutation, we compute the correlation between Z_1, Z_2 as $\mathsf{corr}(Z_1, Z_2) \coloneqq \max_{\Pi \in \mathcal{S}_D} \frac{\langle Z_1, Z_2 \Pi \rangle}{\|Z_1\|_2 \times \|Z_2\|_2}$. This optimization problem over permutations is tractable since it is simply a maximal matching problem in a $D \times D$ weighted graph with weights $Z_1^T Z_2$. Following the methodology of [GWB26], we compare against a baseline of randomly permuting the objects, $\mathsf{corr}_\Phi(Z_1, Z_2) \coloneqq \max_{\Pi \in \mathcal{S}_D} \frac{\langle \Phi Z_1, Z_2 \Pi \rangle}{\|Z_1\|_2 \times \|Z_2\|_2}$ for Φ uniformly drawn from $\mathcal{S}_N$.

Analysis. The difference between $\mathsf{corr}(Z_1, Z_2)$ and the empirical mean of $\mathsf{corr}_\Phi(Z_1, Z_2)$ is on the order of thousands of standard deviations of $\mathsf{corr}_\Phi(Z_1, Z_2)$, see Appendix C.2. This strongly rejects the null that $\mathsf{corr}(Z_1, Z_2)$ and the permutation null $\mathsf{corr}_\Phi(Z_1, Z_2)$ come from the same distribution.

3.2 Bias

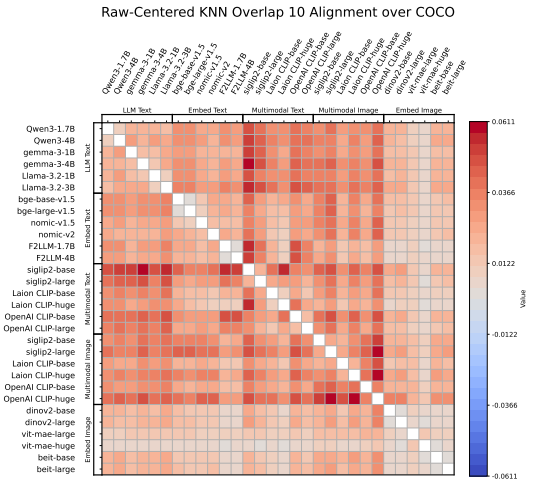


Figure 6: Effect of centering on alignment.

Experiment. We work with the same models and the COCO dataset, averaging over 10 random subsamples of size 1000. In Figure 6 we plot the difference of alignment for "raw" versus "centered and re-normalized" features.

Analysis. Centered features are more aligned across all pairs of models. We show that the trend is consistent across datasets and metrics in Appendix C.3. That centered representations are closer to the Platonic signal is also consistent with the observation that centering improves representation strength [MBV17].

We clarify that we do not claim that model bias is exactly the population mean. Rather, mean

subtraction may remove a dominant first-order anisotropic component, while higher-order biases may remain.

3.3 Noise: Signal-to-Noise Ratio

In our experiment demonstrating the importance of data frequency for alignment, we use text models (LLMs and text-embedding) to embed the most frequent English words paired with their relative frequencies from the wordfreq library [Spe22]. The utility of using words only is that for single words, there is a very simple, clear, and verifiable notion of what "data frequency" is.

Experimental Details. We measure the alignment of models under the chosen metrics by sliding a window of size 500 with step size 250 from more common to less common English words.

Analysis. The leftmost plot in Figure 3 shows a consistent trend for higher alignment over windows with higher average frequency. This trend supports the work [SS20] which establishes that attention-based text embedding models have poor performance on rare words. From our statistical perspective, this is caused by the high noise level in the representations. We furthermore fit a 1-dimensional linear regression for $f^{-1/2}$ versus alignment. We consistently get a good fit (rightmost plot) and negative coefficient (middle plot). The strong linear relationship between $f^{-1/2}$ and alignment is consistent with standard CLT-like estimation error proportional to inverse square root of samples.

We carry out the same experiment for different pairs of models and metrics in Appendix C.4. We also perform two ablation studies to demonstrate that the negative linear trend between $f^{-1/2}$ and alignment does not arise due to plausible confounders. Concretely, we address the confounders of number of tokens, by controlling for only single-token words for all models, and word type, by controlling for a single part of speech (adjective, verb, noun). In both experiments, the trend persists.

3.4 Other Experiments: Further Decomposing the Alignment

Our tripartite framework explains *what drives similarity of representations* – the universal sparse relationships between objects and attributes. Of course, differences between models do exist due to their architecture and training procedure as explained in Figure 1. Which of these differences of trained models are responsible for the similarity (or, lack thereof) of representations?

Experiment. To address this question, we set up the following experiment in which we *regress* the alignment of models (we have 30 models, so these are $\binom{30}{2} = 435$ pairs) on 10 easy-to-quantify model properties. They are respectively: 1) *Architectural*: model size, model depth, dimension of representations, objective. 2) *Training-related:* number of text tokens and number of images used in training. 3) *Mixed:* we also include the year of release of the model due to the innovation factor. More recent models tend to be better and, hence, can be expected to be more aligned to the Platonic ground truth. Whenever the model specifics are numeric, we create features corresponding to the min and max over the two models. For the categorical feature of modality, we one-hot encode the three possibilities (image-image, image-text, text-text). This process gives us 15 features, which we normalize to have unit norm and zero mean. See more in Appendix C.5 about the feature-creation process. Due to the evident correlation of features (for example, depth and size, modality and size, etc), we apply a ridge penalty to obtain interpretable coefficients for the importance of different features.

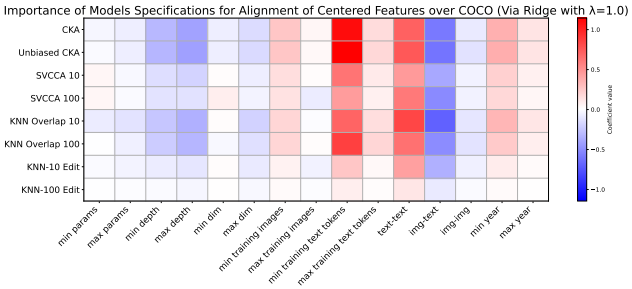


Figure 7: Coefficients of ridge regression when regressing alignment on specifications of pairs of models. The same experiment with different datasets and values of λ appears in Appendix C.5.

Analysis. Figure 7 suggests that *specifications of models have a consistent impact on alignment across metrics*. This statement supports the ordinal consistency of alignment across metrics made in [HCWI24]. Our decomposition also suggests the importance of the following model specifications. First, min training text tokens and min training images positively impact the alignment. This is

consistent with our view that the signal-to-noise ratio in representations scales with training data size; according to this view, the model that has been trained on less data is noisier. Hence, a larger min value implies that both models are less noisy. Likewise, the interpretation of the consistent positive correlation with min year is similar – if both models are more recent, we can expect them to both be closer to the Platonic ground truth. Also expected are the strong trends that text-text objectives strongly increase alignment, while different text-image objectives decrease alignment.

4 Limitations and Future Directions

One limitation of our work is that the models in our experiments are up to 4B parameters, while frontier LLMs currently have more than a trillion parameters. A second limitation comes from the fact that the current literature on SAEs still has many gaps such as polysemanticity and feature absorption [MFIM25, BNKN25], the problem of non-canonical units [LBP+25], dead-neuron problem [BTB+23], the issue for top-k-based models that k varies between objects in practice [JSG+25], and shrinkage problems on L_1-penalized SAEs [RCS+24].

Acknowledgments

We would like to thank Enric Boix-Adsera for a helpful conversation regarding Sparse Authoencoders and Sharut Gupta for many conversations on representation learning. We would also like to thank Victor Butoi and Michael Hla for helpful feedback on the exposition.

References

[AGM14] Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *Conference on Learning Theory*, pages 779–806. PMLR, 2014.

[AGMM15] Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on learning theory*, pages 113–149. PMLR, 2015.

[ALL+16] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

[BBNP25] Kiril Bangachev, Guy Bresler, Iliyas Noman, and Yury Polyanskiy. Global minimizers of sigmoid contrastive loss, 2025.

[BNKN25] Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

[BTB+23] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. Anthropic Research.

[CLM+24] Laure Ciernik, Lorenz Linhardt, Marco Morik, Jonas Dippel, Simon Kornblith, and Lukas Muttenthaler. Training objective drives the consistency of representational similarity across datasets. arXiv preprint, 2024.

[FWL+25] Thomas Fel, Binxu Wang, Michael A. Lepori, Matthew Kowal, Andrew Lee, Randall Balestriero, Sonia Joseph, Ekdeep S. Lubana, Talia Konkle, Demba Ba, and Martin Wattenberg. Into the rabbit hull: From task-relevant concepts in dino to minkowski geometry, 2025.

[GKJ+26] Sharut Gupta, Sanyam Kansal, Stefanie Jegelka, Phillip Isola, and Vikas Garg. Canonicalizing multimodal contrastive representation learning, 2026.

[GKP26] Nikhil Garg, Jon Kleinberg, and Kenny Peng. How many features can a language model store under the linear representation hypothesis?, 2026.

[GlTT+24] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

[GMM24] Florentin Guth, Brice Ménard, and Stéphane Mallat. A rainbow in deep network black boxes. *J. Mach. Learn. Res.*, 25(1), January 2024.

[GSW03] Bernhard Ganter, Gerd Stumme, and R Wille. Formal concept analysis: Methods, and applications in computer science. *TU: Dresden, Germany*, 2003.

[GSW+25] Sharut Gupta, Shobhita Sundaram, Chenyu Wang, Stefanie Jegelka, and Phillip Isola. Better together: Leveraging unpaired multimodal data for stronger unimodal models. arXiv preprint, 2025.

[GWB26] Fabian Gröger, Shuo Wen, and Maria Brbić. Revisiting the platonic representation hypothesis: An aristotelian view, 2026.

[GWLB25] Fabian Gröger, Shuo Wen, Huyen Le, and Maria Brbić. With limited data for multimodal alignment, let the structure guide you. *arXiv preprint arXiv:2506.16895*, 2025.

[HCRS+24] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024.

[HCWI24] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20617–20642. PMLR, 21–27 Jul 2024.

[IHR25] Francesco Insulla, Shuo Huang, and Lorenzo Rosasco. Towards a learning theory of representation alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

[JRR+24] Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. *arXiv preprint arXiv:2403.03867*, 2024.

[JSG+25] Sonia Joseph, Praneet Suresh, Ethan Goldfarb, Lorenz Hufe, Yossi Gandelsman, Robert Graham, Danilo Bzdok, Wojciech Samek, and Blake Aaron Richards. Steering clip's vision transformer with sparse autoencoders. *arXiv preprint arXiv:2504.08729*, 2025.

[KH25] Jeonghyeon Kim and Sangheum Hwang. Enhanced ood detection through cross-modal alignment of multi-modal representations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29979–29988, 2025.

[KKBW25] Daniel J Korchinski, Dhruva Karkada, Yasaman Bahri, and Matthieu Wyart. On the emergence of linear analogies in word embeddings. *arXiv preprint arXiv:2505.18651*, 2025.

[KMB08] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

[KNLH19] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[KXCR20] Matthew A Kelly, Yang Xu, Jesús Calvillo, and David Reitter. Which sentence embeddings and which layers encode syntactic structure? In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 42, 2020.

[KZG+17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[LBP+25] Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. *arXiv preprint arXiv:2502.04878*, 2025.

[LG14] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2177–2185, Cambridge, MA, USA, 2014. MIT Press.

[LIMB+25] Ángela López-Cardona, Sebastián Idesis, Mireia Masias-Bruns, Sergi Abadal, and Ioannis Arapakis. Brain–language model alignment: Insights into the platonic hypothesis and intermediate-layer advantage. arXiv preprint, 2025.

[LLL+24] Chengzhi Lin, Hezheng Lin, Shuchang Liu, Cangguang Ruan, LingJing Xu, Dezhao Yang, Chuyuan Wang, and Yongqi Liu. Dreaming user multimodal representation guided by the platonic representation hypothesis for micro-video recommendation. *arXiv preprint arXiv:2410.03538*, 2024.

[LMB+15] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[Lob25] Alexander Lobashev. An information-geometric view of the platonic hypothesis. In *NeurIPS 2025 Workshop on Symmetry and Geometry in Neural Representations*, 2025.

[LVAW25] Zeyu Michael Li, Hung Anh Vu, Damilola Awofisayo, and Emily Wenger. Exploring causes of representational similarity in machine learning models. *arXiv preprint arXiv:2505.13899*, 2025.

[MBV17] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[MF14] Alireza Makhzani and Brendan J. Frey. k-sparse autoencoders. In *International Conference on Learning Representations (ICLR)*, 2014.

[MFIM25] Gouki Minegishi, Hiroki Furuta, Yusuke Iwasawa, and Yutaka Matsuo. Rethinking evaluation of sparse autoencoders through the representation of polysemous words. *arXiv preprint arXiv:2501.06254*, 2025.

[MJD25] Satiyabooshan Murugaboopathy, Connor T Jerzak, and Adel Daoud. Platonic representations for poverty mapping: Unified vision-language codes or agent-induced novelty? *arXiv preprint arXiv:2508.01109*, 2025.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[MT24] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024.

[MYZ13] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

[PCV24] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[PKB+25] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[RCS+24] Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Janos Kramar, Rohin Shah, and Neel Nanda. Improving sparse decomposition of language model activations with gated sparse autoencoders. *Advances in Neural Information Processing Systems*, 37:775–818, 2024.

[RGYSD17] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[RSV+25] Joséphine Raugel, Marc Szafraniec, Huy V. Vo, Camille Couprie, Patrick Labatut, Piotr Bojanowski, Valentin Wyart, and Jean-Rémi King. Disentangling the factors of convergence between brains and computer vision models, 2025.

[SAC25] Dominik Schnaus, Nikita Araslanov, and Daniel Cremers. It's a (blind) match! towards vision–language correspondence without parallel data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[SDGS18] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[SMC+24] Vighnesh Subramaniam, David Mayo, Colin Conwell, Tomaso Poggio, Boris Katz, Brian Cheung, and Andrei Barbu. Training the untrainable: Introducing inductive bias via representational alignment. *arXiv preprint arXiv:2410.20035*, 2024.

[SPA+19] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *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*, pages 5628–5637. PMLR, 2019.

[Spe22] Robyn Speer. rspeer/wordfreq: v3.0, September 2022.

[SS20] Timo Schick and Hinrich Schütze. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8766–8774, 2020.

[TCM+24] 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. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

[Tho25] Marcus A Thomas. Towards error centric intelligence i, beyond observational learning. *arXiv preprint arXiv:2510.15128*, 2025.

[TSLL25] Yiming Tang, Harshvardhan Saini, Yizhen Liao, and Dianbo Liu. On the theoretical foundation of sparse dictionary learning in mechanistic interpretability. *arXiv preprint arXiv:2512.05534*, 2025.

[VKJ+25] Constantin Venhoff, Ashkan Khakzar, Sonia Joseph, Philip Torr, and Neel Nanda. How visual representations map to language feature space in multimodal llms. *arXiv preprint arXiv:2506.11976*, 2025.

[Wac78] Kenneth W Wachter. The strong limits of random matrix spectra for sample matrices of independent elements. *The Annals of Probability*, pages 1–18, 1978.

[WLD+25] Wei Wu, Can Liao, Zizhen Deng, Zhengrui Guo, and Jinzhuo Wang. Neuron platonic intrinsic representation from dynamics using contrastive learning. In *International Conference on Learning Representations (ICLR)*, 2025. Poster.

[XDF+25] Jingyi Xu, Xin Deng, Yibing Fu, Mai Xu, and Shengxi Li. Mdsc-net: Multi-modal discriminative sparse coding driven rgb-d classification network. *IEEE Transactions on Multimedia*, 27:442–454, 2025.

[Xio26] Bo Xiong. The lattice representation hypothesis of large language models. *arXiv preprint arXiv:2603.01227*, 2026.

[YYK25] Lauren Hyoseo Yoon, Yisong Yue, and Been Kim. Escaping plato's cave: JAM for aligning independently trained vision and language models. arXiv preprint, 2025.

[ZC25] Liu Ziyin and Isaac Chuang. Proof of a perfect platonic representation hypothesis. *arXiv preprint arXiv:2507.01098*, 2025.

[ZMKB23] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023.

A On The Statistical Model

A.1 Derivation of Bounds on Representation Components

Following the Statistical Model of Representations, $f_\theta(X) = A(\theta, f) \times (Z(X) \odot M(X, \theta, f)) + \eta_{X,\theta,f}$, we derive the bounds for the inner product $\langle f_\theta(X_1), f_\theta(X_2)\rangle$ under the assumptions from Section 1.3.

Signal Bound. The signal component is defined as the interaction between shared attributes:

$$S = \sum_{i\,:\,(Z_1)_i=(Z_2)_i=1} (M_1)_i (M_2)_i.$$

By Assumption A.3, for any active attribute i, the magnitude is bounded by $\frac{\phi_{\text{sig}}}{\sqrt{k}} \le M_i \le \frac{\Phi_{\text{sig}}}{\sqrt{k}}$. Since the number of shared attributes is exactly the inner product of the binary vectors $\langle Z_1, Z_2\rangle$, the summation contains $\langle Z_1, Z_2\rangle$ terms. Applying the magnitude bounds:

- **Lower Bound**: $S \ge \langle Z_1, Z_2\rangle \cdot \left(\frac{\phi_{\text{sig}}}{\sqrt{k}}\right)^2 = \frac{\phi_{\text{sig}}^2 \langle Z_1, Z_2\rangle}{k}$.
- **Upper Bound**: $S \le \langle Z_1, Z_2\rangle \cdot \left(\frac{\Phi_{\text{sig}}}{\sqrt{k}}\right)^2 = \frac{\Phi_{\text{sig}}^2 \langle Z_1, Z_2\rangle}{k}$.

Bias Bound. The bias component arises from the dictionary incoherence between different attribute representations:

$$B = \sum_{i\neq j\,:\,(Z_1)_i=1,(Z_2)_j=1} (M_1)_i (M_2)_j \langle A_i, A_j\rangle.$$

There are at most k^2 such pairs (i, j) since each vector is k-sparse. By Assumption A.2, $|\langle A_i, A_j\rangle| \le \epsilon_{\text{dict}}$. Using the upper bound for magnitudes $M_i \le \frac{\Phi_{\text{sig}}}{\sqrt{k}}$,

$$|B| \le k^2 \cdot \left(\frac{\Phi_{\text{sig}}}{\sqrt{k}}\right)^2 \cdot \epsilon_{\text{dict}} = k \times \epsilon_{\text{dict}} \times \Phi_{\text{sig}}^2.$$

Noise Bound. The noise component involves the interactions between the signal and the estimation error η and can be written as:

$$N = \langle f_\theta(X_1), \eta_2\rangle + \langle f_\theta(X_2), \eta_1\rangle - \langle \eta_1, \eta_2\rangle.$$

By Cauchy-Schwarz and the fact that $\|f_\theta(X_1)\|_2 = \|f_\theta(X_2)\|_2 = 1$,

- $|\langle f_\theta(X_1), \eta_2\rangle| \le \|f_\theta(X_1)\|_2 \|\eta_2\|_2 \le \epsilon_{\text{noise}}$.
- $|\langle f_\theta(X_2), \eta_1\rangle| \le \epsilon_{\text{noise}}$.
- $|\langle \eta_1, \eta_2\rangle| \le \|\eta_1\|_2 \|\eta_2\|_2 \le \epsilon_{\text{noise}}^2 < \epsilon_{\text{noise}}$.

Summing these terms and applying the triangle inequality yields the bound $|N| \le 3\epsilon_{\text{noise}}$.

A.2 Proof of Proposition 1

Recall from the Statistical Model of Representations that the inner product of two representations can be decomposed into signal (S), bias (B), and noise (N) components:

$$\langle f_\theta(X_1), f_\theta(X_2)\rangle = S + B + N$$

Recall that these components are bounded as $|B| \le k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2$ and $|N| \le 3\epsilon_{\text{noise}}$.

Proof of Part 1 (Signal Alignment) Assume $\langle Z(X_1), Z(X_2)\rangle \ge \frac{2k}{\phi_{\text{sig}}^2}\left(k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma/2\right)$.

From the lower bound of the signal component, we have $S \ge \langle Z_1, Z_2\rangle \frac{\phi_{\text{sig}}^2}{k}$. Substituting the assumption:

$$S \ge \frac{2k}{\phi_{\text{sig}}^2}\left(k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma/2\right) \cdot \frac{\phi_{\text{sig}}^2}{k} = 2(k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma/2)$$

Using the triangle inequality $\langle f_\theta(X_1), f_\theta(X_2)\rangle \geq S - |B| - |N|$ and applying the upper bounds for bias and noise:

$$\begin{aligned}\langle f_\theta(X_1), f_\theta(X_2)\rangle &\geq 2k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 6\epsilon_{\text{noise}} + \gamma - k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 - 3\epsilon_{\text{noise}} \\ &= k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma\end{aligned}$$

Proof of Part 2 (Alignment Signal) Assume $\langle f_\theta(X_1), f_\theta(X_2)\rangle \geq k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma$. Isolating the signal component, we have $S = \langle f_\theta(X_1), f_\theta(X_2)\rangle - B - N$. By the triangle inequality:

$$S \geq \langle f_\theta(X_1), f_\theta(X_2)\rangle - |B| - |N|$$

Substituting the hypothesis and the component bounds:

$$S \geq (k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 + 3\epsilon_{\text{noise}} + \gamma) - k\epsilon_{\text{dict}}\Phi_{\text{sig}}^2 - 3\epsilon_{\text{noise}} = \gamma$$

From the signal upper bound $S \leq \frac{\Phi_{\text{sig}}^2\langle Z_1, Z_2\rangle}{k}$, we conclude:

$$\langle Z(X_1), Z(X_2)\rangle \geq \frac{k}{\Phi_{\text{sig}}^2} S \geq \frac{k}{\Phi_{\text{sig}}^2}\gamma.$$

B Pseudo-Code

B.1 Top-K SAE

Algorithm 1 Top-K Sparse Autoencoder with Dead Neuron Resampling

Require: dataset $X \in \mathbb{R}^{N \times d_{\text{model}}}$, hidden size d_{sparse}, sparsity level k, batch size B, number of steps T, learning rate η, weight decay λ, resampling period R, print period P, optional decoder renormalization

Ensure: trained parameters $(W_e, b_e, W_d, b_{\text{dec}})$

Initialize encoder $W_e \in \mathbb{R}^{d_{\text{sparse}} \times d_{\text{model}}}$ and $b_e \in \mathbb{R}^{d_{\text{sparse}}}$
Initialize decoder $W_d \in \mathbb{R}^{d_{\text{model}} \times d_{\text{sparse}}}$ with Kaiming-uniform initialization
Initialize decoder bias $b_{\text{dec}} \in \mathbb{R}^{d_{\text{model}}}$ to 0
Set $b_{\text{dec}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} X_i$ ▷ data mean initialization
Initialize AdamW optimizer on $(W_e, b_e, W_d, b_{\text{dec}})$ with learning rate η and weight decay λ
Initialize activity counter $c \in \mathbb{R}^{d_{\text{sparse}}}$ to 0
for $t = 1$ to T **do**
    Sample indices $\mathcal{I} \sim \text{Unif}\{1, \ldots, N\}$ of size B
    $x \leftarrow X[\mathcal{I}]$
    ▷ Forward pass
    $\tilde{x} \leftarrow x - b_{\text{dec}}$
    $f \leftarrow \text{ReLU}(W_e \tilde{x} + b_e)$
    For each row of f, keep only its top-k values:

$$z_{ij} \leftarrow \begin{cases} f_{ij}, & \text{if } f_{ij} \text{ is among the top-}k \text{ entries in row } i \\ 0, & \text{otherwise} \end{cases}$$

    $\hat{x} \leftarrow W_d z + b_{\text{dec}}$
    ▷ Optimization step
    $\mathcal{L}_{\text{recon}} \leftarrow \text{MSE}(\hat{x}, x)$
    Take one AdamW step on $\mathcal{L}_{\text{recon}}$
    if decoder renormalization is enabled **then**
        for $j = 1$ to d_{sparse} **do**
            $W_d[:, j] \leftarrow W_d[:, j] / \max(\|W_d[:, j]\|_2, \varepsilon)$
        end for
    end if
    ▷ Track feature activity
    $c \leftarrow c + \sum_{i=1}^{B} \mathbf{1}[z_i > 0]$
    if $t \bmod R = 0$ **and** $t < 0.8T$ **then**
        $m \leftarrow \text{RESAMPLEDEADNEURONS}(X, c, W_e, b_e, W_d, b_{\text{dec}}, k)$
        $c \leftarrow 0$
    end if
end for
return $(W_e, b_e, W_d, b_{\text{dec}})$

Algorithm 2 Dead Neuron Resampling

function RESAMPLEDEADNEURONS($X, c, W_e, b_e, W_d, b_{\text{dec}}, k$)
  $\mathcal{D} \leftarrow \{j \in \{1, \ldots, d_{\text{sparse}}\} : c_j = 0\}$ ▷ dead neurons
  $m \leftarrow |\mathcal{D}|$
  if $m = 0$ **then**
    return 0
  end if
  Sample indices $\mathcal{I} \sim \text{Unif}\{1, \ldots, N\}$ of size m
  $x \leftarrow X[\mathcal{I}]$
  ▷ Compute residuals under current model
  $\tilde{x} \leftarrow x - b_{\text{dec}}$
  $f \leftarrow \text{ReLU}(W_e \tilde{x} + b_e)$
  Form top-k sparse codes z by keeping only the top-k entries per row of f
  $\hat{x} \leftarrow W_d z + b_{\text{dec}}$
  $r \leftarrow x - \hat{x}$
  ▷ Normalize residuals and use them to reset dead features
  for $q = 1$ to m **do**
    $u_q \leftarrow r_q / (\|r_q\|_2 + 10^{-8})$
    Let $j = \mathcal{D}[q]$
    $W_d[:, j] \leftarrow u_q$
    $W_e[j, :] \leftarrow 0.1\, u_q$
    $b_e[j] \leftarrow 0$
  end for
  return m
end function

C Further Experiments and Experimental Details

We reproduce all experiments from the main paper for further metrics, namely CKA, Unbiased CKA, SVCCA in dimension 10, SVCCA in dimension 100, KNN-10 neighborhood Overlap, KNN-100 neighborhood Overlap, KNN-10 neighborhood edit distance, KNN-100 neighborhood edit distance. We note that for all metrics, larger values mean more similar except for the two edit distances where larger value means less similar.

C.1 Signal: Alignment of Sparse Features

C.1.1 Experiments on COCO

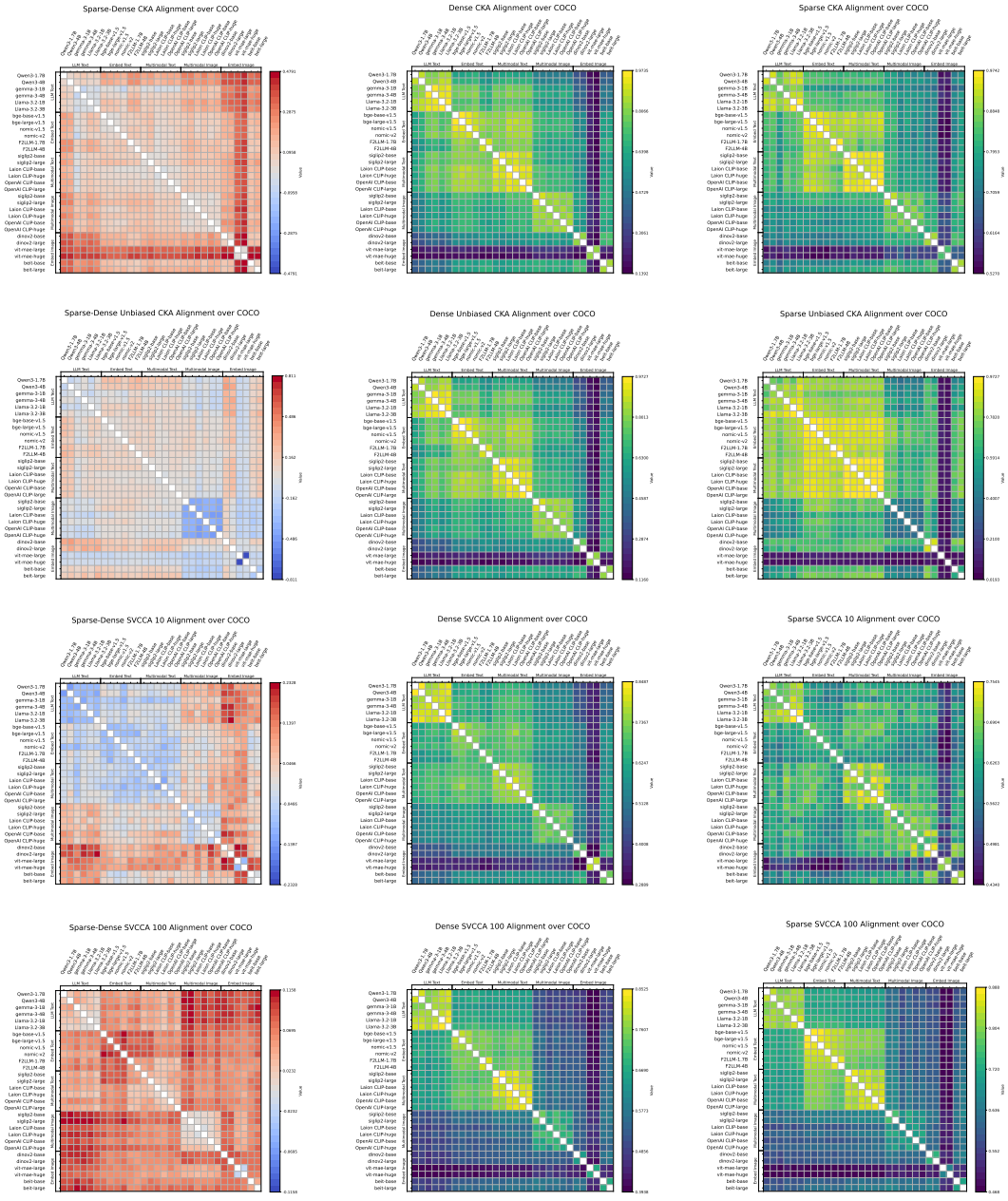


Figure 8: Same plot as in Figure 5 but for the CKA, Unbiased CKA, SVCCA 10, and SVCCA 100 metrics. In addition to differences, we plot the alignment values for raw dense features and for the sparse features. The SAE dimension is 16384. We do an ablation study with dimension 8192.

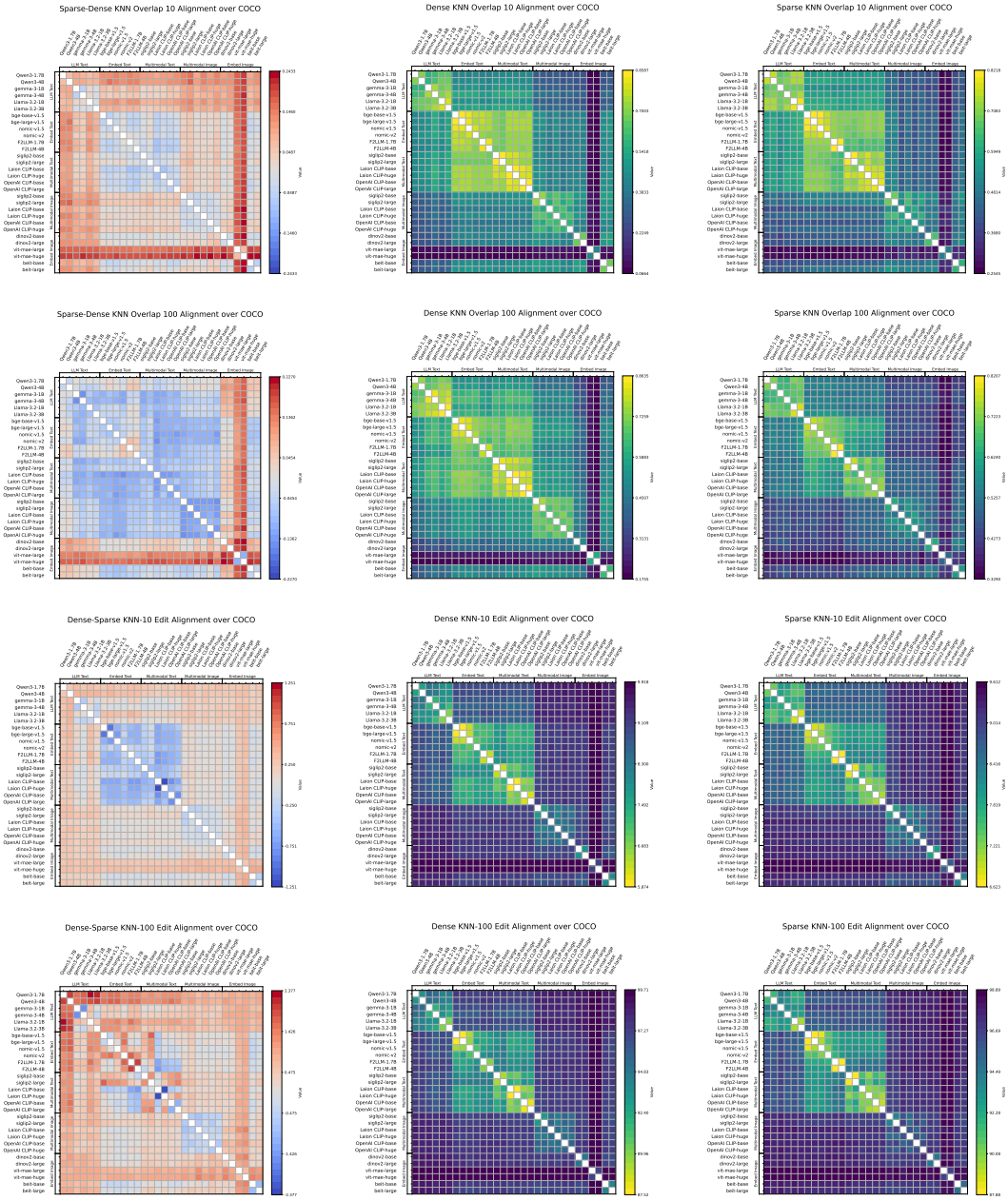


Figure 9: Same plot as in Figure 8 but for the KNN-10 Overlap, KNN-100 Overlap, KNN-10 Edit Distance, and KNN-100 Edit distance metrics. In addition to differences, we plot the alignment values for raw dense features and for the sparse features.

We choose not to interpret spectral and geometric metrics for sparse features such as CKA, unbiased CKA, SVCCA. The reason is that sparse features are trained to be non-negative and sparse, both of which induce a strong geometric bias. Instead, ordinal metrics such as KNN overlap and edit distance do not suffer from this issue.

We point out the trend that the KNN-100 overlap similarity consistently decreased with the introduction of sparse features. We attribute this to the local nature of PRH, as discovered in [GWB26] and elaborated in our Proposition 1. When $k = 100$, for subsets of only $n = 1000$ samples, the KNN graph captures a rather global geometry. We also comment on the KNN-10 overlap similarity in Figure 9 that regions where the improvement of alignment by the introduction of sparse features is low mostly correspond to regions where the alignment is already extremely high – between LLM models (rows 1-6) and between (multimodal) text embeddings (rows 7 - 18).

C.1.2 Experiments on CC3M

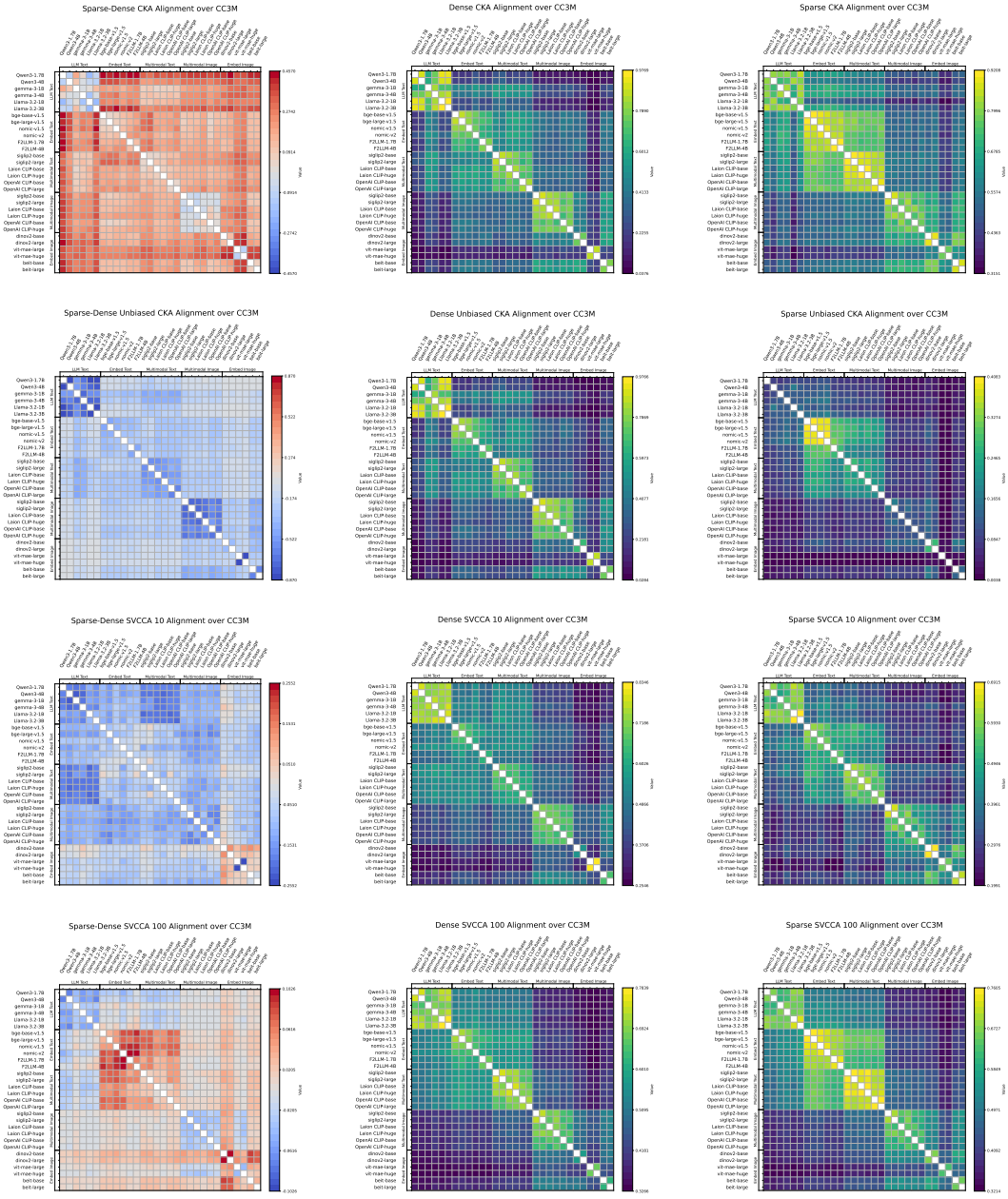


Figure 10: Same plot as in Figure 8 but over CC3M.

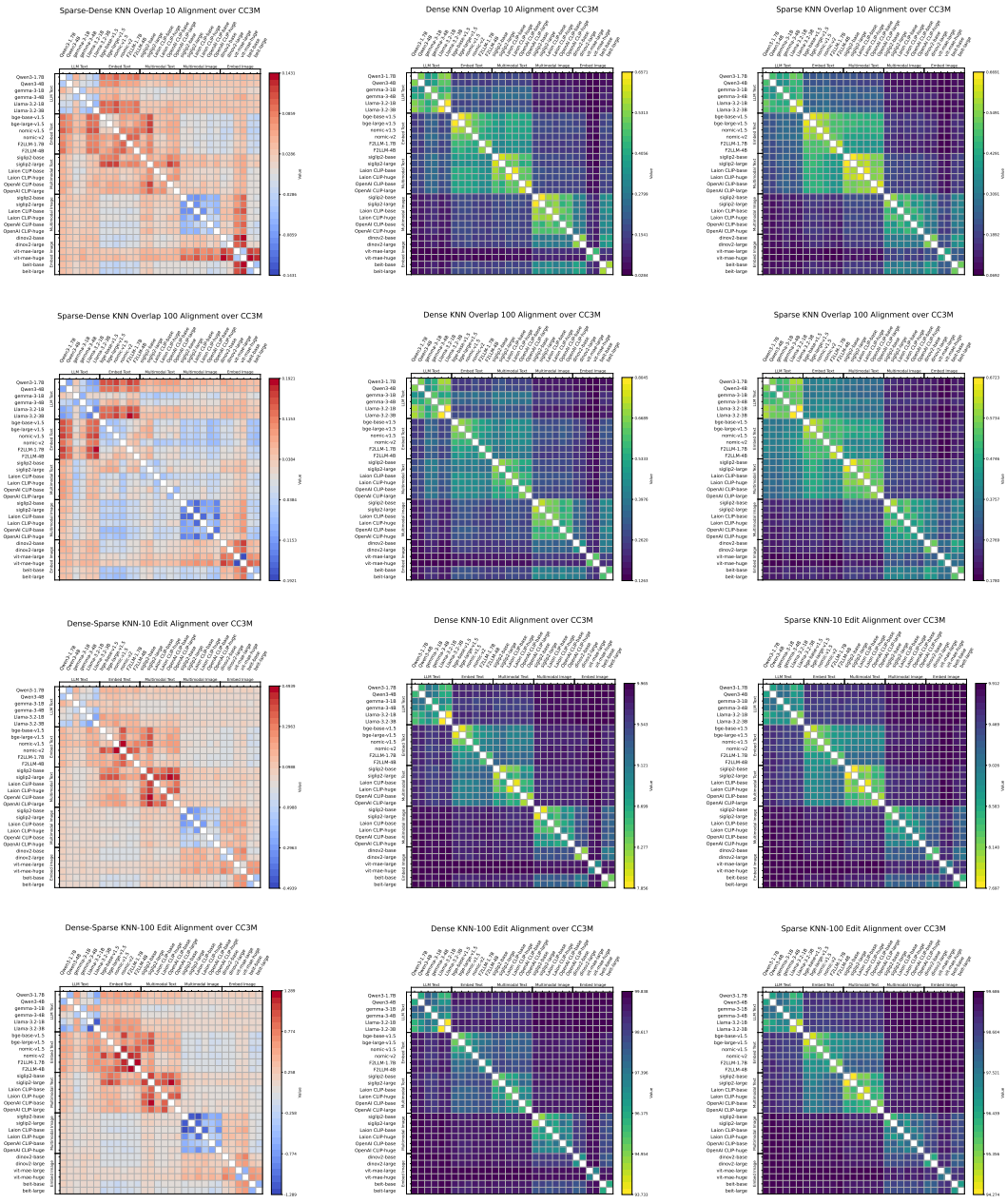


Figure 11: Same plot as in Figure 9 but over CC3M.

C.1.3 Experiments on Visual Genome

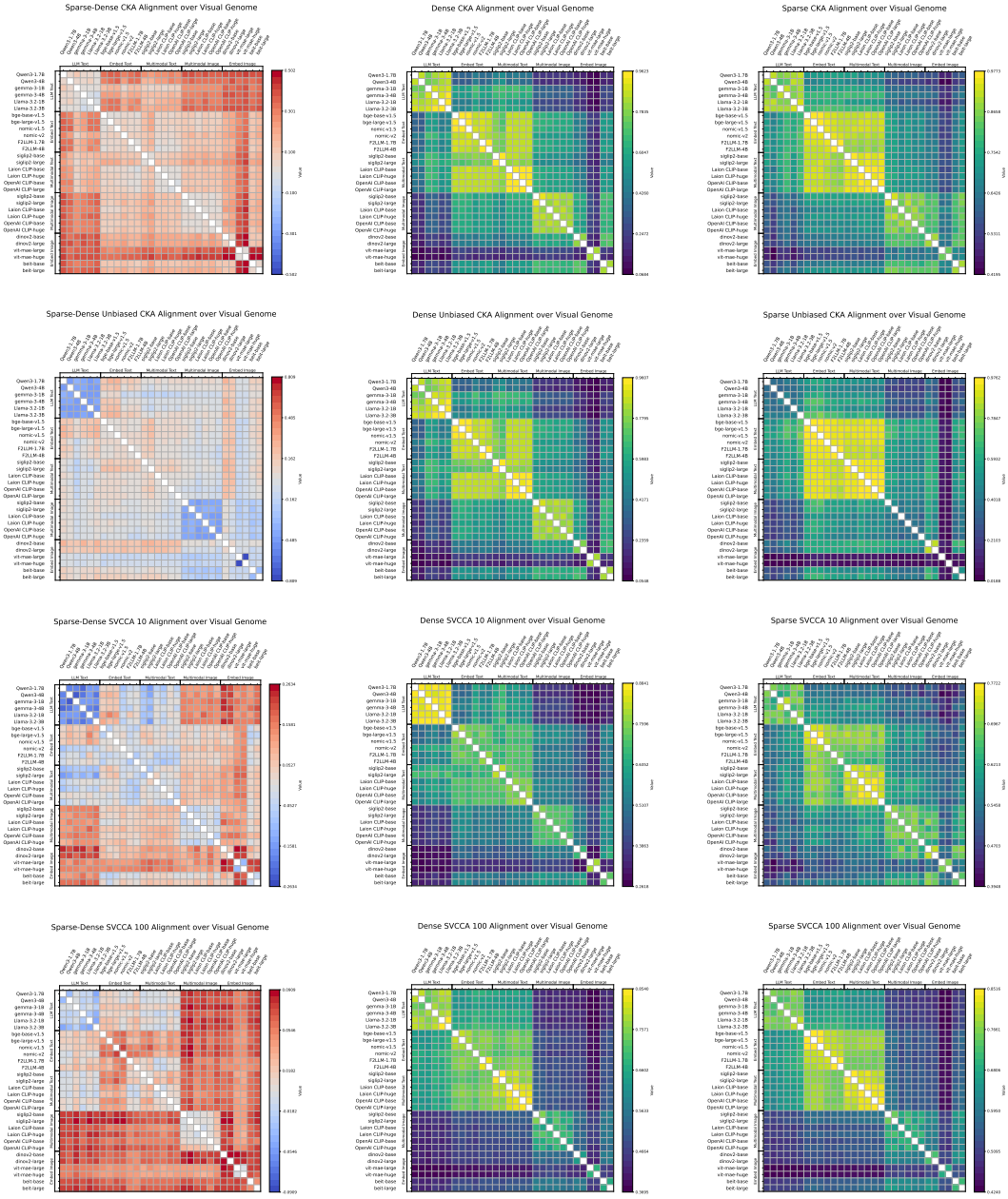


Figure 12: Same plot as in Figure 8 but over Visual Genome.

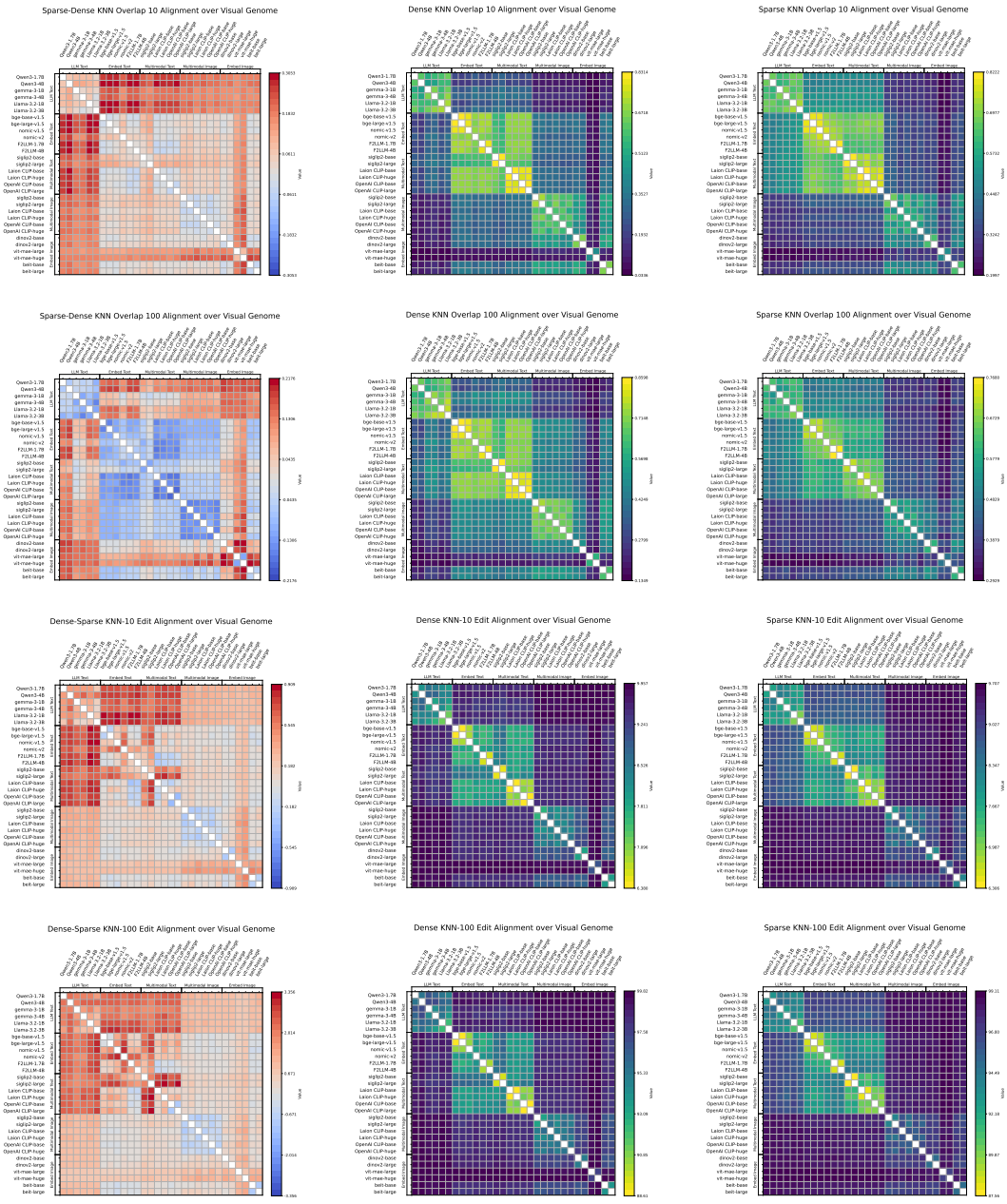


Figure 13: Same plot as in Figure 9 but over Visual Genome.

C.1.4 SAE Dimension Ablation: Experiments on COCO

We provide an ablation study where we change the SAE dimension to $d = 8192$. We present the experiments only for COCO since there is no qualitative or quantitative difference from $d = 16384$.

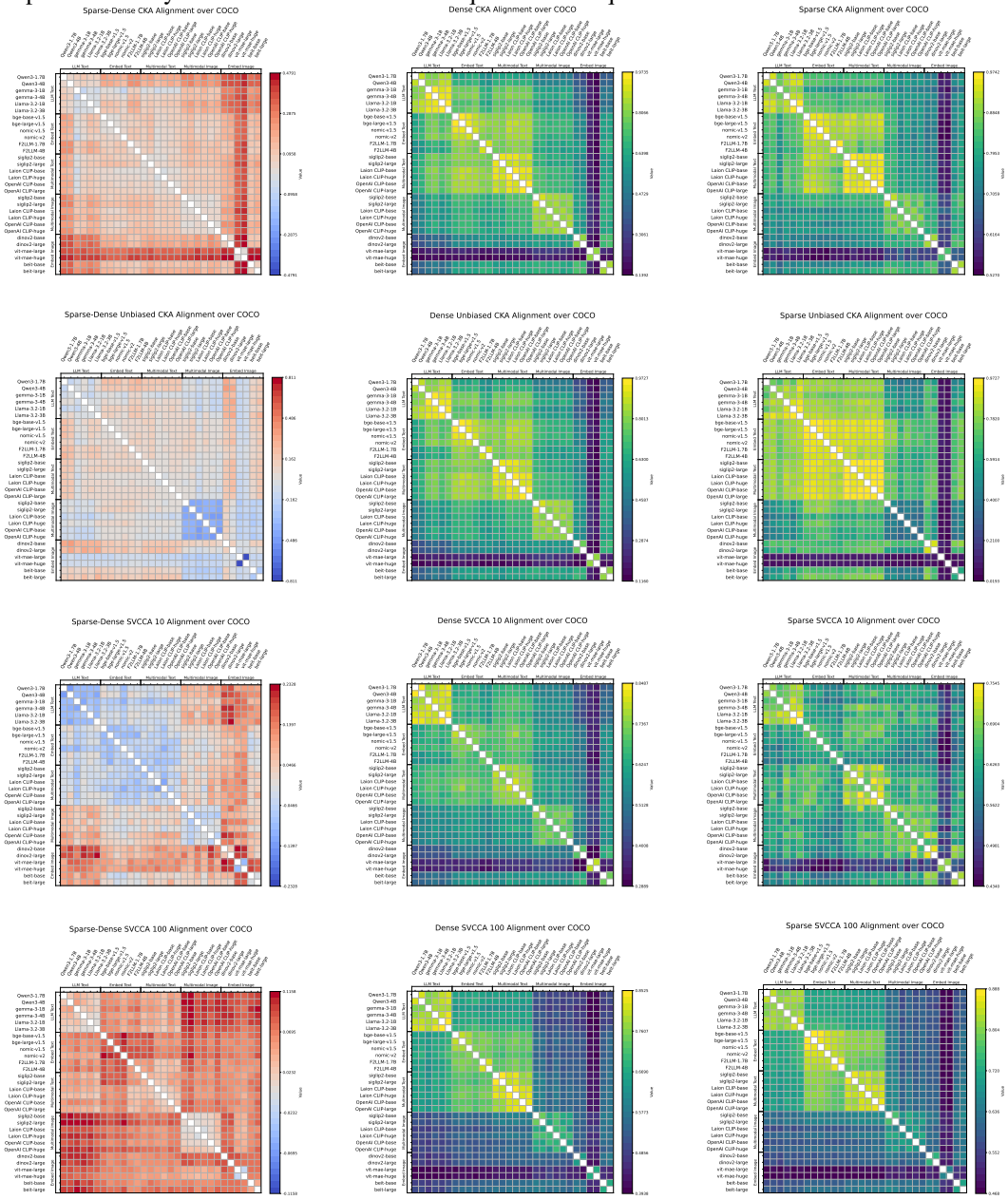


Figure 14: Same plot as in Figure 5 but for the CKA, Unbiased CKA, SVCCA 10 and SVCCA 100 metrics. In addition to differences, we plot the alignment values for raw dense features and for the sparse features. We do an ablation study with dimension 8192.

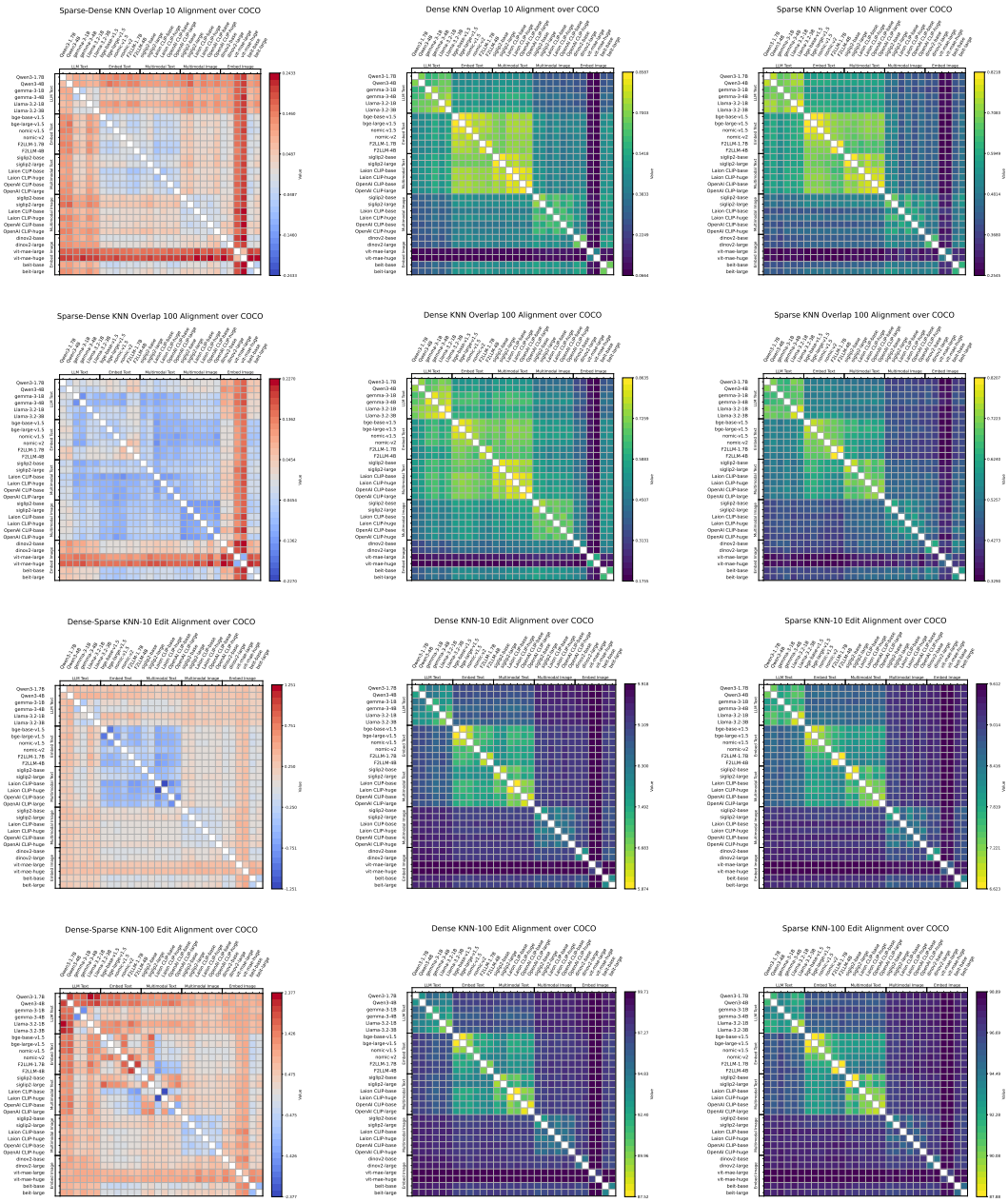


Figure 15: Same plot as in Figure 9 but with $d = 8192$.

C.2 Signal: Correlation of Sparse Features

C.2.1 Experiments on COCO

With varying k. 32 for text models, 64 for multimodal models, 128 for image-only models.

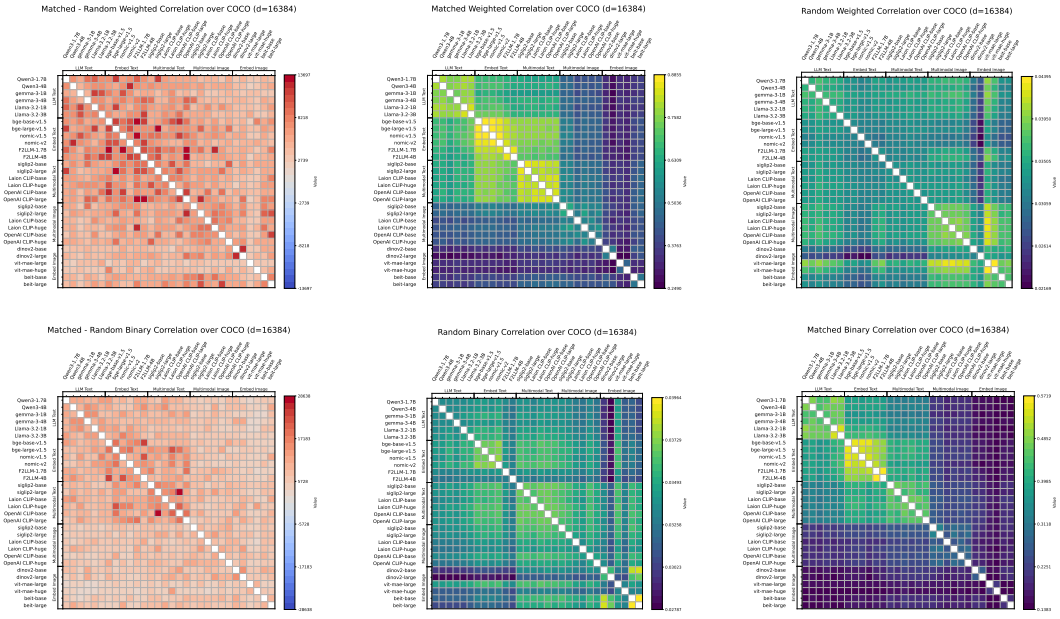


Figure 16: Plotting Experiment 2 from Section 3.1. Concretely, from left to right, we have $[\mathsf{corr}(Z_1, Z_2) - \mathbb{E}_\Phi \mathsf{corr}_\Phi(Z_1, Z_2)]/\mathbb{V}_\Phi(\mathsf{corr}_\Phi(Z_1, Z_2))^{1/2}$, then in the middle plot $\mathsf{corr}(Z_1, Z_2)$ and in the right-most plot $\mathbb{E}_\Phi \mathsf{corr}_\Phi(Z_1, Z_2)$. Here, $\mathbb{E}, \mathbb{V}$ correspond to variance and expectation.

C.2.2 Experiments on CC3M

With varying k. 32 for text models, 64 for multimodal models, 128 for image-only models.

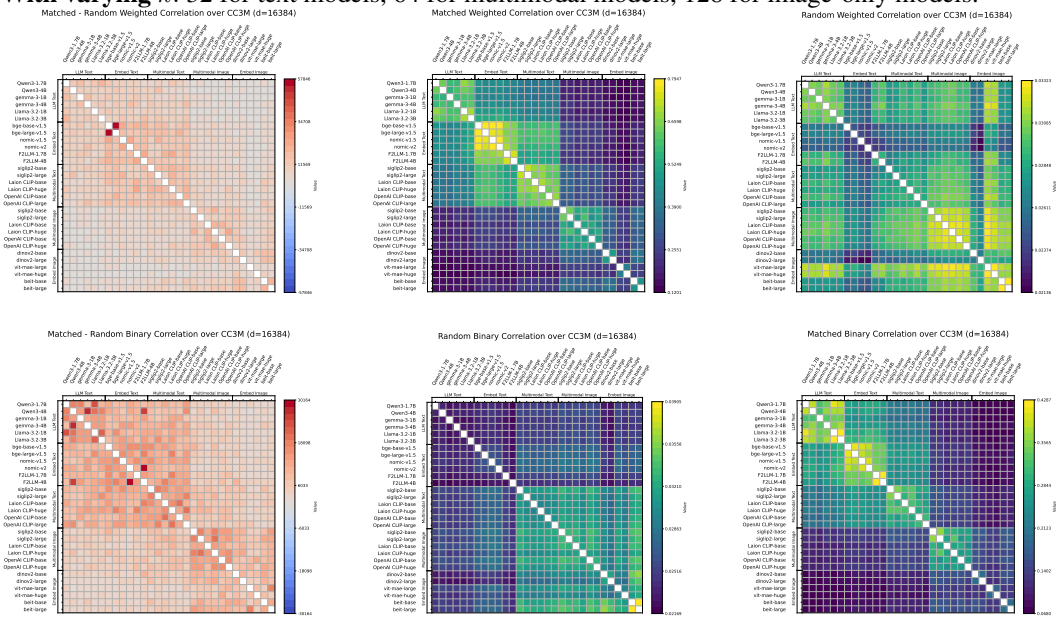


Figure 17: Same plot as in Figure 16 but over CC3M.

C.2.3 Experiments on Visual Genome

With varying k. 32 for text models, 64 for multimodal models, 128 for image-only models.

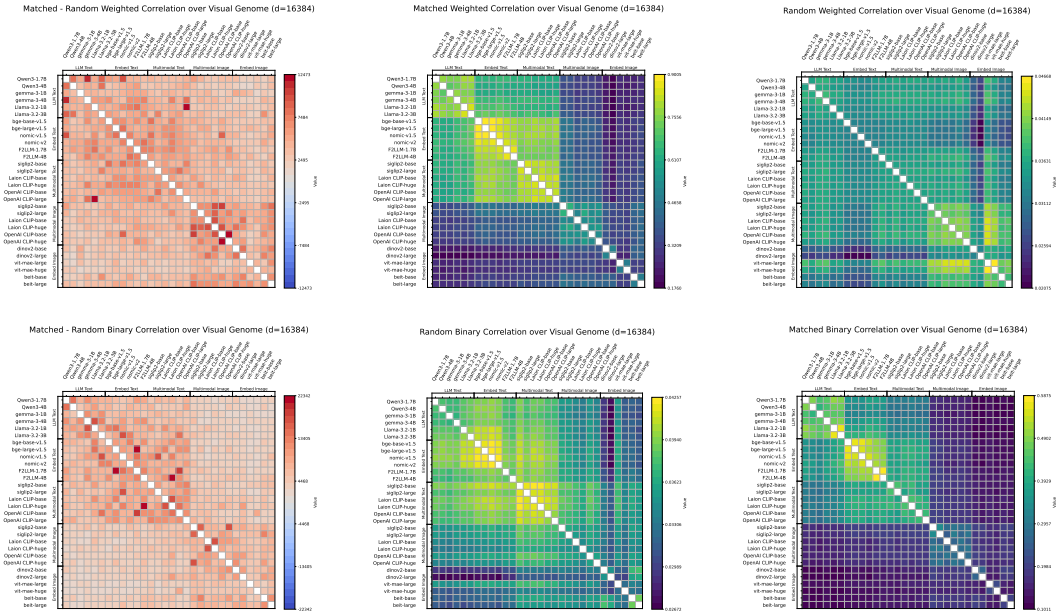


Figure 18: Same plot as in Figure 16 but over Visual Genome.

C.3 Bias: Centering Improves Alignment

C.3.1 Experiments on COCO

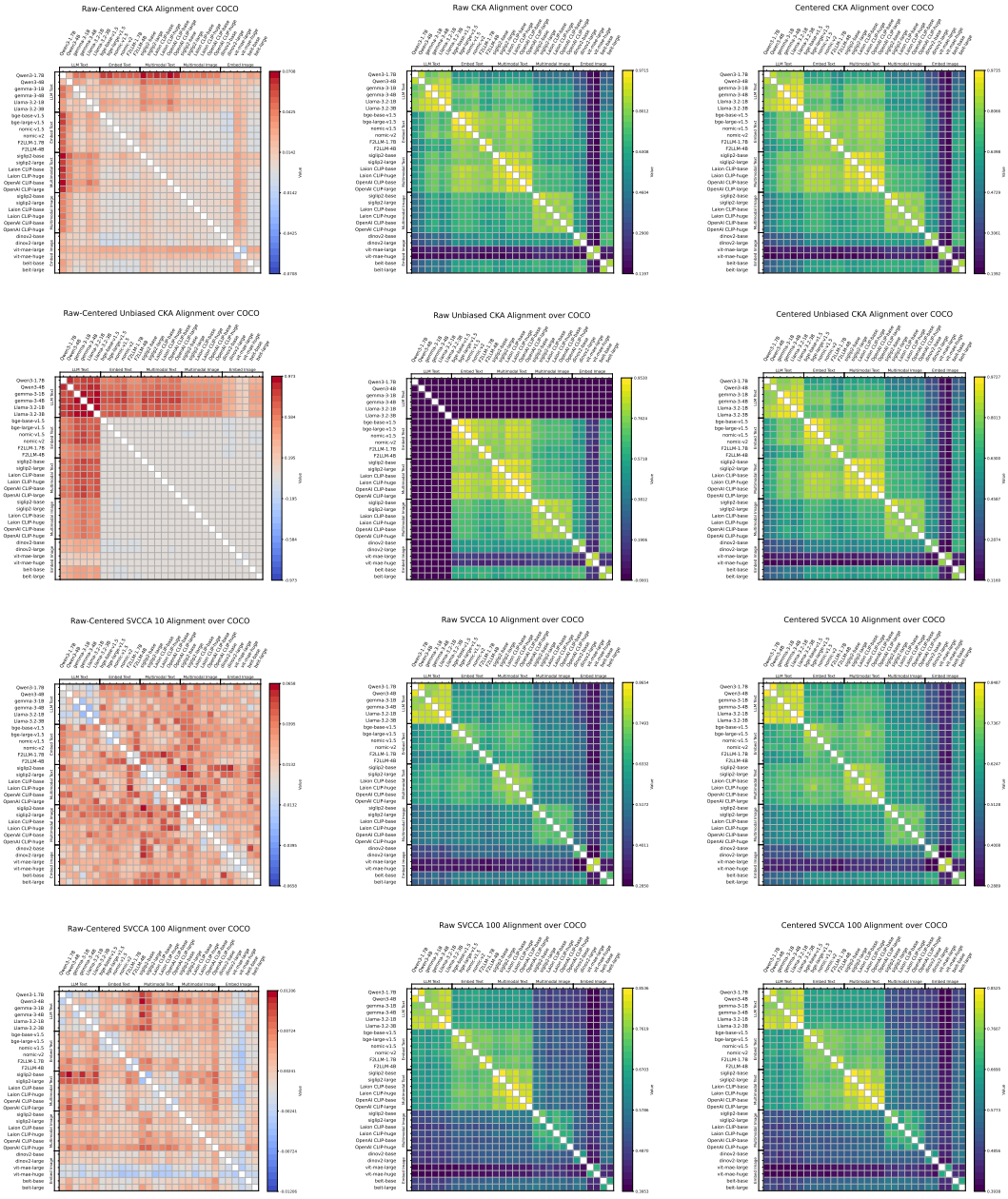


Figure 19: Same plot as in Figure 6 but for the CKA, Unbiased CKA, SVCCA 10 and SVCCA 100 metrics. In addition to differences, we plot the alignment values for raw dense features and for the sparse features.

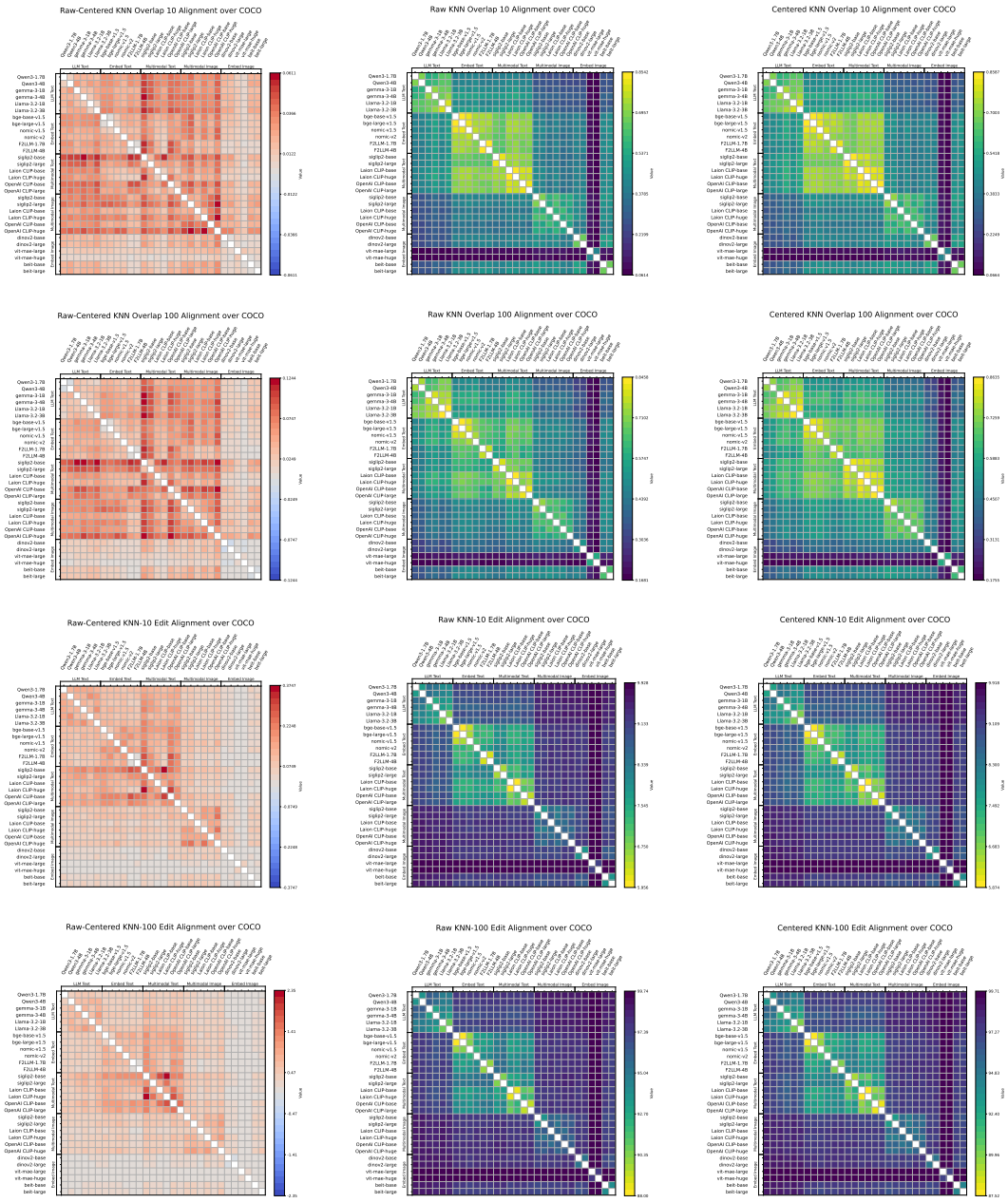


Figure 20: Same plot as in Figure 6 but for the KNN-10 Overlap, KNN-100 Overlap, KNN-10 Edit Distance, and KNN-100 Edit distance metrics. In addition to differences, we plot the alignment values for raw features and for the centered features.

We remark how consistent the improvement is across metrics. For nearly all pairs of models and metrics, there is improvement in the alignment after centering and re-normalizing.

C.3.2 Experiments on CC3M

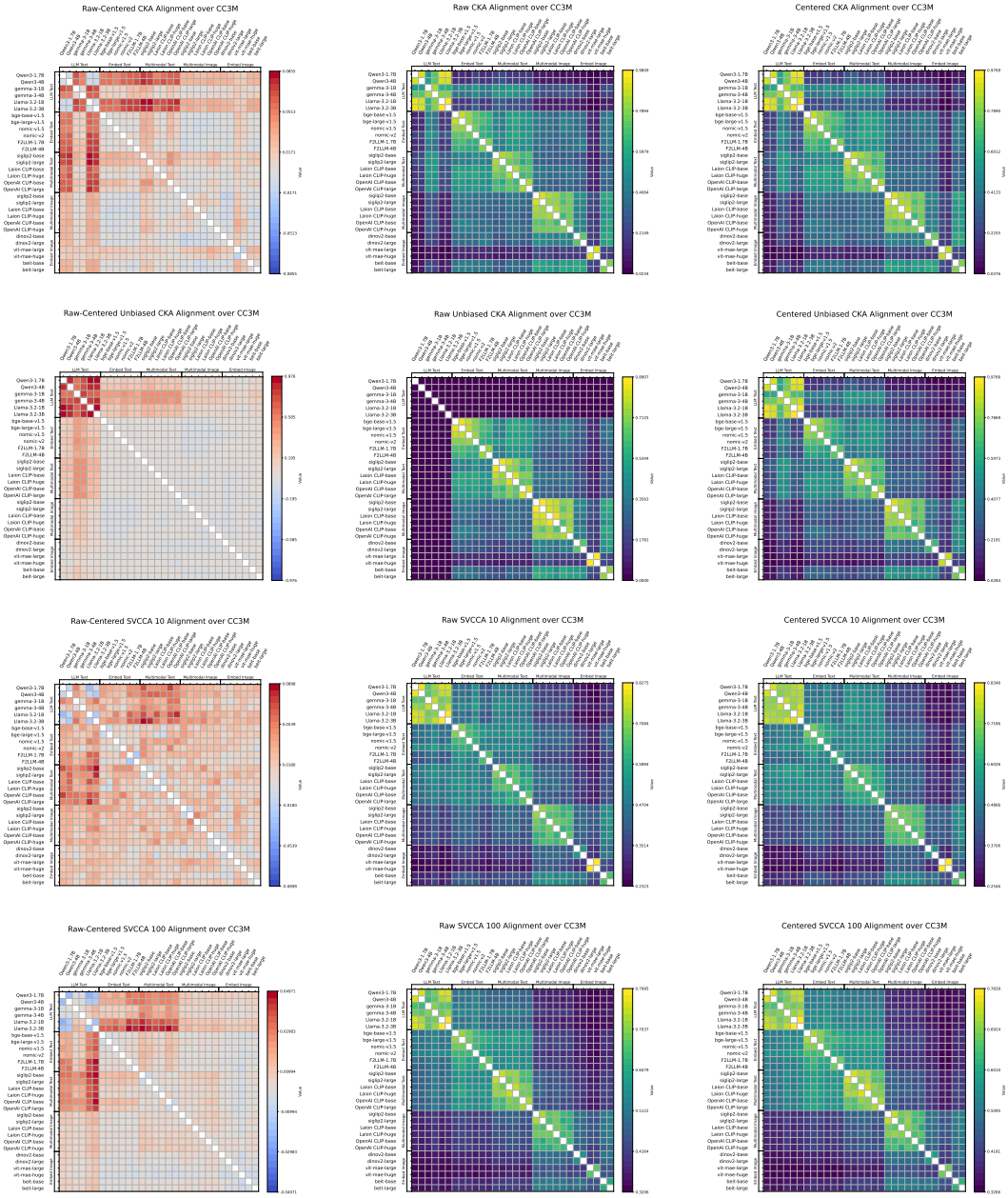


Figure 21: Same plot as in Figure 19 but over CC3M.

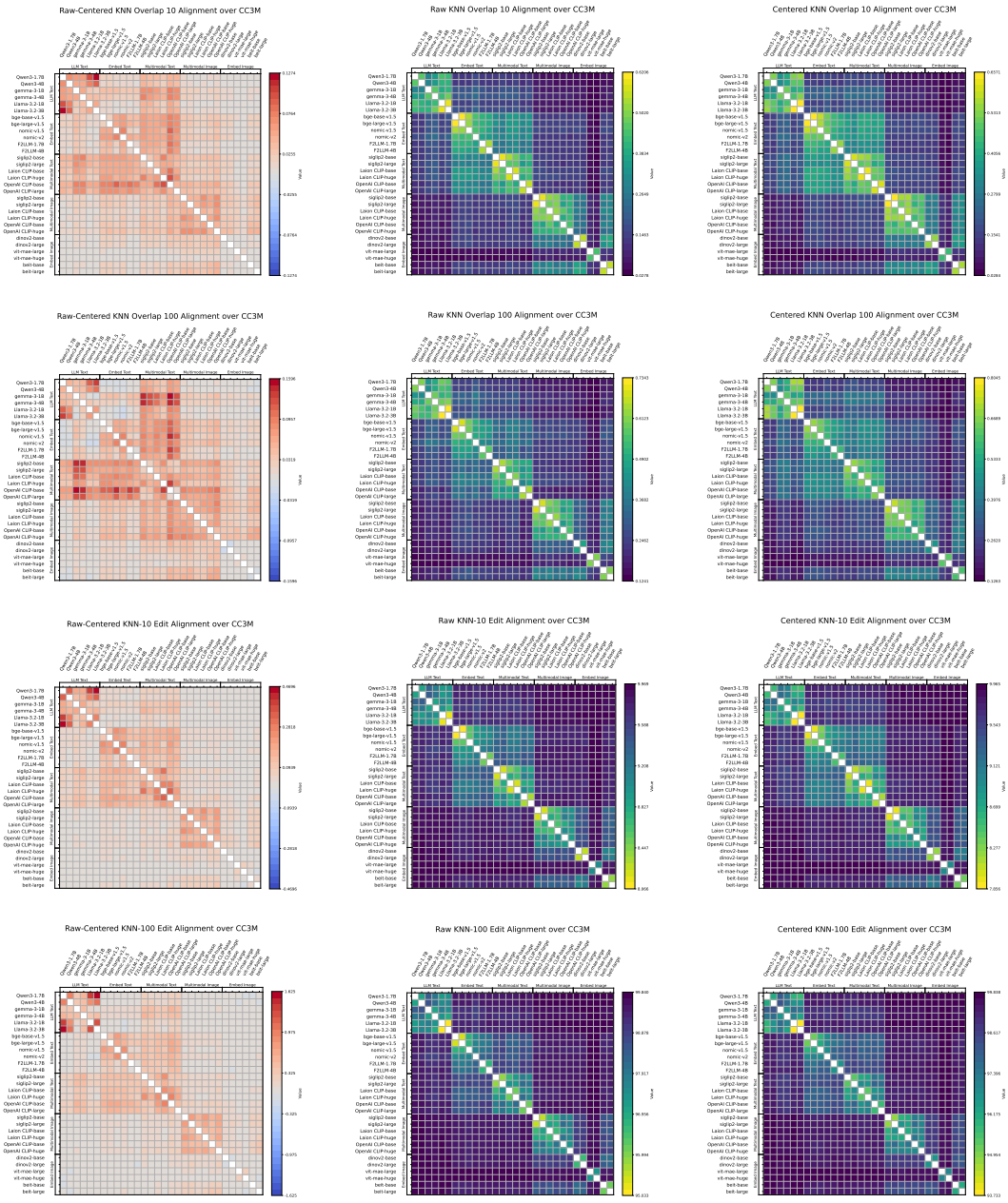


Figure 22: Same plot as in Figure 6 but over CC3M.

C.3.3 Experiments on Visual Genome

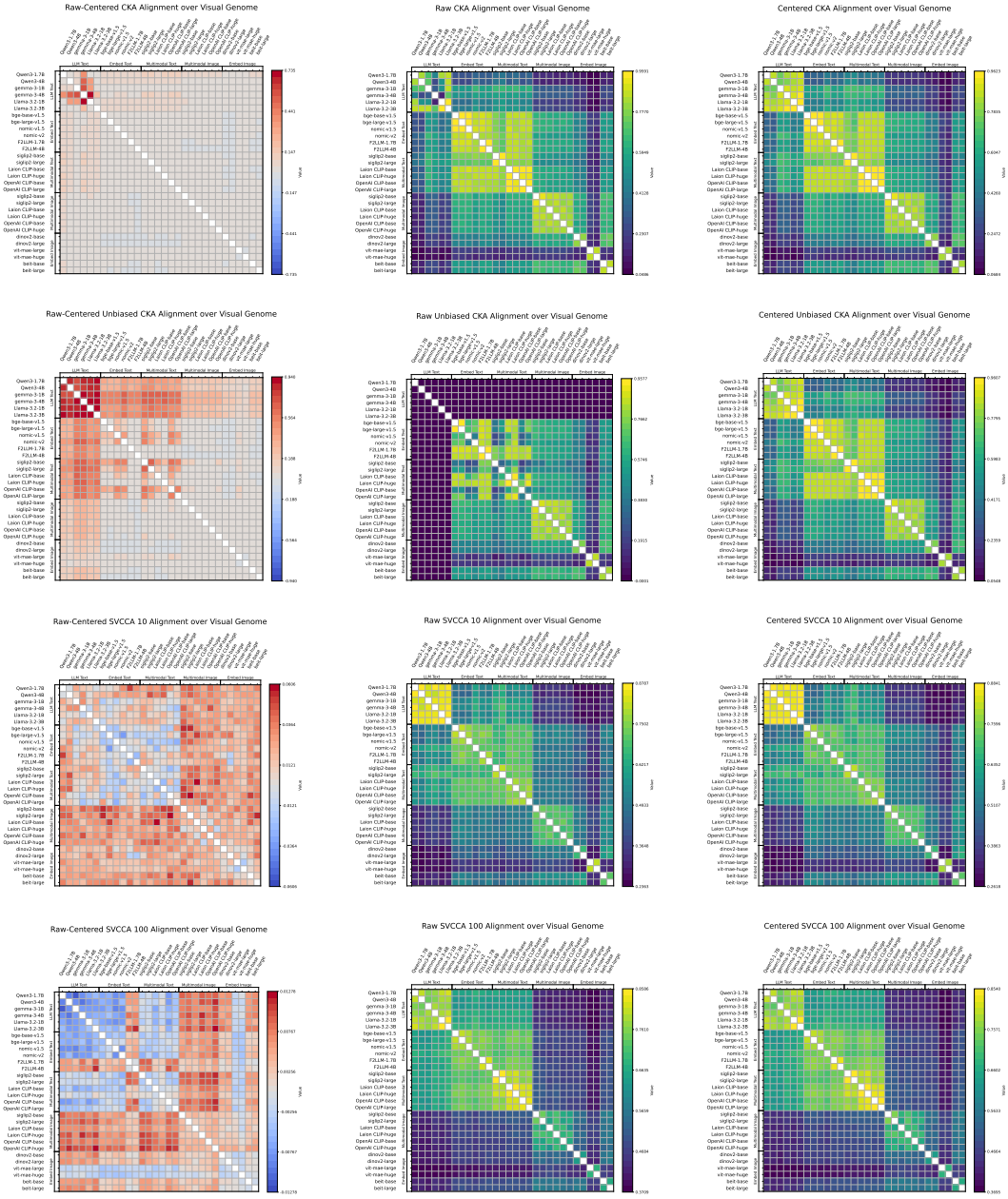


Figure 23: Same plot as in Figure 19 but over Visual Genome.

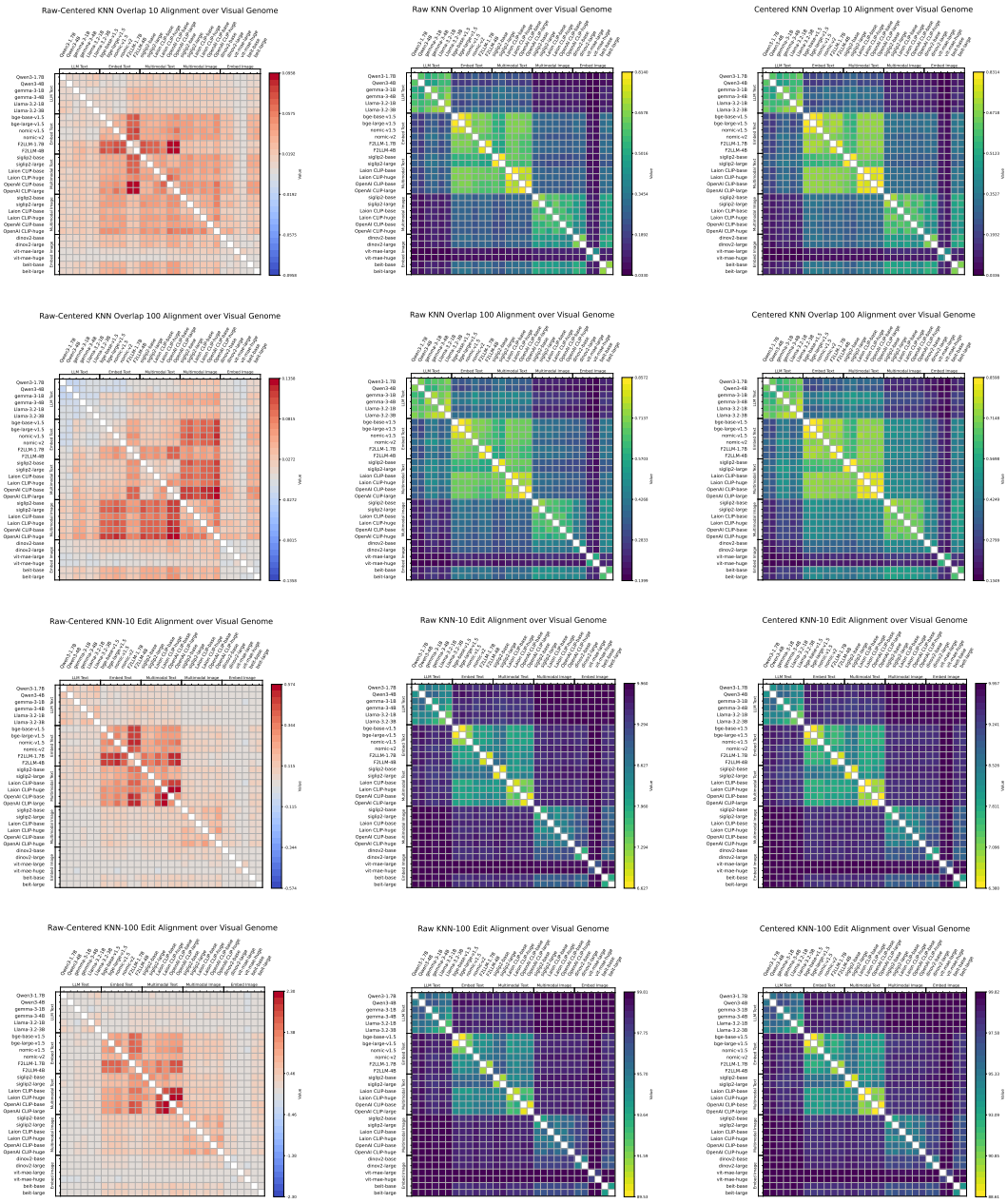


Figure 24: Same plot as in Figure 6 but over Visual Genome.

C.4 Noise: Higher Alignment for More Frequent Data

We reproduce the experiment from Figure 3 for other metrics as well, even though again our focus is on the KNN-10 overlap metric due to the local nature of the Platonic alignment observed in [GWB26].

C.4.1 Text vs LLM

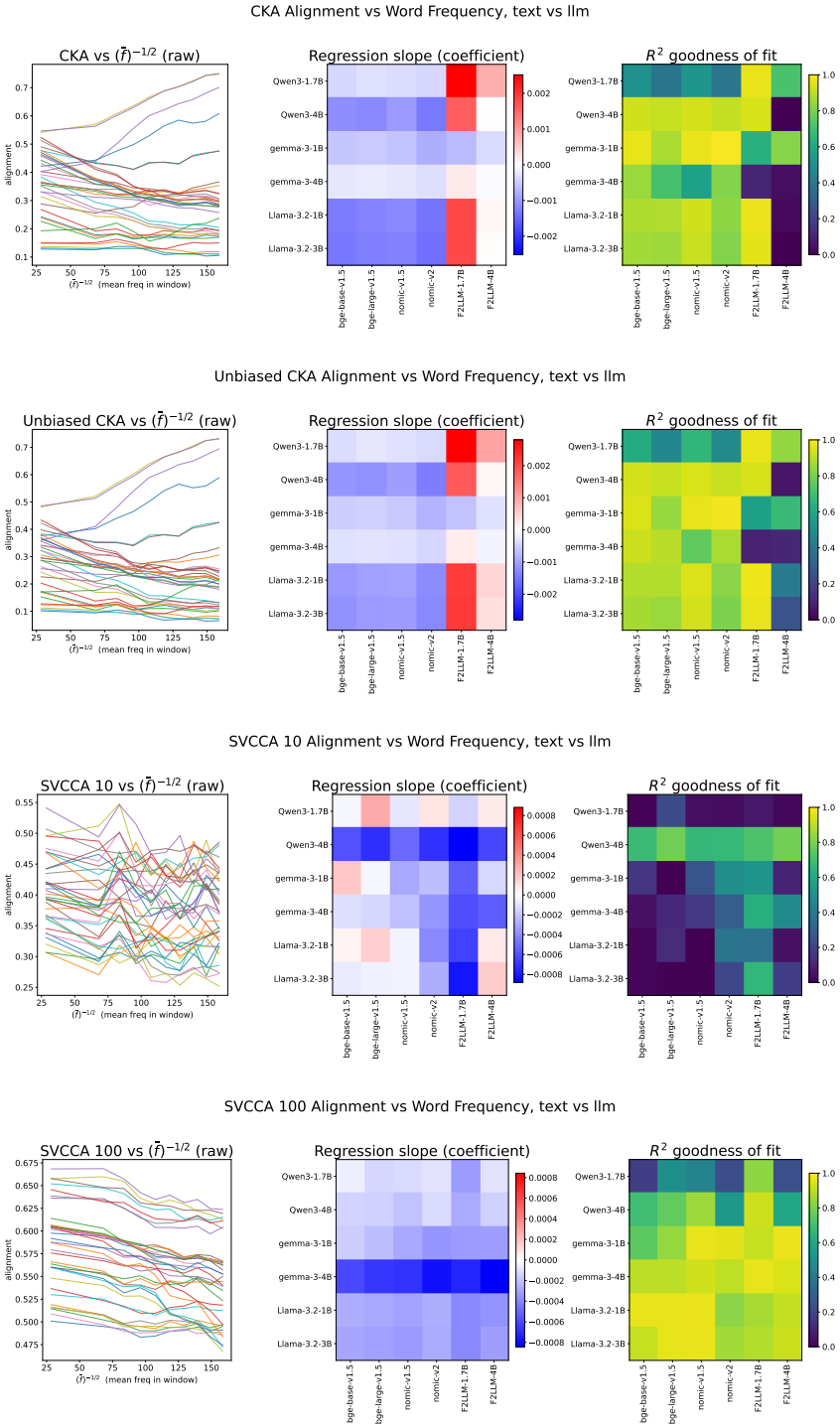


Figure 25: Same plot as in Figure 3 but for the CKA, Unbiased CKA, SVCCA 10 and SVCCA 100.

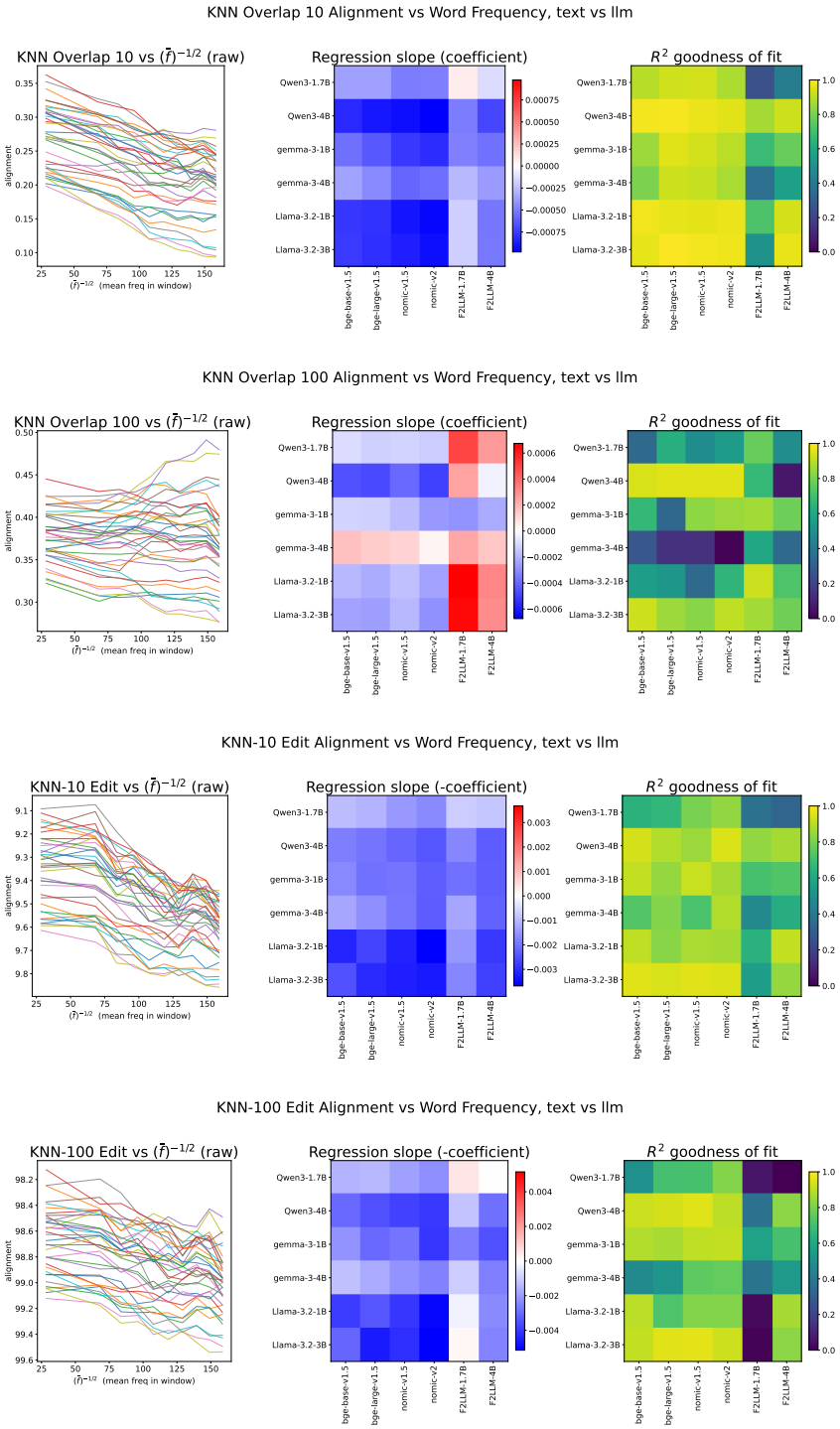


Figure 26: Same plot as in Figure 3 but for the KNN-10 Overlap, KNN-100 Overlap, KNN-10 Edit Distance, and KNN-100 Edit distance metrics. For the two KNN neighborhood edit distances, we plot the negative regression coefficients as higher values mean lower similarity unlike for other metrics.

We can see that with the exception of the SVCCA-10 metric, the nearly linear negative relationship between alignment and inverse square-root mean frequency is very consistent.

C.4.2 Text vs Text

For the case of text-text and LLM-LLM models, we skip models from the same family as they are trained on similar data, often distilled from bigger models etc., which makes the noises terms highly correlated and the alignment trends uninterpretable.

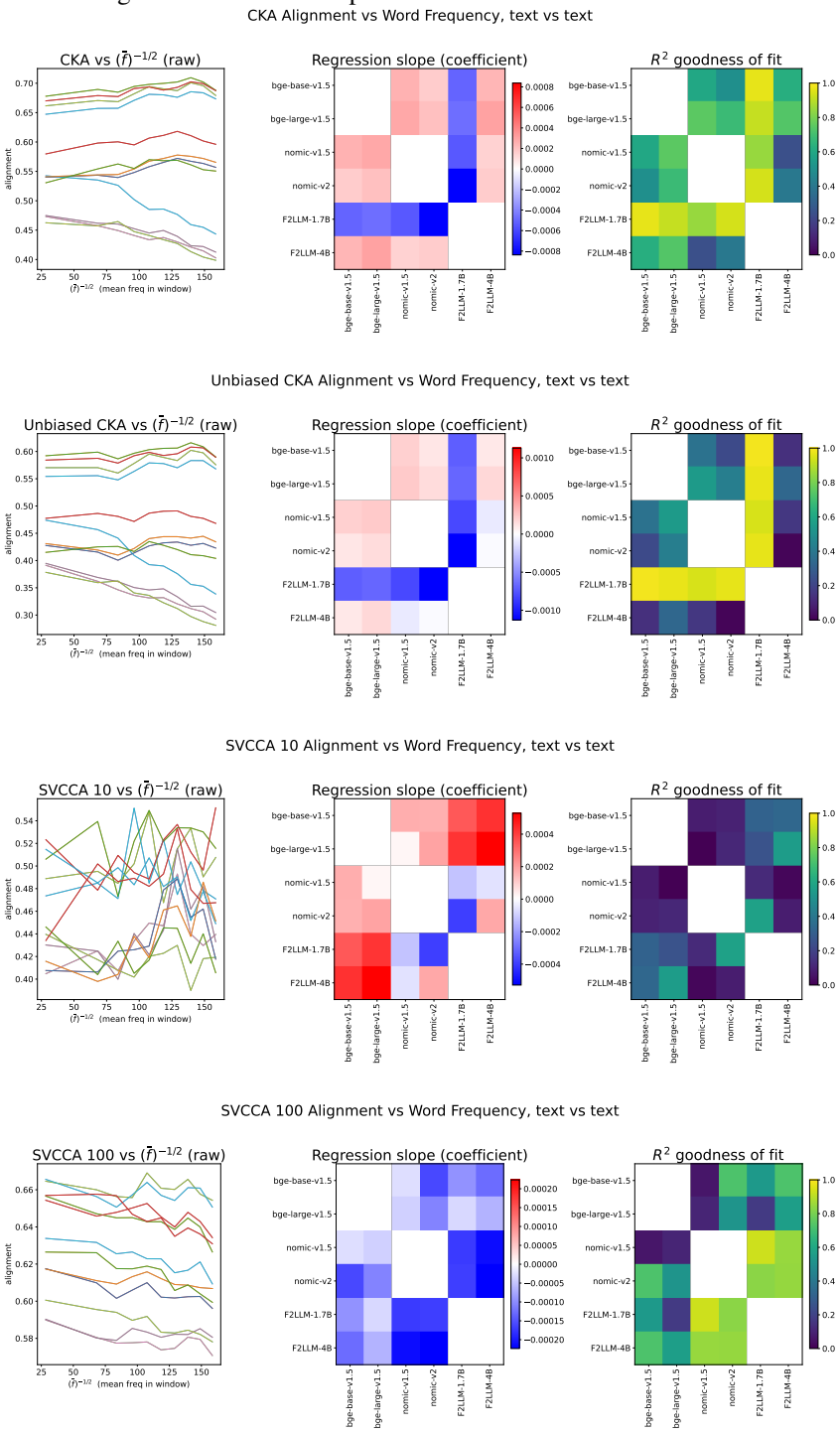


Figure 27: Same plot as in Figure 25 but for text embedding vs text embedding models.

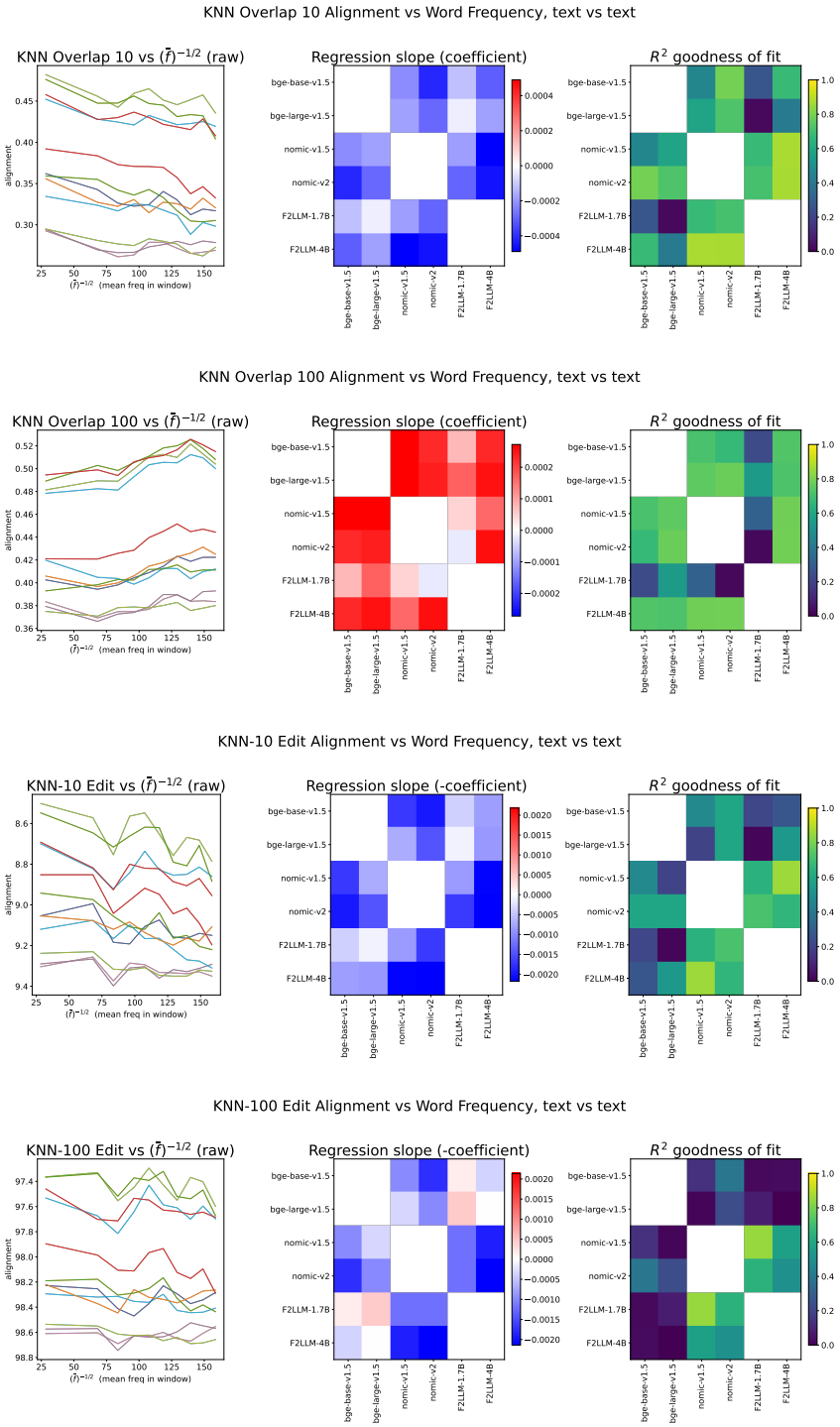


Figure 28: Same plot as in Figure 26 but for text embedding vs text embedding models.

C.4.3 LLM vs LLM

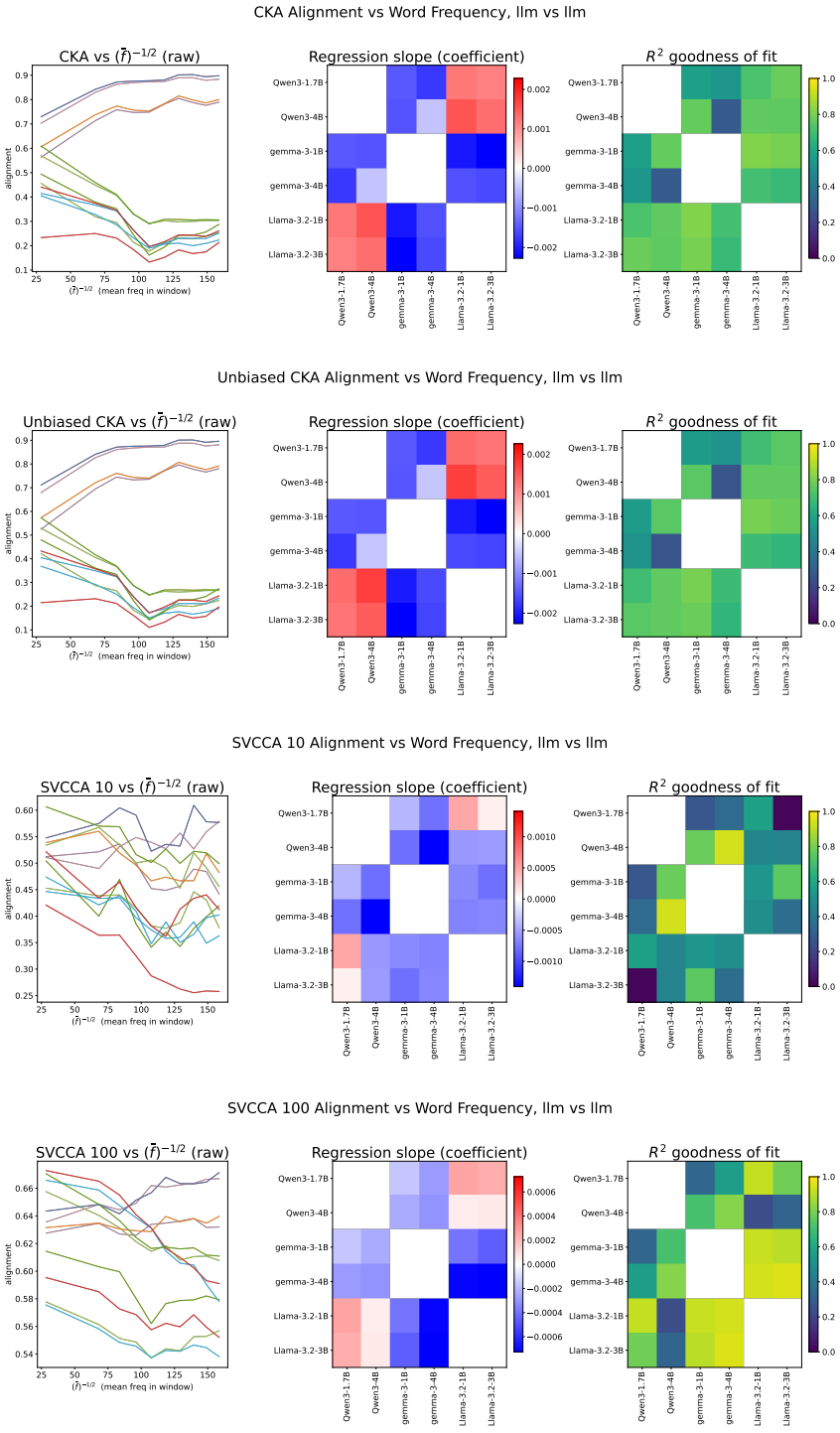


Figure 29: Same plot as in Figure 25 but for LLM vs LLM models.

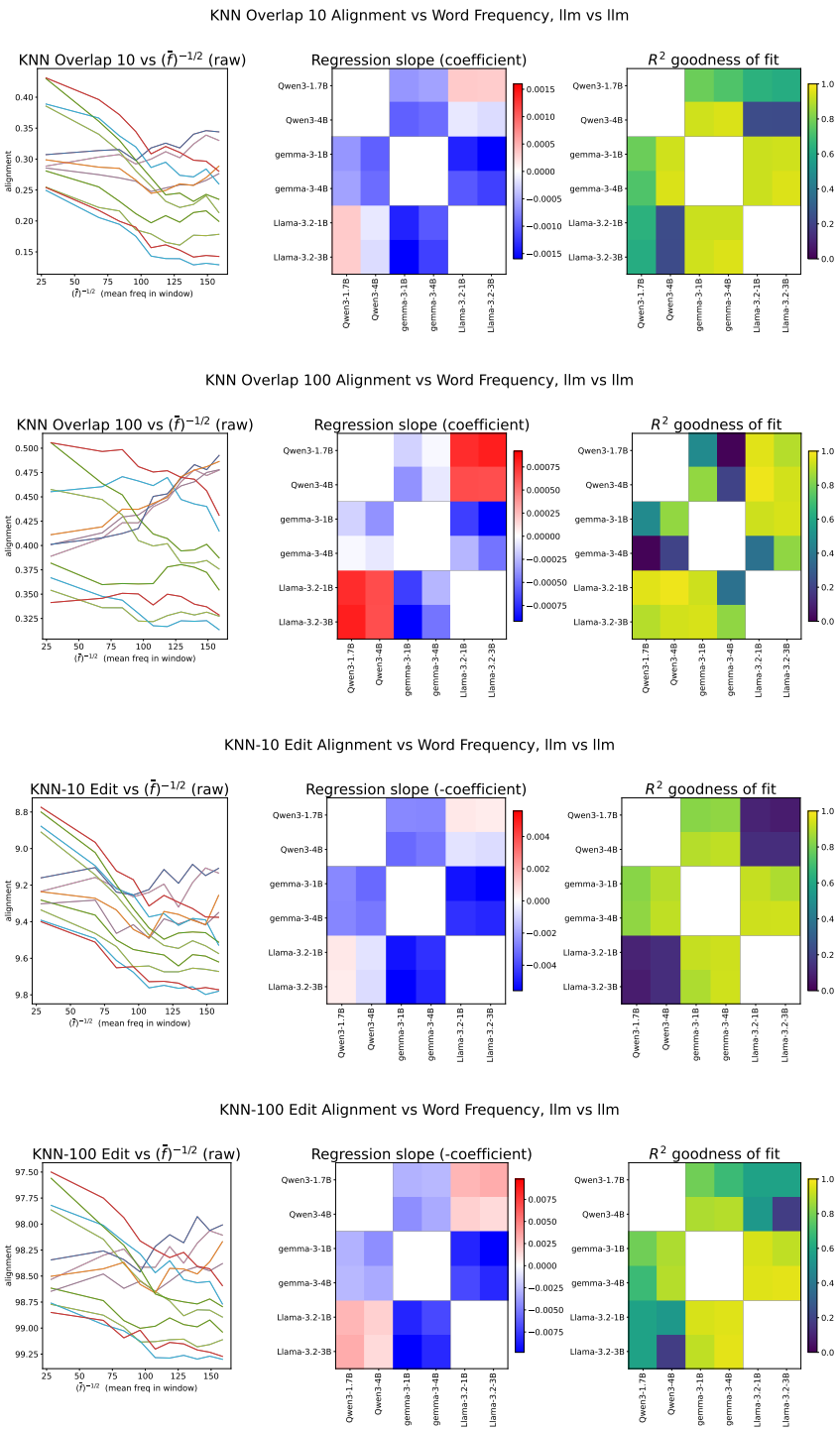


Figure 30: Same plot as in Figure 26 but for LLM vs LLM models.

The trends are somewhat weaker for LLM vs LLM and text embedding vs text embedding models, even though still strong with respect to the KNN-10 Overlap metric (as well as the two KNN neighborhood distance metrics). Again, we notice that typically the positive correlation between data frequency and alignment is stronger for larger models within the same family. This is consistent with our (Statistical Model of Representations) since one could expect larger models to have a smaller bias due to their greater expressivity, so alignment trends are more dependent on noise.

C.4.4 Ablating The Number of Tokens

We repeat the same experiments but restricting only to words that are composed of a single token in each model, thus controlling for the number of tokens. The results are consistent. We only present them for the KNN-10 metric for compactness.

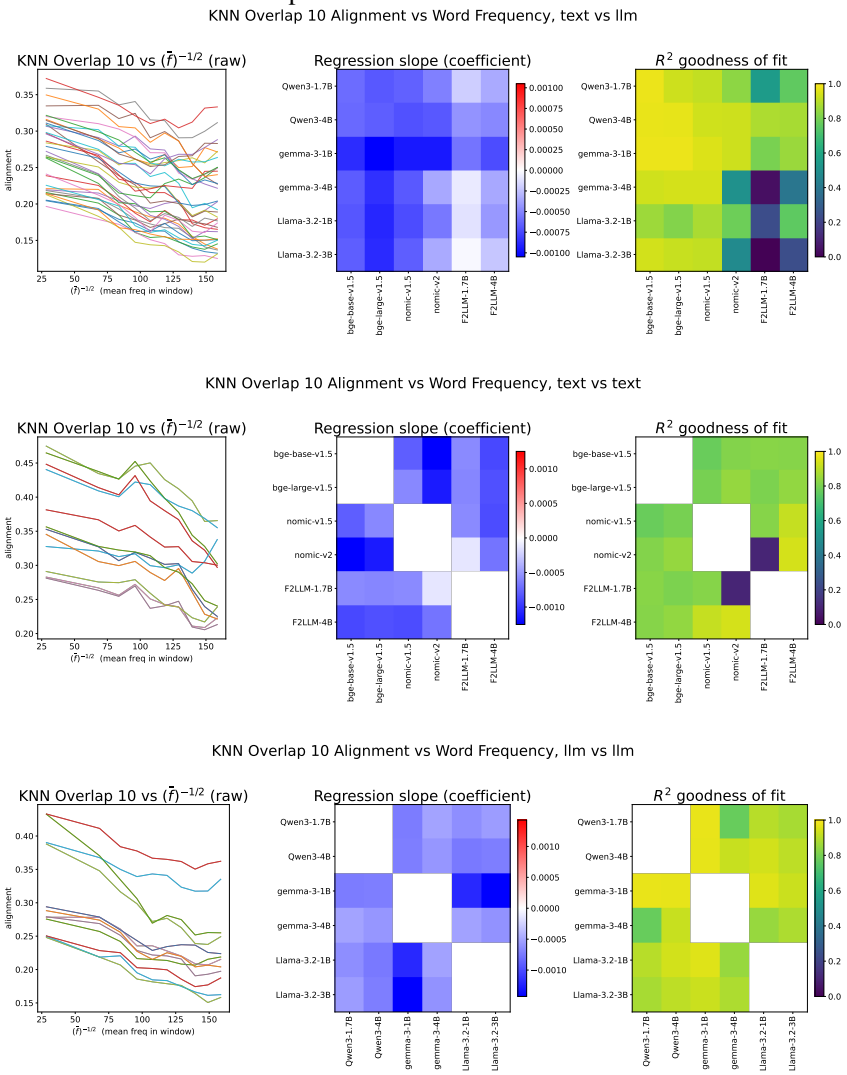


Figure 31: Same plot as in Figure 26 but only for single-token words.

C.4.5 Ablating The Word Type

We repeat the same experiments but restricting only to words that are the same part of speech, thus controlling for the type of word. The results are consistent. We only present them for the KNN-10 metric and nouns for compactness.

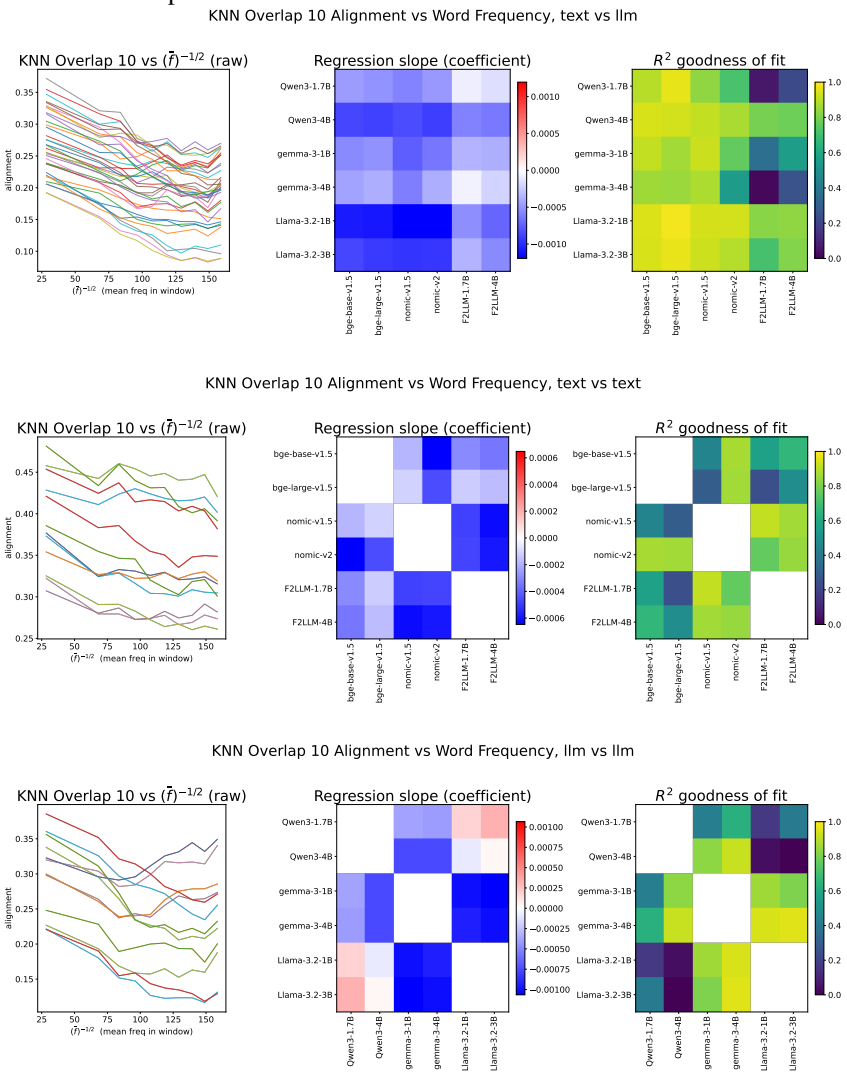


Figure 32: Same plot as in Figure 26 but only for nouns.

C.5 Decomposing the Alignment

C.5.1 Exact Definition of Features

1. min params: Suppose that the two models i, j have respectively N_i and N_j parameters. Then, first recorded is $\mathsf{minparams}'_{ij} = \min(\log N_i, \log N_j)$. Once computed for all models, minparams is formed by centering and normalizing to unit norm the vector $(\mathsf{minparams}'_{ij})_{i \neq j}$.
2. max params: Same as above but $\max$ instead of $\min$.
3. min depth: Suppose that the two models i, j have respectively depths L_i and L_j. Then, first recorded is $\mathsf{mindepth}'_{ij} = \min(L_i, L_j)$. Once computed for all models, mindepth is formed by centering and normalizing to unit norm the vector $(\mathsf{mindepth}'_{ij})_{i \neq j}$.
4. max depth: Same as above but $\max$ instead of $\min$.
5. min dimension: Suppose that the two models i, j have respectively dimension d_i and d_j. The dimension is the dimension of the representation, i.e. the output for multimodal and text embedding models. For image models, it is the dimension of the CLS token. In LLMs, it is the width of the respective layer. Then, first recorded is $\mathsf{mindimension}'_{ij} = \min(\log d_i, \log d_j)$. Once computed for all models, mindimension is formed by centering and normalizing to unit norm the vector $(\mathsf{mindimension}'_{ij})_{i \neq j}$.
6. max dimension: Same as above but $\max$ instead of $\min$.
7. min training images: Suppose that the two models i, j have been trained respectively on K_i and K_j images. Then, first recorded is $\mathsf{minimages}'_{ij} = \min(\log K_i, \log K_j)$. Once computed for all models, minimages is formed by centering and normalizing to unit norm the vector $(\mathsf{minimages}'_{ij})_{i \neq j}$.
8. max training images: Same as above but $\max$ instead of $\min$.
9. min training text tokens: Suppose that the two models i, j have been trained respectively on T_i and T_j text tokens. Then, first recorded is $\mathsf{mintokens}'_{ij} = \min(\log T_i, \log T_j)$. Once computed for all models, mintokens is formed by centering and normalizing to unit norm the vector $(\mathsf{mintokens}'_{ij})_{i \neq j}$.
10. max training text tokens: Same as above but $\max$ instead of $\min$.
11. min year: Suppose that the two models i, j have been released respectively in years Y_i and Y_j. Then, first recorded is $\mathsf{minyear}'_{ij} = \min(Y_i, Y_j)$. Once computed for all models, minyear is formed by centering and normalizing to unit norm the vector $(\mathsf{minyear}'_{ij})_{i \neq j}$.
12. max year: Same as above but $\max$ instead of $\min$.
13. text-text, text-img, img-img: Those variables one-hot encode the modality of the represented data. Again, they are centered and normalized.

C.5.2 Model Specifics

Table 1: Specifications of Used Models: Architecture

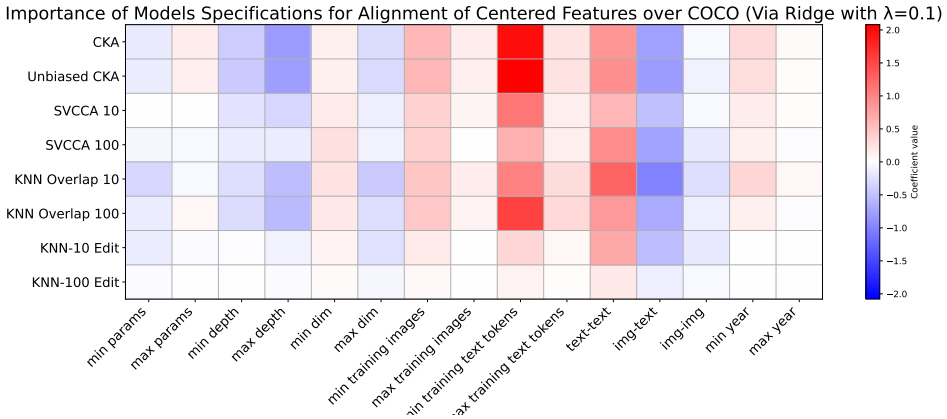
Model Identifier	Params (B)	Depth	Width
Llama-3.2-1B	1.0	16	2048
Llama-3.2-3B	3.0	28	3072
Qwen3-1.7B	1.7	28	2048
Qwen3-4B	4.0	36	2560
Gemma-3-1B	1.0	26	1152
Gemma-3-4B	4.0	32	2560
BGE-Base-en-v1.5	0.11	12	768
BGE-Large-en-v1.5	0.33	24	1024
F2LLM-1.7B	1.7	28	2048
F2LLM-4B	4.0	36	2560
Nomic-Embed-v1.5	0.14	12	768
Nomic-Embed-v2-MoE	0.47	12	768
SigLIP2-Base-T	0.086	12	768
SigLIP2-Large-T	0.43	24	1024
LAION-CLIP-B/32-T	0.15	12	512
LAION-CLIP-H/14-T	0.98	24	1024
OpenAI-CLIP-B/32-T	0.15	12	512
OpenAI-CLIP-L/14-T	0.43	12	768
SigLIP2-Base-I	0.086	12	768
SigLIP2-Large-I	0.43	24	1024
LAION-CLIP-B/32-I	0.15	12	768
LAION-CLIP-H/14-I	0.98	32	1280
OpenAI-CLIP-B/32-I	0.15	12	768
OpenAI-CLIP-L/14-I	0.43	24	1024
BEiT-Base	0.086	12	768
BEiT-Large	0.30	24	1024
DINOv2-Base	0.086	12	768
DINOv2-Large	0.30	24	1024
ViT-MAE-Huge	0.63	32	1280
ViT-MAE-Large	0.30	24	1024

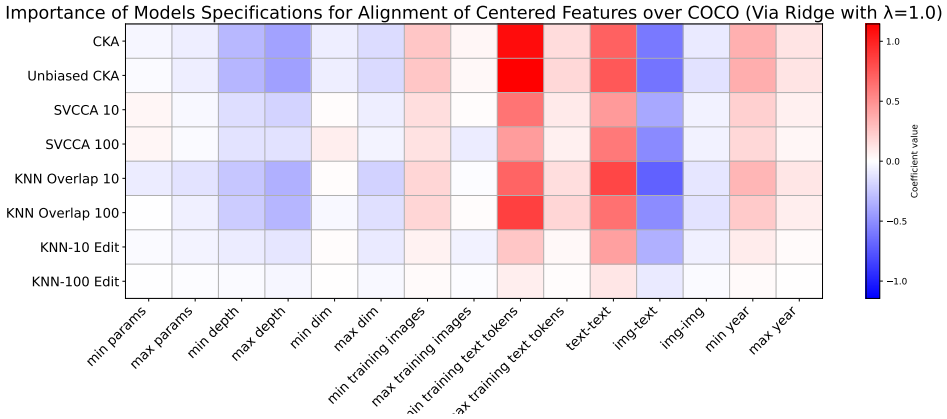
Table 2: Specifications of Used Models: Data and Modality; MM = Multimodal

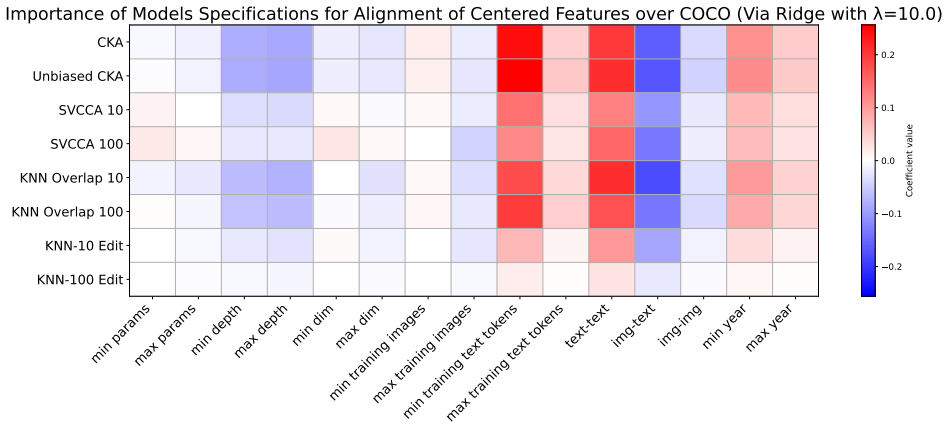
Model Identifier	Text Tok.	Img Tok.	Modality	Year
Llama-3.2-1B	9.00E+12	0	LLM	2024
Llama-3.2-3B	9.00E+12	0	LLM	2024
Qwen3-1.7B	3.60E+13	0	LLM	2025
Qwen3-4B	3.60E+13	0	LLM	2025
Gemma-3-1B	2.00E+12	0	LLM	2025
Gemma-3-4B	4.00E+12	4.00E+11	LLM	2025
BGE-Base-en-v1.5	4.00E+11	0	Text Emb.	2023
BGE-Large-en-v1.5	4.00E+11	0	Text Emb.	2023
F2LLM-1.7B	3.60E+13	0	Text Emb.	2025
F2LLM-4B	3.60E+13	0	Text Emb.	2025
Nomic-Embed-v1.5	6.00E+10	0	Text Emb.	2024
Nomic-Embed-v2-MoE	1.40E+12	0	Text Emb.	2025
SigLIP2-Base-T	2.16E+11	2.16E+11	MM, Text	2025
SigLIP2-Large-T	2.16E+11	2.16E+11	MM, Text	2025
LAION-CLIP-B/32-T	6.12E+11	6.12E+11	MM, Text	2022
LAION-CLIP-H/14-T	5.76E+11	5.76E+11	MM, Text	2023
OpenAI-CLIP-B/32-T	7.20E+09	7.20E+09	MM, Text	2021
OpenAI-CLIP-L/14-T	2.34E+11	2.34E+11	MM, Text	2021
SigLIP2-Base-I	2.16E+11	2.16E+11	MM, Image	2025
SigLIP2-Large-I	2.16E+11	2.16E+11	MM, Image	2025
LAION-CLIP-B/32-I	6.12E+11	6.12E+11	MM, Image	2022
LAION-CLIP-H/14-I	5.76E+11	5.76E+11	MM, Image	2023
OpenAI-CLIP-B/32-I	7.20E+09	7.20E+09	MM, Image	2021
OpenAI-CLIP-L/14-I	2.34E+11	2.34E+11	MM, Image	2021
BEiT-Base	0	2.20E+12	Image Found.	2021
BEiT-Large	0	2.20E+12	Image Found.	2021
DINOv2-Base	0	1.42E+08	Image Found.	2023
DINOv2-Large	0	1.42E+08	Image Found.	2023
ViT-MAE-Huge	0	5.26E+11	Image Found.	2021
ViT-MAE-Large	0	4.03E+11	Image Found.	2021

We repeat the same experiment from Figure 7 with varying values of the penalty λ and over three different datasets: COCO, CC3M, Visual Genome. The relative importance of features for alignment is very consistent between penalty values and datasets.

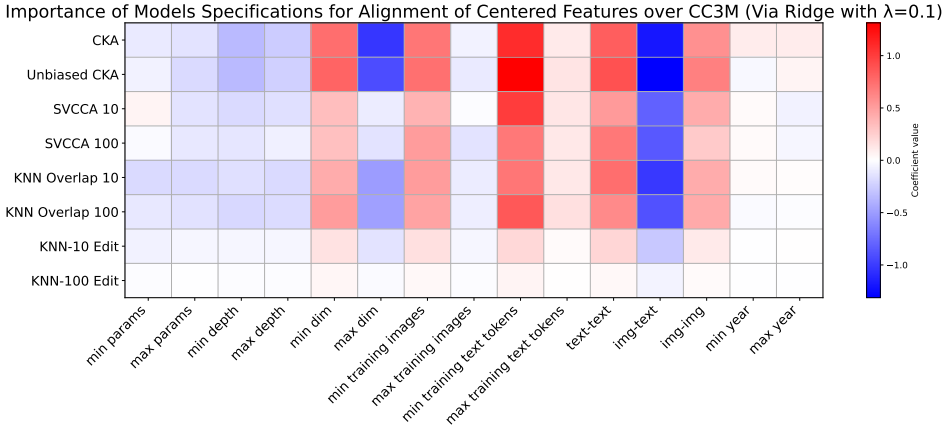
C.5.3 Experiments on COCO

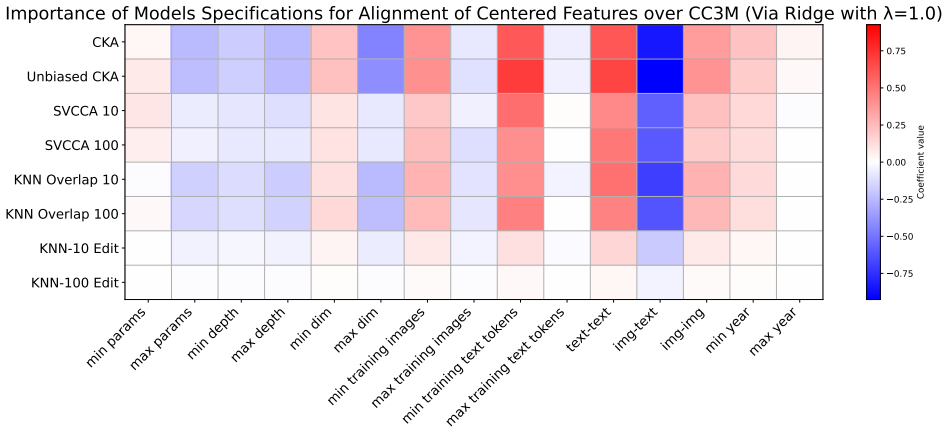


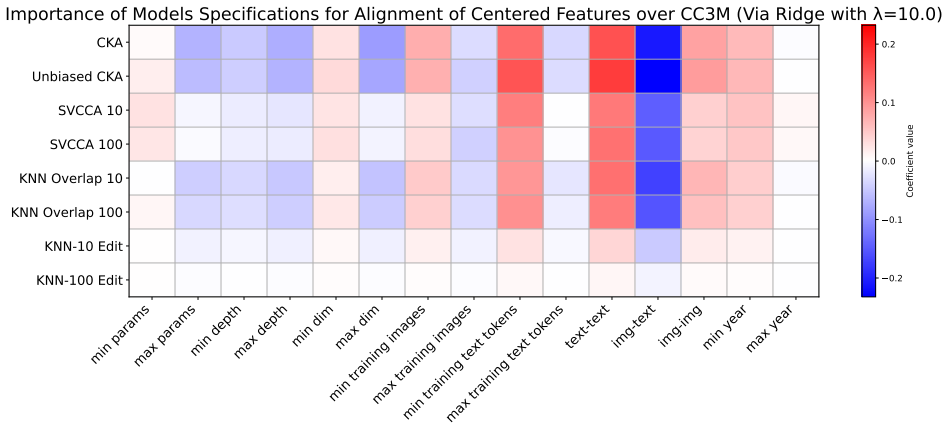




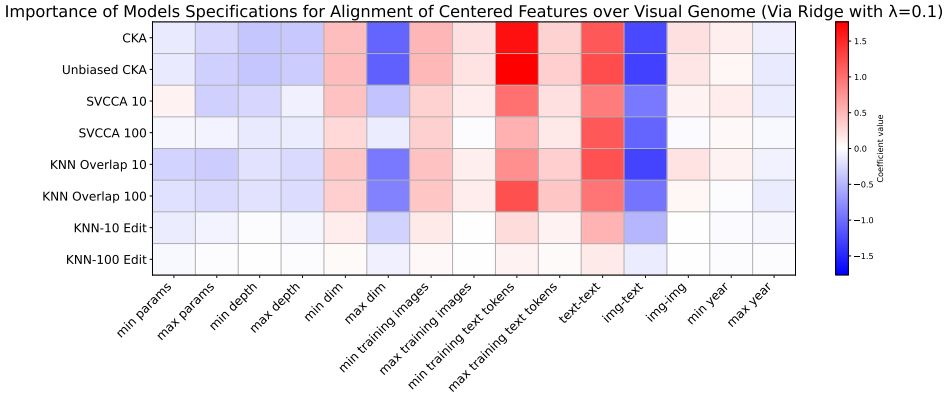
C.5.4 Experiments on CC3M

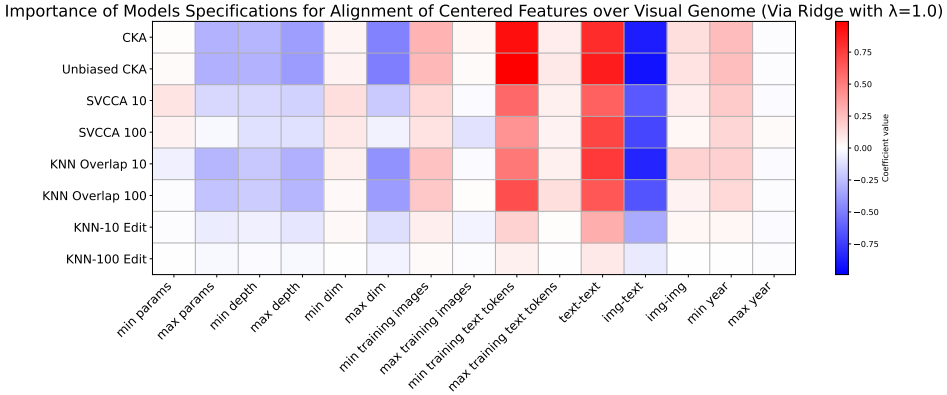


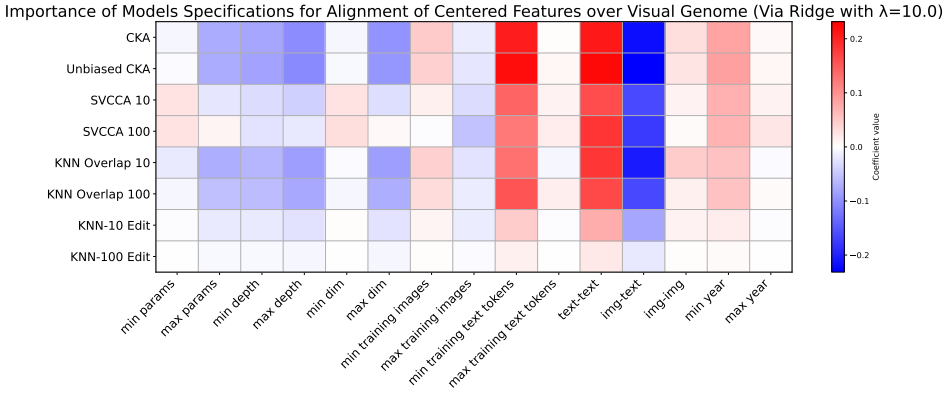




C.5.5 Experiments on Visual Genome







C.6 Dimension and Incoherence

C.6.1 Experiments on COCO

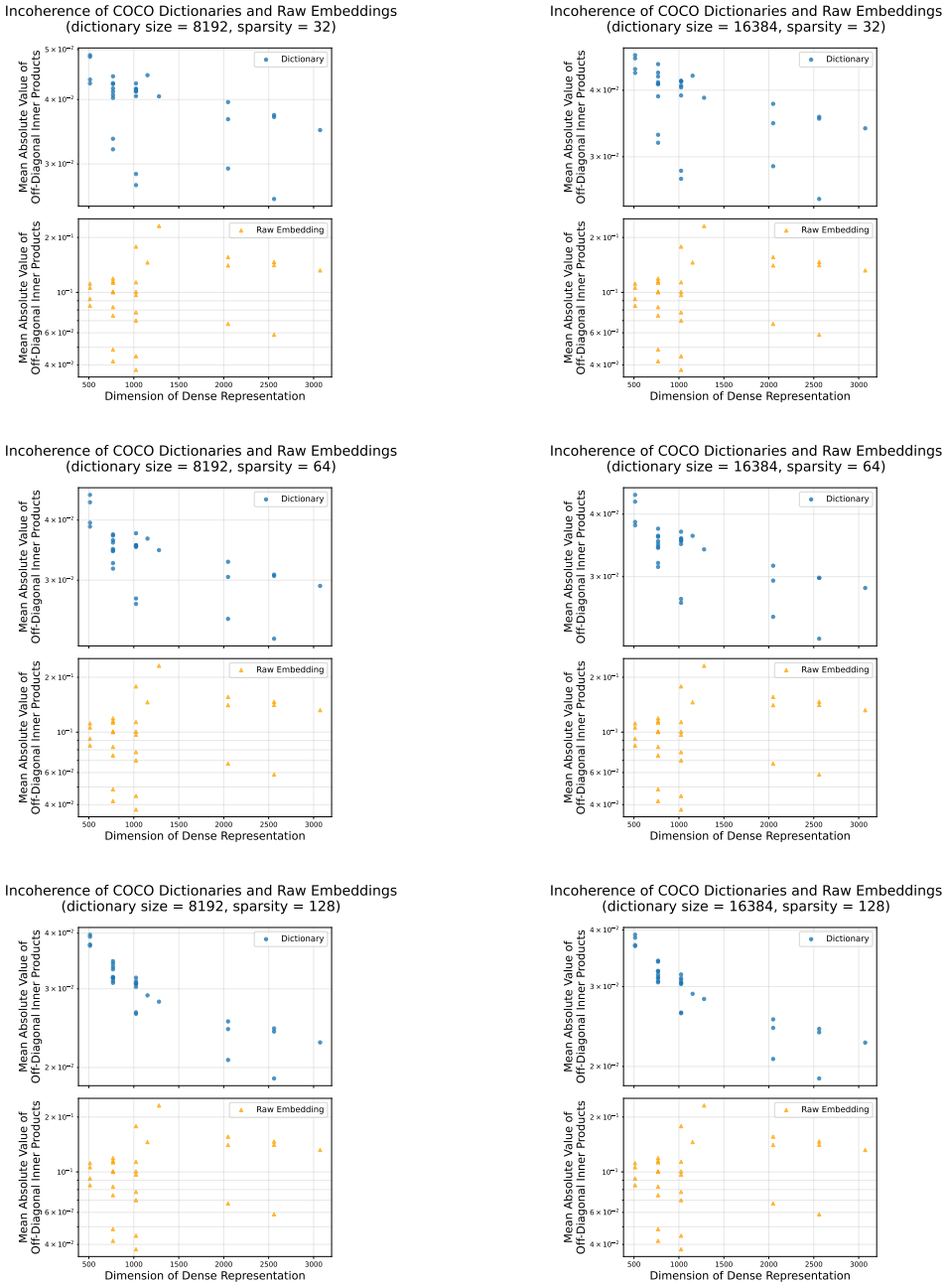


Figure 33: Plotted is the mean absolute value of inner products of distinct features from SAEs trained for each model in Table 1. The SAEs are trained as in Section B.1 with truncated features. We can see that the trend that higher dimensionality leads to smaller ϵ_{dict} is consistent across SAE parameters, although it is higher for higher sparsities.

C.6.2 Experiments on CC3M

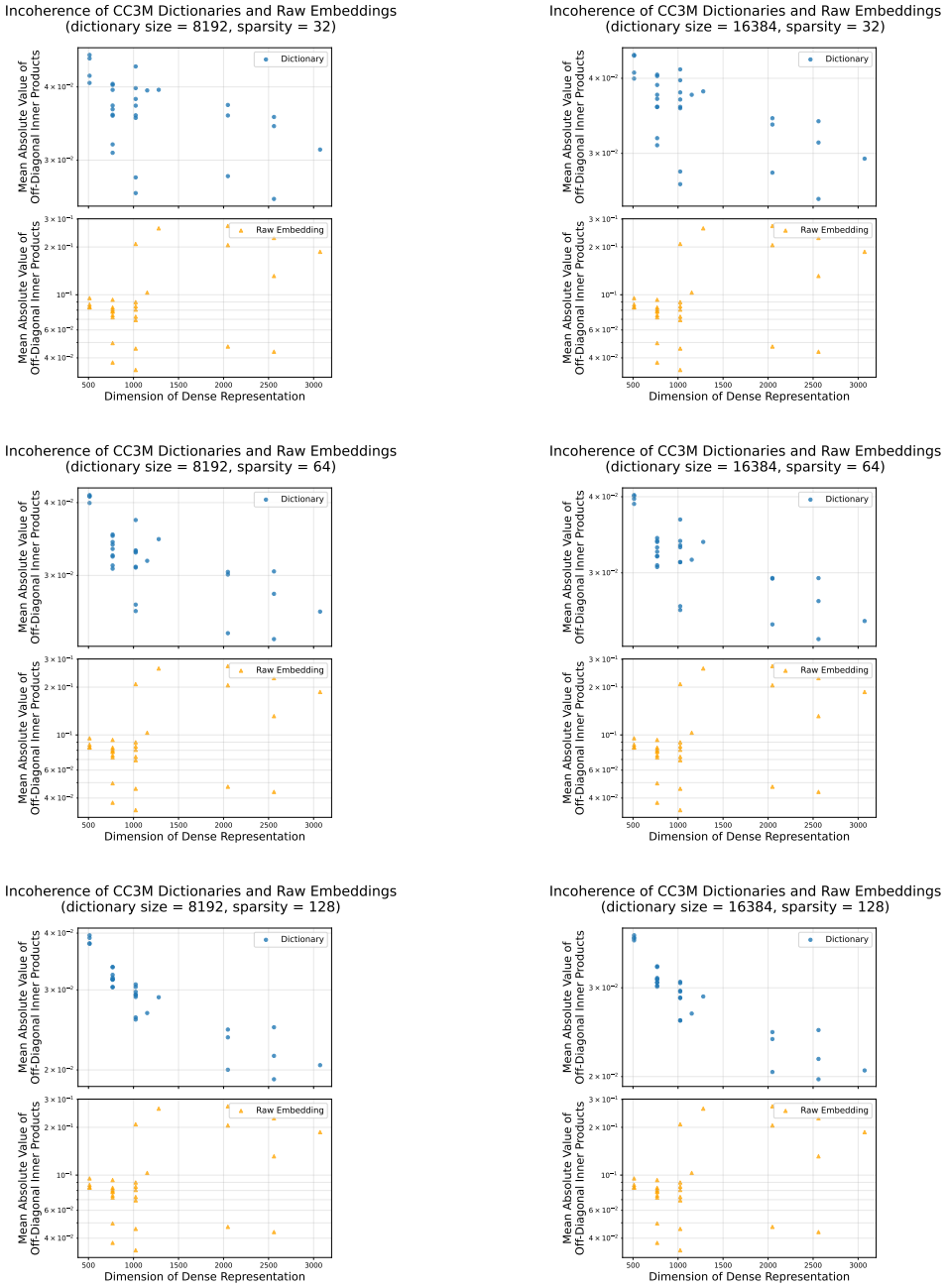


Figure 34: The same figure as Figure 33 but over the CC3M dataset. Qualitatively, all trends are similar.

C.6.3 Experiments on Visual Genome

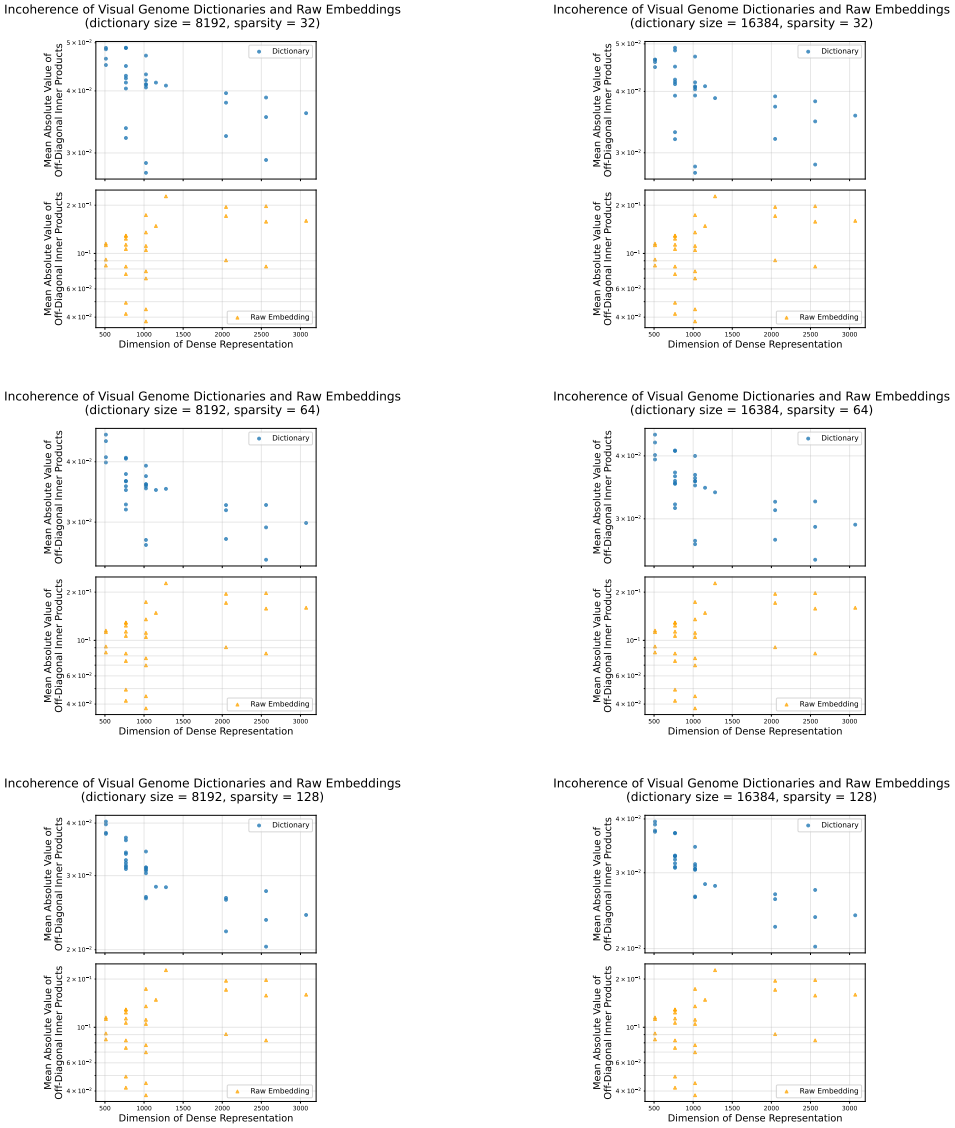


Figure 35: The same figure as Figure 33 but over the Visual Genome dataset. Qualitatively, all trends are similar.

C.7 Magnitudes of Sparse Features

Here, we plot statistics for the non-zero values of $\sqrt{k} \times M(X, \theta, f)$ for the sparse features we trained over COCO, CC3M, Visual Genome. We give the 5th and 95th percentiles of the values and their ratio. This shows that most entries are on the order of $\Theta(1/\sqrt{k})$ in support of Assumption A.3). In the case of COCO, we ablate the dimension by showing results for $d = 8192, 16384$. For the other datasets we only show $d = 16384$ for brevity.

C.7.1 Experiments on COCO

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	6501	0.0204259	0.334828	16.3923
BAAI__bge-large-en-v1.5_text	6703	0.0204214	0.331682	16.2419
codefuse-ai__F2LLM-1.7B_text	8128	0.025685	0.345818	13.4638
codefuse-ai__F2LLM-4B_text	8162	0.0270543	0.331568	12.2557
facebook__dinov2-base_img	8182	0.036548	0.254256	6.95678
facebook__dinov2-large_img	8173	0.0307982	0.269656	8.75558
facebook__vit-mae-huge_img	7064	0.0304735	0.153104	5.02419
facebook__vit-mae-large_img	5320	0.0371731	0.17026	4.58019
google__gemma-3-1b-it_text	7281	0.0326729	0.261633	8.00764
google__gemma-3-4b-it_text	7923	0.0311795	0.287769	9.22945
google__siglip2-base-patch16-256_img	7245	0.0403547	0.252237	6.25049
google__siglip2-base-patch16-256_text	6953	0.023171	0.348481	15.0395
google__siglip2-large-patch16-256_img	7104	0.0406788	0.259245	6.37297
google__siglip2-large-patch16-256_text	7395	0.0248135	0.346706	13.9724
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	7354	0.0433992	0.232559	5.3586
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	6894	0.019666	0.346289	17.6085
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	7784	0.0422671	0.250111	5.91739
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	6800	0.0213895	0.346613	16.2048
meta-llama__Llama-3.2-1B-Instruct_text	7557	0.0331361	0.283795	8.56452
meta-llama__Llama-3.2-3B-Instruct_text	7656	0.0315294	0.305695	9.69556
microsoft__beit-base-patch16-224_img	8188	0.038307	0.219244	5.72335
microsoft__beit-large-patch16-224_img	8185	0.0365769	0.22952	6.27498
nomic-ai__nomic-embed-text-v1.5_text	6088	0.017752	0.332005	18.7024
nomic-ai__nomic-embed-text-v2-moe_text	5997	0.0220674	0.379232	17.1852
openai__clip-vit-base-patch32_img	7507	0.0395703	0.226191	5.71618
openai__clip-vit-base-patch32_text	7259	0.0213545	0.336768	15.7704
openai__clip-vit-large-patch14_img	7984	0.0400789	0.242927	6.06122
openai__clip-vit-large-patch14_text	7305	0.0228015	0.332887	14.5994
Qwen__Qwen3-1.7B-Base_text	8029	0.0316816	0.263245	8.30907
Qwen__Qwen3-4B-Base_text	7962	0.0317245	0.274417	8.65001

Table 3: Statistics of nonzero sparse feature entries for COCO (truncated sparse features), pre-truncation sparse dimension $d = 8192$, and pre-truncation sparsity $k = 32$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	6862	0.0122533	0.229056	18.6934
BAAI__bge-large-en-v1.5_text	7009	0.0123748	0.223808	18.0858
codefuse-ai__F2LLM-1.7B_text	7735	0.0155105	0.24688	15.9169
codefuse-ai__F2LLM-4B_text	7858	0.0166596	0.2394	14.3701
facebook__dinov2-base_img	8179	0.0245707	0.187224	7.6198
facebook__dinov2-large_img	8179	0.0207208	0.197674	9.53989
facebook__vit-mae-huge_img	8019	0.0190106	0.103233	5.4303
facebook__vit-mae-large_img	7973	0.0231411	0.115512	4.99162
google__gemma-3-1b-it_text	7831	0.0196218	0.17284	8.80854
google__gemma-3-4b-it_text	8001	0.0189279	0.186128	9.83351
google__siglip2-base-patch16-256_img	7974	0.0257828	0.185305	7.18714
google__siglip2-base-patch16-256_text	7290	0.0143483	0.242921	16.9303
google__siglip2-large-patch16-256_img	7862	0.0256438	0.19303	7.52735
google__siglip2-large-patch16-256_text	7444	0.0156715	0.24251	15.4746
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	8084	0.0286987	0.170161	5.92922
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	7544	0.0122092	0.23524	19.2675
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	7987	0.0277527	0.180794	6.51445
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	7276	0.0132696	0.232559	17.5257
meta-llama__Llama-3.2-1B-Instruct_text	8023	0.0193464	0.189485	9.79434
meta-llama__Llama-3.2-3B-Instruct_text	8025	0.0181734	0.201008	11.0605
microsoft__beit-base-patch16-224_img	8183	0.0266547	0.158994	5.96496
microsoft__beit-large-patch16-224_img	8182	0.0260675	0.164443	6.30836
nomic-ai__nomic-embed-text-v1.5_text	6300	0.0108577	0.226129	20.8265
nomic-ai__nomic-embed-text-v2-moe_text	6709	0.0120219	0.263876	21.9496
openai__clip-vit-base-patch32_img	8080	0.0260672	0.173387	6.65154
openai__clip-vit-base-patch32_text	7675	0.0133075	0.218397	16.4116
openai__clip-vit-large-patch14_img	8075	0.0267408	0.179849	6.72564
openai__clip-vit-large-patch14_text	7459	0.0143854	0.226057	15.7143
Qwen__Qwen3-1.7B-Base_text	7981	0.0191093	0.179467	9.3916
Qwen__Qwen3-4B-Base_text	8043	0.0189182	0.185603	9.81081

Table 4: Statistics of nonzero sparse feature entries for COCO (truncated sparse features), pre-truncation sparse dimension $d = 8192$, and pre-truncation sparsity $k = 64$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	6576	0.00947677	0.159422	16.8224
BAAI__bge-large-en-v1.5_text	6692	0.00952511	0.160723	16.8736
codefuse-ai__F2LLM-1.7B_text	6711	0.0109136	0.191012	17.5021
codefuse-ai__F2LLM-4B_text	6878	0.0120174	0.190678	15.8667
facebook__dinov2-base_img	8145	0.0167647	0.133462	7.96089
facebook__dinov2-large_img	8159	0.0141138	0.13977	9.90304
facebook__vit-mae-huge_img	7508	0.0129893	0.0728865	5.61129
facebook__vit-mae-large_img	7477	0.0158122	0.0890399	5.63108
google__gemma-3-1b-it_text	7139	0.0131372	0.121057	9.2148
google__gemma-3-4b-it_text	7520	0.0125978	0.129427	10.2738
google__siglip2-base-patch16-256_img	7596	0.0183	0.147516	8.06099
google__siglip2-base-patch16-256_text	7091	0.0102159	0.175918	17.22
google__siglip2-large-patch16-256_img	7273	0.0178537	0.157	8.79372
google__siglip2-large-patch16-256_text	6836	0.0112409	0.184169	16.3838
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	7631	0.0212935	0.134371	6.31041
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	7226	0.00875272	0.157099	17.9486
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	7525	0.0199689	0.134361	6.72851
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	6789	0.00932585	0.159463	17.0991
meta-llama__Llama-3.2-1B-Instruct_text	7231	0.0123421	0.13002	10.5347
meta-llama__Llama-3.2-3B-Instruct_text	7458	0.0117533	0.143917	12.2448
microsoft__beit-base-patch16-224_img	8073	0.0186828	0.112873	6.04155
microsoft__beit-large-patch16-224_img	8123	0.0190029	0.114191	6.00911
nomic-ai__nomic-embed-text-v1.5_text	6132	0.0082975	0.158105	19.0545
nomic-ai__nomic-embed-text-v2-moe_text	6411	0.00847488	0.192248	22.6845
openai__clip-vit-base-patch32_img	7753	0.0184021	0.128132	6.9629
openai__clip-vit-base-patch32_text	7344	0.00973346	0.147666	15.171
openai__clip-vit-large-patch14_img	7800	0.0189537	0.130877	6.9051
openai__clip-vit-large-patch14_text	7212	0.0104516	0.157168	15.0377
Qwen__Qwen3-1.7B-Base_text	7394	0.0127484	0.124095	9.73415
Qwen__Qwen3-4B-Base_text	7447	0.0126204	0.13062	10.3499

Table 5: Statistics of nonzero sparse feature entries for COCO (truncated sparse features), pre-truncation sparse dimension $d = 8192$, and pre-truncation sparsity $k = 128$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	12156	0.0209062	0.334887	16.0185
BAAI__bge-large-en-v1.5_text	12634	0.0209635	0.32843	15.6667
codefuse-ai__F2LLM-1.7B_text	16232	0.0263172	0.342903	13.0296
codefuse-ai__F2LLM-4B_text	16336	0.027954	0.330471	11.822
facebook__dinov2-base_img	16316	0.0370923	0.248351	6.69549
facebook__dinov2-large_img	16302	0.0320621	0.263345	8.21361
facebook__vit-mae-huge_img	12167	0.0300413	0.151626	5.04725
facebook__vit-mae-large_img	8429	0.0367916	0.170366	4.63058
google__gemma-3-1b-it_text	13509	0.0329931	0.269656	8.17308
google__gemma-3-4b-it_text	15421	0.031239	0.280447	8.97747
google__siglip2-base-patch16-256_img	12600	0.0398165	0.248102	6.23113
google__siglip2-base-patch16-256_text	13421	0.0233936	0.343155	14.6688
google__siglip2-large-patch16-256_img	11990	0.0402536	0.254713	6.32772
google__siglip2-large-patch16-256_text	14299	0.0248937	0.340488	13.6777
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	13138	0.0432908	0.227514	5.25549
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	12445	0.0198198	0.340883	17.1991
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	13751	0.0421843	0.24899	5.90244
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	12834	0.0218466	0.345366	15.8087
meta-llama__Llama-3.2-1B-Instruct_text	13775	0.0336688	0.286113	8.49786
meta-llama__Llama-3.2-3B-Instruct_text	14243	0.0317138	0.296067	9.33558
microsoft__beit-base-patch16-224_img	16331	0.0389659	0.220325	5.65431
microsoft__beit-large-patch16-224_img	16348	0.0373687	0.227565	6.08971
nomic-ai__nomic-embed-text-v1.5_text	11309	0.018072	0.329122	18.2117
nomic-ai__nomic-embed-text-v2-moe_text	11172	0.0221846	0.358485	16.1592
openai__clip-vit-base-patch32_img	13599	0.0395022	0.228092	5.77415
openai__clip-vit-base-patch32_text	13370	0.021659	0.330874	15.2765
openai__clip-vit-large-patch14_img	14855	0.0399646	0.240326	6.01348
openai__clip-vit-large-patch14_text	14012	0.0229268	0.32795	14.3042
Qwen__Qwen3-1.7B-Base_text	15622	0.0318452	0.260091	8.16736
Qwen__Qwen3-4B-Base_text	15127	0.0322912	0.277167	8.58336

Table 6: Statistics of nonzero sparse feature entries for COCO (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 32$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	13425	0.0121456	0.221162	18.2093
BAAI__bge-large-en-v1.5_text	14080	0.0124886	0.215718	17.2732
codefuse-ai__F2LLM-1.7B_text	15641	0.015605	0.238566	15.2877
codefuse-ai__F2LLM-4B_text	15951	0.0169069	0.233267	13.7971
facebook__dinov2-base_img	16349	0.0251458	0.183733	7.30671
facebook__dinov2-large_img	16300	0.0218632	0.19413	8.87933
facebook__vit-mae-huge_img	16135	0.0186105	0.098865	5.31231
facebook__vit-mae-large_img	16064	0.0228066	0.113366	4.97077
google__gemma-3-1b-it_text	15833	0.0193333	0.172215	8.9077
google__gemma-3-4b-it_text	16197	0.0190315	0.187962	9.87633
google__siglip2-base-patch16-256_img	15904	0.0253582	0.180948	7.13568
google__siglip2-base-patch16-256_text	14425	0.0141629	0.238086	16.8106
google__siglip2-large-patch16-256_img	15625	0.0250484	0.18953	7.56654
google__siglip2-large-patch16-256_text	14830	0.0154696	0.234929	15.1864
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	16248	0.0284984	0.166281	5.83476
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	14551	0.0121581	0.225188	18.5216
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	15906	0.0273307	0.175952	6.43791
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	14355	0.013256	0.231981	17.5001
meta-llama__Llama-3.2-1B-Instruct_text	16062	0.0195725	0.191495	9.78384
meta-llama__Llama-3.2-3B-Instruct_text	16196	0.0182568	0.196727	10.7755
microsoft__beit-base-patch16-224_img	16364	0.0271664	0.155521	5.72477
microsoft__beit-large-patch16-224_img	16340	0.0267062	0.159824	5.98453
nomic-ai__nomic-embed-text-v1.5_text	11943	0.0108978	0.223007	20.4635
nomic-ai__nomic-embed-text-v2-moe_text	13154	0.0118922	0.256222	21.5455
openai__clip-vit-base-patch32_img	16206	0.0256195	0.168311	6.56963
openai__clip-vit-base-patch32_text	14997	0.0135262	0.218082	16.1229
openai__clip-vit-large-patch14_img	16190	0.0265251	0.176135	6.64031
openai__clip-vit-large-patch14_text	14616	0.0144087	0.225442	15.6462
Qwen__Qwen3-1.7B-Base_text	16147	0.0191057	0.176288	9.22695
Qwen__Qwen3-4B-Base_text	16203	0.0192802	0.187183	9.70855

Table 7: Statistics of nonzero sparse feature entries for COCO (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 64$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	12953	0.00921178	0.148766	16.1495
BAAI__bge-large-en-v1.5_text	13322	0.00922429	0.150628	16.3295
codefuse-ai__F2LLM-1.7B_text	13374	0.0107951	0.181575	16.8202
codefuse-ai__F2LLM-4B_text	13935	0.0117504	0.177851	15.1358
facebook__dinov2-base_img	16335	0.0168717	0.131751	7.80896
facebook__dinov2-large_img	16330	0.0144381	0.138797	9.61319
facebook__vit-mae-huge_img	15186	0.0127273	0.0686851	5.39665
facebook__vit-mae-large_img	15179	0.0152865	0.0845693	5.53228
google__gemma-3-1b-it_text	14454	0.012964	0.113653	8.76685
google__gemma-3-4b-it_text	15251	0.0123482	0.123099	9.96902
google__siglip2-base-patch16-256_img	15346	0.0176359	0.140577	7.9711
google__siglip2-base-patch16-256_text	13790	0.00994756	0.172252	17.316
google__siglip2-large-patch16-256_img	14669	0.0170101	0.147325	8.66105
google__siglip2-large-patch16-256_text	13731	0.0108356	0.174262	16.0824
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	15571	0.0206177	0.130748	6.34155
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	13924	0.00858519	0.150882	17.5747
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	15182	0.0190137	0.127393	6.70005
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	13568	0.00908236	0.156768	17.2608
meta-llama__Llama-3.2-1B-Instruct_text	14908	0.0122443	0.125104	10.2173
meta-llama__Llama-3.2-3B-Instruct_text	15092	0.0115528	0.135239	11.7062
microsoft__beit-base-patch16-224_img	16247	0.0187374	0.109942	5.86751
microsoft__beit-large-patch16-224_img	16305	0.0190576	0.114827	6.02529
nomic-ai__nomic-embed-text-v1.5_text	11706	0.00806524	0.1498	18.5736
nomic-ai__nomic-embed-text-v2-moe_text	12364	0.00816995	0.179353	21.9528
openai__clip-vit-base-patch32_img	15716	0.0176926	0.127141	7.18613
openai__clip-vit-base-patch32_text	14336	0.00962948	0.1523	15.816
openai__clip-vit-large-patch14_img	15775	0.0183329	0.132362	7.2199
openai__clip-vit-large-patch14_text	14018	0.0102086	0.154717	15.1555
Qwen__Qwen3-1.7B-Base_text	15177	0.0125856	0.118007	9.37631
Qwen__Qwen3-4B-Base_text	15230	0.0125451	0.125952	10.04

Table 8: Statistics of nonzero sparse feature entries for COCO (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 128$.

C.7.2 Experiments on CC3M

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	13920	0.0254295	0.311401	12.2457
BAAI__bge-large-en-v1.5_text	13831	0.026407	0.303213	11.4823
codefuse-ai__F2LLM-1.7B_text	16310	0.0306691	0.284788	9.28584
codefuse-ai__F2LLM-4B_text	16345	0.0327587	0.269746	8.23433
facebook__dinov2-base_img	16214	0.0380363	0.247877	6.51686
facebook__dinov2-large_img	16219	0.0330927	0.262779	7.9407
facebook__vit-mae-huge_img	13601	0.0263153	0.130909	4.97464
facebook__vit-mae-large_img	10839	0.0327	0.148462	4.54012
google__gemma-3-1b-it_text	14404	0.0345314	0.217325	6.29354
google__gemma-3-4b-it_text	15952	0.0323284	0.203945	6.30854
google__siglip2-base-patch16-256_img	13684	0.0396712	0.228396	5.75724
google__siglip2-base-patch16-256_text	14513	0.0292539	0.282951	9.67224
google__siglip2-large-patch16-256_img	13259	0.0411318	0.231333	5.62419
google__siglip2-large-patch16-256_text	14814	0.0329464	0.277085	8.41017
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	14661	0.0408718	0.213936	5.2343
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	15403	0.0270586	0.285249	10.5419
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	15240	0.0415339	0.221113	5.32368
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	14969	0.0305111	0.284787	9.33387
meta-llama__Llama-3.2-1B-Instruct_text	15554	0.0250882	0.207663	8.27732
meta-llama__Llama-3.2-3B-Instruct_text	15899	0.0252918	0.214321	8.47392
microsoft__beit-base-patch16-224_img	16234	0.0411063	0.208686	5.07673
microsoft__beit-large-patch16-224_img	16283	0.039738	0.210243	5.29074
nomic-ai__nomic-embed-text-v1.5_text	13240	0.0225723	0.320595	14.203
nomic-ai__nomic-embed-text-v2-moe_text	13178	0.0237316	0.337105	14.2049
openai__clip-vit-base-patch32_img	14149	0.0408863	0.214074	5.23584
openai__clip-vit-base-patch32_text	15307	0.029697	0.272682	9.18212
openai__clip-vit-large-patch14_img	15072	0.0408191	0.215579	5.28132
openai__clip-vit-large-patch14_text	15134	0.0312728	0.277359	8.86902
Qwen__Qwen3-1.7B-Base_text	15721	0.0222427	0.185882	8.35699
Qwen__Qwen3-4B-Base_text	15813	0.0232599	0.201925	8.68125

Table 9: Statistics of nonzero sparse feature entries for CC3M (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 32$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	14852	0.016976	0.232482	13.6947
BAAI__bge-large-en-v1.5_text	14482	0.0172617	0.219226	12.7002
codefuse-ai__F2LLM-1.7B_text	15930	0.0204981	0.215745	10.5251
codefuse-ai__F2LLM-4B_text	16269	0.0218171	0.197833	9.06778
facebook__dinov2-base_img	16348	0.0257843	0.180466	6.99908
facebook__dinov2-large_img	16262	0.0230792	0.191679	8.30527
facebook__vit-mae-huge_img	16196	0.0165786	0.0880321	5.31
facebook__vit-mae-large_img	16092	0.020561	0.0975483	4.74433
google__gemma-3-1b-it_text	15631	0.0222217	0.150841	6.78801
google__gemma-3-4b-it_text	16164	0.0197148	0.144143	7.31142
google__siglip2-base-patch16-256_img	15612	0.0249098	0.166845	6.69794
google__siglip2-base-patch16-256_text	15529	0.019116	0.20296	10.6173
google__siglip2-large-patch16-256_img	15459	0.0254063	0.169271	6.66256
google__siglip2-large-patch16-256_text	15214	0.0217111	0.200358	9.2284
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	16172	0.0267453	0.156716	5.85957
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	16190	0.0177759	0.198928	11.1909
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	15818	0.02685	0.159632	5.94533
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	15349	0.0199849	0.199043	9.95967
meta-llama__Llama-3.2-1B-Instruct_text	16136	0.0151176	0.143841	9.51482
meta-llama__Llama-3.2-3B-Instruct_text	15936	0.0152475	0.151104	9.91012
microsoft__beit-base-patch16-224_img	16325	0.0285476	0.151634	5.3116
microsoft__beit-large-patch16-224_img	16217	0.0282752	0.150382	5.31852
nomic-ai__nomic-embed-text-v1.5_text	14371	0.0146403	0.231188	15.7913
nomic-ai__nomic-embed-text-v2-moe_text	14431	0.0148379	0.258115	17.3956
openai__clip-vit-base-patch32_img	16153	0.026202	0.155622	5.93932
openai__clip-vit-base-patch32_text	16222	0.019699	0.189125	9.60072
openai__clip-vit-large-patch14_img	16016	0.0271577	0.157692	5.80654
openai__clip-vit-large-patch14_text	15835	0.0208797	0.188855	9.04492
Qwen__Qwen3-1.7B-Base_text	16243	0.0135206	0.1262	9.33389
Qwen__Qwen3-4B-Base_text	16131	0.014687	0.131423	8.94826

Table 10: Statistics of nonzero sparse feature entries for CC3M (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 64$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	14303	0.0146377	0.182631	12.4767
BAAI__bge-large-en-v1.5_text	14068	0.0139354	0.169947	12.1953
codefuse-ai__F2LLM-1.7B_text	14125	0.0153332	0.169556	11.0581
codefuse-ai__F2LLM-4B_text	15147	0.0159894	0.140653	8.79667
facebook__dinov2-base_img	16344	0.0176766	0.123655	6.9954
facebook__dinov2-large_img	16311	0.0165585	0.128942	7.78708
facebook__vit-mae-huge_img	15552	0.0111773	0.055826	4.9946
facebook__vit-mae-large_img	15165	0.0139581	0.0644786	4.61943
google__gemma-3-1b-it_text	15014	0.0162736	0.1093	6.71638
google__gemma-3-4b-it_text	15163	0.0137327	0.101828	7.415
google__siglip2-base-patch16-256_img	14792	0.0179614	0.13477	7.50329
google__siglip2-base-patch16-256_text	15329	0.0146317	0.1537	10.5046
google__siglip2-large-patch16-256_img	14066	0.0174327	0.128601	7.37699
google__siglip2-large-patch16-256_text	14486	0.0164128	0.143507	8.7436
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	14939	0.0211209	0.131191	6.21146
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	15804	0.0147165	0.152747	10.3793
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	14682	0.0197453	0.115771	5.86322
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	14663	0.0157474	0.137498	8.73149
meta-llama__Llama-3.2-1B-Instruct_text	14912	0.0110526	0.101962	9.22514
meta-llama__Llama-3.2-3B-Instruct_text	15181	0.0107867	0.104912	9.72611
microsoft__beit-base-patch16-224_img	16255	0.0200309	0.105119	5.24783
microsoft__beit-large-patch16-224_img	16211	0.0207458	0.105733	5.09659
nomic-ai__nomic-embed-text-v1.5_text	13653	0.0129699	0.181755	14.0137
nomic-ai__nomic-embed-text-v2-moe_text	14216	0.0117553	0.206009	17.5248
openai__clip-vit-base-patch32_img	15319	0.0193033	0.120389	6.23671
openai__clip-vit-base-patch32_text	15813	0.016171	0.144277	8.92197
openai__clip-vit-large-patch14_img	15455	0.0201092	0.118739	5.90469
openai__clip-vit-large-patch14_text	15490	0.0168784	0.14567	8.63054
Qwen__Qwen3-1.7B-Base_text	15321	0.010005	0.0818743	8.18334
Qwen__Qwen3-4B-Base_text	15425	0.0108446	0.0882932	8.14165

Table 11: Statistics of nonzero sparse feature entries for CC3M (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 128$.

C.7.3 Experiments on Visual Genome

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	9587	0.0258465	0.294767	11.4045
BAAI__bge-large-en-v1.5_text	9458	0.0251976	0.294414	11.6842
codefuse-ai__F2LLM-1.7B_text	12253	0.0314145	0.292297	9.30452
codefuse-ai__F2LLM-4B_text	13250	0.0327303	0.28757	8.78605
facebook__dinov2-base_img	16312	0.0369236	0.252441	6.83685
facebook__dinov2-large_img	16295	0.0318728	0.267687	8.39862
facebook__vit-mae-huge_img	11802	0.0303962	0.155136	5.10379
facebook__vit-mae-large_img	8466	0.0371367	0.173793	4.67981
google__gemma-3-1b-it_text	9008	0.0333885	0.233623	6.99711
google__gemma-3-4b-it_text	11425	0.031951	0.231986	7.26069
google__siglip2-base-patch16-256_img	12248	0.0399248	0.253171	6.34119
google__siglip2-base-patch16-256_text	9888	0.0289573	0.293752	10.1443
google__siglip2-large-patch16-256_img	11782	0.0402489	0.257826	6.40578
google__siglip2-large-patch16-256_text	10414	0.0307455	0.282401	9.1851
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	12854	0.0428923	0.231121	5.38841
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	11322	0.0269072	0.307126	11.4143
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	13682	0.041509	0.249676	6.015
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	10699	0.029232	0.293988	10.0571
meta-llama__Llama-3.2-1B-Instruct_text	8295	0.0305928	0.225879	7.38339
meta-llama__Llama-3.2-3B-Instruct_text	8918	0.0314317	0.239794	7.62903
microsoft__beit-base-patch16-224_img	16309	0.0388014	0.225075	5.80069
microsoft__beit-large-patch16-224_img	16358	0.0371547	0.236853	6.37478
nomic-ai__nomic-embed-text-v1.5_text	9404	0.0242776	0.304992	12.5627
nomic-ai__nomic-embed-text-v2-moe_text	8001	0.0289007	0.298165	10.3169
openai__clip-vit-base-patch32_img	13494	0.0391038	0.228169	5.83494
openai__clip-vit-base-patch32_text	11921	0.0275645	0.294967	10.701
openai__clip-vit-large-patch14_img	14821	0.039517	0.241423	6.10934
openai__clip-vit-large-patch14_text	11422	0.0282427	0.290792	10.2962
Qwen__Qwen3-1.7B-Base_text	10365	0.028953	0.221531	7.6514
Qwen__Qwen3-4B-Base_text	9710	0.0290642	0.205751	7.0792

Table 12: Statistics of nonzero sparse feature entries for Visual Genome (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 32$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	13218	0.015027	0.193868	12.9013
BAAI__bge-large-en-v1.5_text	13397	0.0144733	0.187733	12.971
codefuse-ai__F2LLM-1.7B_text	15704	0.0181161	0.202969	11.2038
codefuse-ai__F2LLM-4B_text	15987	0.0188635	0.198664	10.5317
facebook__dinov2-base_img	16364	0.025118	0.185073	7.36813
facebook__dinov2-large_img	16331	0.0217118	0.194673	8.96623
facebook__vit-mae-huge_img	16115	0.0189728	0.104986	5.53349
facebook__vit-mae-large_img	16029	0.0228774	0.118996	5.20144
google__gemma-3-1b-it_text	15698	0.0197309	0.143592	7.27752
google__gemma-3-4b-it_text	15873	0.0187409	0.145958	7.78824
google__siglip2-base-patch16-256_img	15983	0.0253152	0.183988	7.26788
google__siglip2-base-patch16-256_text	14472	0.0164526	0.203958	12.3967
google__siglip2-large-patch16-256_img	15693	0.0249241	0.187682	7.53016
google__siglip2-large-patch16-256_text	15001	0.0180389	0.19326	10.7135
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	16275	0.0282605	0.169612	6.00173
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	15058	0.0156479	0.197197	12.6022
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	16039	0.0269098	0.179611	6.67456
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	14669	0.0167427	0.203237	12.1389
meta-llama__Llama-3.2-1B-Instruct_text	15515	0.0177032	0.143266	8.09264
meta-llama__Llama-3.2-3B-Instruct_text	15509	0.0177851	0.157042	8.82998
microsoft__beit-base-patch16-224_img	16365	0.0272404	0.156832	5.75734
microsoft__beit-large-patch16-224_img	16355	0.0267243	0.162864	6.0942
nomic-ai__nomic-embed-text-v1.5_text	12464	0.0137996	0.202276	14.6581
nomic-ai__nomic-embed-text-v2-moe_text	13234	0.0148908	0.202929	13.6278
openai__clip-vit-base-patch32_img	16216	0.0255316	0.1704	6.67408
openai__clip-vit-base-patch32_text	15272	0.0163975	0.198987	12.1353
openai__clip-vit-large-patch14_img	16206	0.0260699	0.176949	6.78749
openai__clip-vit-large-patch14_text	14799	0.0168074	0.195592	11.6372
Qwen__Qwen3-1.7B-Base_text	15791	0.0172485	0.13732	7.96129
Qwen__Qwen3-4B-Base_text	15711	0.0171786	0.129968	7.5657

Table 13: Statistics of nonzero sparse feature entries for Visual Genome (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 64$.

Model	d_{trunc}	5%	95%	95%/5%
BAAI__bge-base-en-v1.5_text	13282	0.0107109	0.123221	11.5042
BAAI__bge-large-en-v1.5_text	13727	0.0100894	0.126666	12.5543
codefuse-ai__F2LLM-1.7B_text	14317	0.0116631	0.143572	12.3099
codefuse-ai__F2LLM-4B_text	14049	0.0122185	0.140684	11.514
facebook__dinov2-base_img	16341	0.0167531	0.130932	7.8154
facebook__dinov2-large_img	16346	0.0143212	0.137849	9.62553
facebook__vit-mae-huge_img	15237	0.012855	0.0738263	5.74299
facebook__vit-mae-large_img	15270	0.0155369	0.0901772	5.80407
google__gemma-3-1b-it_text	14503	0.0124713	0.090303	7.24089
google__gemma-3-4b-it_text	15135	0.0118836	0.098183	8.26206
google__siglip2-base-patch16-256_img	15435	0.0175791	0.140541	7.99478
google__siglip2-base-patch16-256_text	13879	0.0110849	0.142973	12.898
google__siglip2-large-patch16-256_img	14796	0.0168805	0.150169	8.896
google__siglip2-large-patch16-256_text	14343	0.012156	0.134844	11.0928
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_img	15658	0.0203661	0.133905	6.57488
laion__CLIP-ViT-B-32-laion2B-s34B-b79K_text	14446	0.0109357	0.125573	11.4829
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_img	15349	0.0188075	0.129284	6.87407
laion__CLIP-ViT-H-14-laion2B-s32B-b79K_text	13865	0.0113247	0.130727	11.5435
meta-llama__Llama-3.2-1B-Instruct_text	14982	0.0110692	0.0934486	8.44225
meta-llama__Llama-3.2-3B-Instruct_text	15068	0.0108947	0.105156	9.65201
microsoft__beit-base-patch16-224_img	16287	0.0188108	0.112288	5.96935
microsoft__beit-large-patch16-224_img	16325	0.0191548	0.114931	6.00014
nomic-ai__nomic-embed-text-v1.5_text	13087	0.00968237	0.128162	13.2367
nomic-ai__nomic-embed-text-v2-moe_text	13168	0.00933232	0.133336	14.2876
openai__clip-vit-base-patch32_img	15817	0.0176415	0.128041	7.25797
openai__clip-vit-base-patch32_text	14699	0.011835	0.123648	10.4477
openai__clip-vit-large-patch14_img	15864	0.0181518	0.132014	7.27279
openai__clip-vit-large-patch14_text	14456	0.0120083	0.132492	11.0334
Qwen__Qwen3-1.7B-Base_text	15126	0.0111363	0.0887757	7.97173
Qwen__Qwen3-4B-Base_text	15385	0.0112631	0.0877425	7.79026

Table 14: Statistics of nonzero sparse feature entries for Visual Genome (truncated sparse features), pre-truncation sparse dimension $d = 16384$, and pre-truncation sparsity $k = 128$.

C.8 Residuals of Sparse Autoencoders

C.8.1 Experiments on COCO

Here, we show the residuals for trained autoencoders subject to different sparsity and dimension. The main goal of this section is to demonstrate the small residual errors, thus justifying the sparsity assumption A.1) in Section 1.3.

Model	$k = 32$	$k = 64$	$k = 128$
BAAI/bge-base-en-v1.5_text	0.1875	0.1619	0.1349
BAAI/bge-large-en-v1.5_text	0.1922	0.1669	0.1425
codefuse-ai/F2LLM-1.7B_text	0.2631	0.2413	0.2261
codefuse-ai/F2LLM-4B_text	0.3041	0.2848	0.2742
facebook/dinov2-base_img	0.3919	0.3476	0.3008
facebook/dinov2-large_img	0.395	0.36	0.3219
facebook/vit-mae-huge_img	0.318	0.2717	0.2333
facebook/vit-mae-large_img	0.3392	0.278	0.2332
google/gemma-3-1b-it_text	0.2771	0.2337	0.2009
google/gemma-3-4b-it_text	0.2798	0.2424	0.2151
google/siglip2-base-patch16-256_img	0.3411	0.2869	0.2401
google/siglip2-base-patch16-256_text	0.2077	0.1786	0.1546
google/siglip2-large-patch16-256_img	0.3424	0.2879	0.2442
google/siglip2-large-patch16-256_text	0.2251	0.195	0.1686
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_img	0.3569	0.2947	0.2331
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_text	0.1808	0.1537	0.1292
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_img	0.3777	0.327	0.2796
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_text	0.1997	0.1718	0.1488
meta-llama/Llama-3.2-1B-Instruct_text	0.2827	0.239	0.209
meta-llama/Llama-3.2-3B-Instruct_text	0.2729	0.233	0.2064
microsoft/beit-base-patch16-224_img	0.4253	0.3786	0.3287
microsoft/beit-large-patch16-224_img	0.4536	0.4148	0.3703
nomic-ai/nomic-embed-text-v1.5_text	0.1673	0.1444	0.1183
nomic-ai/nomic-embed-text-v2-moe_text	0.1888	0.1581	0.1343
openai/clip-vit-base-patch32_img	0.33	0.274	0.2222
openai/clip-vit-base-patch32_text	0.1953	0.166	0.1376
openai/clip-vit-large-patch14_img	0.3687	0.3201	0.2723
openai/clip-vit-large-patch14_text	0.2135	0.1861	0.1596
Qwen/Qwen3-1.7B-Base_text	0.2942	0.2571	0.2311
Qwen/Qwen3-4B-Base_text	0.286	0.248	0.2208

Table 15: Residuals for dataset coco, sparse dimension d=8192, and k in {32, 64, 128}.

Model	$k = 32$	$k = 64$	$k = 128$
BAAI/bge-base-en-v1.5_text	0.1775	0.1523	0.1282
BAAI/bge-large-en-v1.5_text	0.1812	0.1566	0.1342
codefuse-ai/F2LLM-1.7B_text	0.2449	0.2227	0.2109
codefuse-ai/F2LLM-4B_text	0.2817	0.2616	0.2532
facebook/dinov2-base_img	0.3551	0.3135	0.2682
facebook/dinov2-large_img	0.3578	0.3249	0.2877
facebook/vit-mae-huge_img	0.3039	0.253	0.2182
facebook/vit-mae-large_img	0.3278	0.259	0.219
google/gemma-3-1b-it_text	0.2626	0.2188	0.1886
google/gemma-3-4b-it_text	0.2657	0.2299	0.203
google/siglip2-base-patch16-256_img	0.3191	0.2622	0.2221
google/siglip2-base-patch16-256_text	0.1955	0.1685	0.1465
google/siglip2-large-patch16-256_img	0.3224	0.2635	0.2262
google/siglip2-large-patch16-256_text	0.212	0.1823	0.1589
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_img	0.3307	0.267	0.2176
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_text	0.1695	0.144	0.1227
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_img	0.3518	0.2973	0.2593
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_text	0.1882	0.1612	0.1405
meta-llama/Llama-3.2-1B-Instruct_text	0.2692	0.2257	0.197
meta-llama/Llama-3.2-3B-Instruct_text	0.2619	0.2225	0.1959
microsoft/beit-base-patch16-224_img	0.3877	0.3418	0.2946
microsoft/beit-large-patch16-224_img	0.4141	0.3753	0.3336
nomic-ai/nomic-embed-text-v1.5_text	0.1581	0.1351	0.1113
nomic-ai/nomic-embed-text-v2-moe_text	0.1789	0.1486	0.1275
openai/clip-vit-base-patch32_img	0.3054	0.2497	0.2069
openai/clip-vit-base-patch32_text	0.1833	0.1555	0.1306
openai/clip-vit-large-patch14_img	0.3396	0.2915	0.2515
openai/clip-vit-large-patch14_text	0.1994	0.1744	0.1513
Qwen/Qwen3-1.7B-Base_text	0.2784	0.2426	0.2172
Qwen/Qwen3-4B-Base_text	0.2719	0.2347	0.2086

Table 16: Residuals for dataset coco, sparse dimension d=16384, and k in {32, 64, 128}.

C.8.2 Experiments on CC3M

Model	$k = 32$	$k = 64$	$k = 128$
BAAI/bge-base-en-v1.5_text	0.294	0.2581	0.2207
BAAI/bge-large-en-v1.5_text	0.305	0.2717	0.2379
codefuse-ai/F2LLM-1.7B_text	0.3925	0.3703	0.3536
codefuse-ai/F2LLM-4B_text	0.4361	0.4098	0.3927
facebook/dinov2-base_img	0.4345	0.386	0.338
facebook/dinov2-large_img	0.451	0.4106	0.3704
facebook/vit-mae-huge_img	0.3056	0.2633	0.2236
facebook/vit-mae-large_img	0.3292	0.2747	0.2275
google/gemma-3-1b-it_text	0.3862	0.3447	0.3081
google/gemma-3-4b-it_text	0.3787	0.3444	0.3175
google/siglip2-base-patch16-256_img	0.3581	0.3085	0.2599
google/siglip2-base-patch16-256_text	0.3132	0.2789	0.2422
google/siglip2-large-patch16-256_img	0.3657	0.3149	0.2686
google/siglip2-large-patch16-256_text	0.346	0.3102	0.2722
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_img	0.3696	0.312	0.2476
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_text	0.3021	0.2638	0.2261
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_img	0.407	0.36	0.3075
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_text	0.3483	0.3162	0.2775
meta-llama/Llama-3.2-1B-Instruct_text	0.3323	0.2975	0.2715
meta-llama/Llama-3.2-3B-Instruct_text	0.3399	0.307	0.2813
microsoft/beit-base-patch16-224_img	0.4645	0.4152	0.3631
microsoft/beit-large-patch16-224_img	0.498	0.4586	0.4115
nomic-ai/nomic-embed-text-v1.5_text	0.2695	0.2355	0.1983
nomic-ai/nomic-embed-text-v2-moe_text	0.276	0.2409	0.2103
openai/clip-vit-base-patch32_img	0.3583	0.3013	0.2431
openai/clip-vit-base-patch32_text	0.3167	0.276	0.2317
openai/clip-vit-large-patch14_img	0.4035	0.3552	0.3026
openai/clip-vit-large-patch14_text	0.3508	0.3145	0.2725
Qwen/Qwen3-1.7B-Base_text	0.3252	0.2935	0.2681
Qwen/Qwen3-4B-Base_text	0.3256	0.2934	0.2688

Table 17: Residuals for dataset cc3m, sparse dimension d=8192, and k in {32, 64, 128}.

Model	$k = 32$	$k = 64$	$k = 128$
BAAI/bge-base-en-v1.5_text	0.2689	0.2369	0.2039
BAAI/bge-large-en-v1.5_text	0.2809	0.2496	0.219
codefuse-ai/F2LLM-1.7B_text	0.3595	0.3358	0.3226
codefuse-ai/F2LLM-4B_text	0.4031	0.3751	0.3586
facebook/dinov2-base_img	0.4042	0.3594	0.3122
facebook/dinov2-large_img	0.4169	0.3797	0.3404
facebook/vit-mae-huge_img	0.2945	0.251	0.2135
facebook/vit-mae-large_img	0.3194	0.2617	0.2177
google/gemma-3-1b-it_text	0.3602	0.3175	0.2861
google/gemma-3-4b-it_text	0.3545	0.3196	0.294
google/siglip2-base-patch16-256_img	0.3387	0.2894	0.2465
google/siglip2-base-patch16-256_text	0.2887	0.2582	0.2283
google/siglip2-large-patch16-256_img	0.3459	0.2923	0.2533
google/siglip2-large-patch16-256_text	0.3208	0.2857	0.2547
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_img	0.3475	0.2922	0.2366
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_text	0.2759	0.2413	0.2105
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_img	0.3816	0.3348	0.291
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_text	0.3218	0.2906	0.2598
meta-llama/Llama-3.2-1B-Instruct_text	0.3104	0.2749	0.2514
meta-llama/Llama-3.2-3B-Instruct_text	0.3183	0.2853	0.2607
microsoft/beit-base-patch16-224_img	0.4346	0.3879	0.3389
microsoft/beit-large-patch16-224_img	0.4649	0.4286	0.3868
nomic-ai/nomic-embed-text-v1.5_text	0.2454	0.2153	0.1813
nomic-ai/nomic-embed-text-v2-moe_text	0.2498	0.2205	0.1946
openai/clip-vit-base-patch32_img	0.3381	0.2838	0.2336
openai/clip-vit-base-patch32_text	0.2907	0.2529	0.2184
openai/clip-vit-large-patch14_img	0.3774	0.3324	0.2877
openai/clip-vit-large-patch14_text	0.3232	0.291	0.2579
Qwen/Qwen3-1.7B-Base_text	0.3044	0.2725	0.2496
Qwen/Qwen3-4B-Base_text	0.3053	0.274	0.2503

Table 18: Residuals for dataset cc3m, sparse dimension d=16384, and k in {32, 64, 128}.

C.8.3 Experiments on Visual Genome

Model	$k = 32$	$k = 64$	$k = 128$
BAAI/bge-base-en-v1.5_text	0.2294	0.1795	0.1412
BAAI/bge-large-en-v1.5_text	0.2268	0.1796	0.1431
codefuse-ai/F2LLM-1.7B_text	0.3082	0.2586	0.2302
codefuse-ai/F2LLM-4B_text	0.3321	0.2852	0.2589
facebook/dinov2-base_img	0.3875	0.3431	0.2962
facebook/dinov2-large_img	0.3908	0.356	0.3179
facebook/vit-mae-huge_img	0.3182	0.2718	0.2335
facebook/vit-mae-large_img	0.3385	0.2777	0.2333
google/gemma-3-1b-it_text	0.2994	0.2401	0.2033
google/gemma-3-4b-it_text	0.2907	0.243	0.2097
google/siglip2-base-patch16-256_img	0.3372	0.2823	0.2365
google/siglip2-base-patch16-256_text	0.2601	0.2085	0.1727
google/siglip2-large-patch16-256_img	0.3392	0.2837	0.2399
google/siglip2-large-patch16-256_text	0.2829	0.2296	0.1919
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_img	0.3516	0.2896	0.2296
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_text	0.2426	0.1938	0.1586
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_img	0.3716	0.3208	0.2744
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_text	0.2678	0.2172	0.1815
meta-llama/Llama-3.2-1B-Instruct_text	0.2692	0.215	0.1814
meta-llama/Llama-3.2-3B-Instruct_text	0.2748	0.2221	0.1881
microsoft/beit-base-patch16-224_img	0.4204	0.3736	0.3237
microsoft/beit-large-patch16-224_img	0.4504	0.411	0.3662
nomic-ai/nomic-embed-text-v1.5_text	0.217	0.1696	0.1309
nomic-ai/nomic-embed-text-v2-moe_text	0.2395	0.1834	0.1479
openai/clip-vit-base-patch32_img	0.3248	0.269	0.2188
openai/clip-vit-base-patch32_text	0.2491	0.2004	0.1618
openai/clip-vit-large-patch14_img	0.3626	0.3144	0.2667
openai/clip-vit-large-patch14_text	0.2664	0.2187	0.182
Qwen/Qwen3-1.7B-Base_text	0.2799	0.23	0.1988
Qwen/Qwen3-4B-Base_text	0.2681	0.2183	0.1863

Table 19: Residuals for dataset visual_genome, sparse dimension d=8192, and k in {32, 64, 128}.

Model	$k = 32$	$k = 64$	$k = 128$
BAAI/bge-base-en-v1.5_text	0.2211	0.171	0.1341
BAAI/bge-large-en-v1.5_text	0.219	0.17	0.1354
codefuse-ai/F2LLM-1.7B_text	0.294	0.2425	0.2142
codefuse-ai/F2LLM-4B_text	0.3156	0.2667	0.2407
facebook/dinov2-base_img	0.3478	0.3063	0.2608
facebook/dinov2-large_img	0.3508	0.318	0.2804
facebook/vit-mae-huge_img	0.3015	0.2512	0.2177
facebook/vit-mae-large_img	0.327	0.2568	0.2196
google/gemma-3-1b-it_text	0.2863	0.2244	0.1899
google/gemma-3-4b-it_text	0.2797	0.2298	0.1985
google/siglip2-base-patch16-256_img	0.3147	0.257	0.219
google/siglip2-base-patch16-256_text	0.2496	0.1959	0.1641
google/siglip2-large-patch16-256_img	0.318	0.258	0.2223
google/siglip2-large-patch16-256_text	0.2706	0.2147	0.1809
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_img	0.3247	0.2601	0.2141
laion/CLIP-ViT-B-32-laion2B-s34B-b79K_text	0.2319	0.1829	0.1499
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_img	0.3437	0.2901	0.2535
laion/CLIP-ViT-H-14-laion2B-s32B-b79K_text	0.2564	0.2042	0.1714
meta-llama/Llama-3.2-1B-Instruct_text	0.2606	0.2025	0.1715
meta-llama/Llama-3.2-3B-Instruct_text	0.2655	0.211	0.1799
microsoft/beit-base-patch16-224_img	0.3803	0.3344	0.2864
microsoft/beit-large-patch16-224_img	0.4077	0.3686	0.3261
nomic-ai/nomic-embed-text-v1.5_text	0.2092	0.1608	0.1247
nomic-ai/nomic-embed-text-v2-moe_text	0.2323	0.1738	0.1399
openai/clip-vit-base-patch32_img	0.299	0.2433	0.2013
openai/clip-vit-base-patch32_text	0.2377	0.1876	0.1531
openai/clip-vit-large-patch14_img	0.3318	0.284	0.2443
openai/clip-vit-large-patch14_text	0.2542	0.2055	0.172
Qwen/Qwen3-1.7B-Base_text	0.2689	0.2172	0.1879
Qwen/Qwen3-4B-Base_text	0.2583	0.2077	0.1775

Table 20: Residuals for dataset visual_genome, sparse dimension d=16384, and k in {32, 64, 128}.