# A Preliminary Investigation Into Interpretable Adversarial Attacks Through Feature Interference in Superposition

Daniel Wilkinson* and Chris Anley

NCC Group plc

**Abstract**

Recent work in mechanistic interpretability has shown that underparameterised neural networks tend to represent features of their training data using combinations of neurons within a layer. The resultant feature vectors are not perfectly orthogonal to each other, and slightly interfere with each other. During training, for a training scheme to be effective, the model must learn to account for the interference that results from combinations of features in the training dataset. However, we find that out-of-distribution combinations of features can be used to constructively interfere in the direction of a target feature. This constructive interference can be used to create low frequency adversarial samples, which effectively control the model by exploiting the geometry of features packed into a toy model using mechanistic interpretability techniques, rather than exploiting the fragility of the features themselves with gradient descent-based methods. The resulting adversarial samples are expected to be more resistant to real-world perturbations as they use legitimate feature detectors trained into the model. We additionally investigate whether larger models would be likely to have sufficient feature interference for a tractable number of features to collectively have a meaningful sway over a target feature. We believe that this area warrants further attention to determine whether the attack would scale to frontier models, as perturbation-resistant human-interpretable adversarial samples could have a large impact on the security of large ML models.

## 1 Introduction

Recent progress in machine learning with the introduction of large models, such as large language models (LLMs) and diffusion models, have shown a significant change to parameter count over earlier machine learning techniques. However, the scale of data used to train these models has grown at an even greater rate, meaning that from a feature-packing perspective, these large models behave as though they are severely underparameterised. Researchers at Anthropic have shown that when encoding more features than there are available dimensions in the activation space of a layer of an artificial neural network, combinations of neurons can be observed working together to represent features as vectors which to not align with the basis vectors of the activation space (i.e. individual neuron activations).[1] In their investigation, the authors simulate vectors of independent feature activations by generating random values, which are set to 0 with a probabilistic sparsity filter. They then train a model similar to an autoencoder to compress these features into a smaller space, with the model using the same projection layer as the down projection transposed for the up projection. They investigate the effects of feature sparsity, and feature correlation on the number of

*Work conducted while at NCC Group. Current contact address: Amber Wolf, daniel.wilkinson@amberwolf.com

features that are learned, as well as the effect on the geometry of the compressed activation space. We adopt their term 'superposition' to reference the phenomenon of non-orthogonal features being packed into the model's weights.[1] Subsequent work has demonstrated that large language models exhibit superposition within their activations, and that sparse autoencoders (SAEs) can be used to project activations into a higher dimensional space, where features align with single neurons, allowing their activations to be interpreted more easily.[2][3]

A natural consequence of superposition is that since feature vectors are not all perfectly orthogonal, it is not always possible to perfectly reconstruct the feature activations from only the neuron activations; features can interfere with each other based on the angle between their respective vectors in activation space. Elhage et al. were able to demonstrate that the number of features that are learned is a balance between its value for the given task, compared to the interference that it has with other features. They showed that feature sparsity, and importance to the task are strong factors, and also briefly considered the impact of feature correlation on the number of features learned.[1]

Our hypothesis is that during training, models learn to account for the interference between combinations of features that are likely to appear together in the training data, but that out-of-distribution combinations of features may collectively interfere with some target feature sufficiently for subsequent layers of the model to consider the target feature as active. We propose that there is limited training signal to penalise interference between out-of-distribution combinations precisely because they are out-of-distribution, and therefore do not cause errors during training that would increase loss. Therefore, we predict that it would be possible to create a crafted adversarial input, which does not contain a target feature, but does contain a combination of seemingly unrelated features, which would constructively interfere to steer model behaviour. In this way, an adversary could attack a model to obtain fine-grained control over it (e.g. by activating behavioural features), by supplying an input that a knowledgeable human could construct using mechanistic interpretability techniques, but that other humans would not identify as malicious. If the attack works against large language models, then perhaps an adversary could activate a 'bomb' feature in subsequent layers to the model's safety circuitry.[4] Alternatively, if the attack works on large models used for vision in embodied applications, then perhaps an adversary could manipulate the model's understanding of its environment to cause it to cause harm.

We find in this initial work that this attack is successful in small toy models trained to classify the MNIST dataset, which are forced into using superposition.[5] We additionally provide some evidence to suggest that with additional effort, the attack could be adapted to work on current large models.

The topic of adversarial inputs to control model behaviour is not new. Many prior techniques exist, which often primarily use hill-climbing algorithms to maximise the probability that a model will match some target output by making small perturbations to the input.[6][7] While the authors show strong results against image classifiers, their methods require accurate fine-grained control of the input, and primarily optimise for target model outputs, without being able to control logic flow within the model.

We aim to provide a method of generating interpretable adversarial samples, which can be used with multiple modalities, where exact input control is not possible, such as by constructing an unusual environment for an embodied agent to record on an attacker-inaccessible camera. We also aim to provide a mechanism for more targeted model manipulation, to activate specific features rather than optimising for specific outputs. For example, if applied to a large language model, to bypass safety circuitry[4] by having unrelated features activate the "bomb" feature in subsequent layers, after the model has decided not to refuse the user's question. Real-world attacks against the transformer architecture run into issues with gathering features together through the attention

mechanism, as is discussed in Section 4, however if this challenge is overcome, then the attack may permit sophisticated jailbreaking attacks despite existing safety training.

To assess our hypothesis, we train a small MLP classifier on the MNIST dataset,[5] which is branched to act as an overparameterised model on one branch, and also to pass feature activations through an underparameterised layer to force the model to use superposition in the other. We use MNIST as a well-studied dataset with clear identifiable features. We analyse the latent space to identify adversarial feature combinations that would interfere with a target feature, and construct adversarial examples to abuse this interference. We further show that interpretable samples can be crafted by eye provided that the identified interfering features are interpretable.

We make no claims on whether this method explains traditional adversarial perturbation attacks, but offer it as an alternative that can construct samples that exploit the geometry of the packing of feature vectors in a model, rather than exploiting the fragility of features themselves. This alternative could be useful in real-world attacks where adversaries do not have pixel-level control over the input to the model, and as a method for humans to be able to manually construct adversarial attacks without the use of gradient descent in sample creation.

Finally, we acknowledge that there is a substantial difference between a working proof-of-concept in a small toy model, and a repeatable exploit in a full-size frontier model. This is especially the case when aiming to exploit a lack of orthogonality in high-dimensional space. We repeat and attempt to build on part of Anthropic's work on toy models, to provide results from experiments which show trends in interference angles as model capacity, feature sparsity, and crucially for real-world applicability, feature correlation are varied.

Our contributions are:

- a method for an interpretable adversarial attack with an understanding of the mechanisms of its effect on the model, which works on real features extracted from the MNIST dataset
- additional datapoints around feature interference as model size and feature correlation change, as well as an investigation into feature interference in Gemma-Scope SAE

## 2 Related Work

This work aims to link the plethora of existing work in adversarial attacks on ML models with the more recent mechanistic interpretability work on superposition in underparameterised models.

Adversarial examples were initially identified in 2013 by Szegedy et al.[6], who found that images close to a target sample would be misclassified by image classifiers. This work was extended by Goodfellow et al., who used a fast gradient based optimisation on target samples to find nearby adversarial samples. While this technique has proven effective against a variety of models, it requires very fine control over the pixel values in an image.

In order to transfer adversarial attacks to the physical world, Eykholt et al.[7] optimised to find perturbations which consistently caused misclassification in image classifiers across a variety of real-world conditions. Their method involved sampling a number of images of the target object, from different viewing angles and in different conditions. Additionally, they applied synthetic transformations, such as cropping, brightness adjustments, and spatial transformations. To minimise the human-detectability of their attack, they restrict the final perturbation to a mask with the most impact on the classifier. They were able to show that printed perturbation matrices could be stuck to STOP signs in the real world, to reliably cause misclassification in their target models.

Tsipras et al.[8] subsequently went on to show that 'fragile features' are a natural consequence of training for accuracy in classification tasks, as there is training signal from using spurious cor-

relations in the training data, which may open the model to be more susceptible to adversarial attack. While the findings of this paper do not concern themselves with the feature geometry or inner workings of the model, the identified trade-off between standard accuracy and adversarial robustness appears to hold true for the problem of superposition finding the optimal number of features to encode in the model's weights.

Similarly, in the field of natural language processing (NLP), large language models have been shown to be susceptible to adversarial attacks. Typically referred to as 'jailbreaking', various techniques have been identified to bypass the safety training of large language models, in order to cause them to generate unsavoury outputs. Many rely on human-understandable tricks, such as instructing a model to start its response with "Absolutely! Here's " after asking for unethical information.[9] However, in the context of this work, a more applicable method was identified by Zou et al, who were able to construct universal adversarial suffixes using a "Greedy Coordinate Gradient" algorithm, which finds a sequence of tokens to append to a malicious prompt, which will cause the model to comply.[10]

Recent mechanistic interpretability work by Anthropic has introduced the concept of superposition, where it was observed that combinations of neurons in an artificial neural network can work together to encode features of their input data.[1] Where in an overparameterised network (i.e. there are more neurons in a layer than features to encode), an ideal model would fire a single neuron to represent the presence of a feature, in underparameterised networks, the vectors that represent a feature in the activation space will not all be aligned with the basis vectors of the space (i.e. the individual neurons). If the number of features encoded is greater than the number of dimensions in the space, then necessarily, the feature vectors cannot all be orthogonal. Crucially for this work, this means that independent features can interfere with each other, with the strength of the interference affected by the angle between the vectors (their cosine similarity).

# 3 The Attack

The aim of our attack is to use an out-of-distribution combination of features, which collectively interfere with some target feature sufficiently for the model to make an error during feature decoding, concluding that the target feature was active despite not being present in the input.

For learned feature-vectors

$$\{v_1, ..., v_n\} \subset \mathbb{R}^D$$

We aim to construct an input such that the total interference on target feature $t$ from activations $a$ of other features is:

$$I_t = \sum_{i \neq t} \left| \frac{a_i v_i \cdot v_t}{||v_t||} \right| > \max(a)$$

$||v_t||$ is Euclidean norm ($\sqrt{v_t \cdot v_t}$)

## 3.1 The model

Our experiment requires investigation of feature vectors in activation space, with an under-parameterised activation space to force the model into superposition. To isolate the effect of superposition from other aspects of machine learning, we train a branched model.

- One branch acts as a simple single hidden-layer MLP with ReLU activation. We use a large hidden dimension to provide an excess of capacity for features, and following Bricken et al.[2], we add an L1 loss on hidden layer activations to encourage sparsity of activations. The intention is to form monosemantic features in a small model, in order to simulate Bricken et al.'s observations on large models which can be tracked through the forced superposition.
- The second branch, which is trained at the same time as the first, has broad similarity to Elhage et al.'s toy model setup, using a single linear projection matrix as an auto-encoder to compress high dimensional features into a smaller superposition space, and using the same matrix transposed (with a bias and ReLU activation) to recover the high-dimensional features. The same final classification layer is used on the recovered features as the original features from the first branch. Both sets of class predictions are output by the model to contrast the effect of superposition.

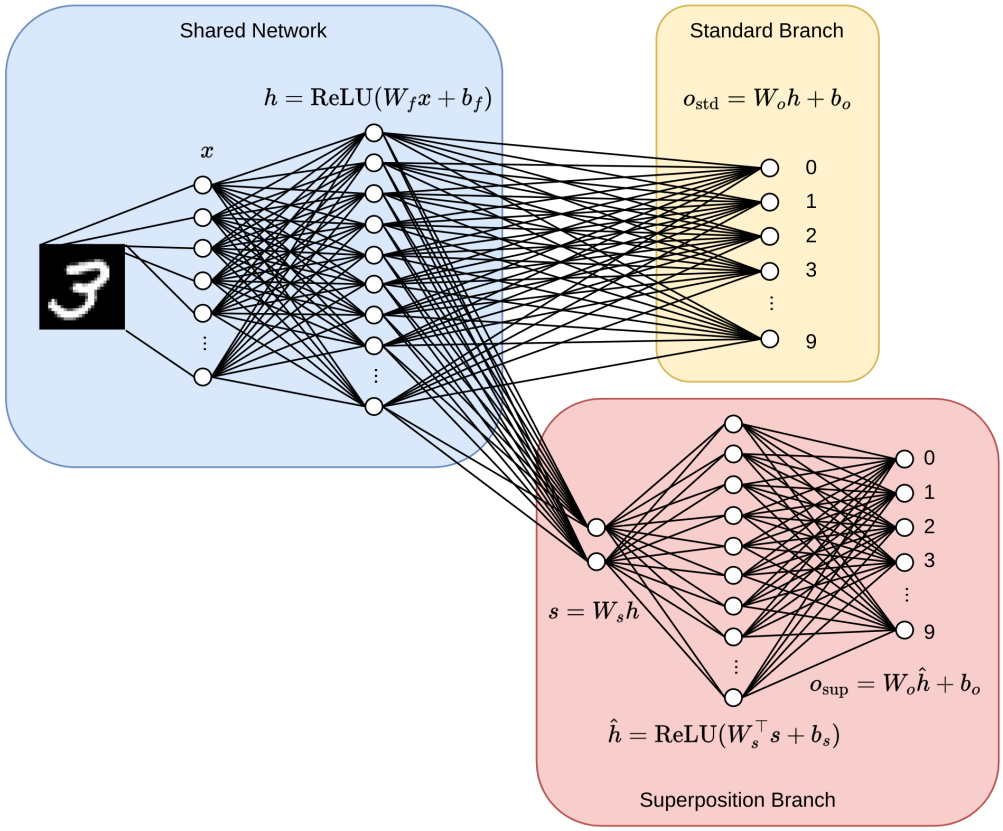


Figure 1: Main experiment model structure

### 3.1.1 Terminology

Throughout this paper, we will refer to the hidden layer of the first branch of the model as "the features", and the outputs of the first branch as the "standard predictions". For the second branch, we will refer to the projection matrix as the "encoder matrix", the down-projected space as the "superposition space", the up-projected values (after bias and ReLU) as "recovered features", and refer to the outputs of the second branch as "superposition predictions".

### 3.1.2 The training

We trained the model on the MNIST dataset, as a well-researched dataset which has easily identifiable feature. We use a loss function consisting of categorical cross-entropy loss on the predictions from both branches of the model each with a coefficient of 1. Additionally, an L1 penalty was added for the feature activations, with a coefficient of $3e-4$. This penalty encourages sparsity of feature activations, which aligns features to neurons in this layer, and encourages only as many features to be learned as are necessary to accomplish the task. Our model used a feature dimension of 128 (which we empirically found to be an excess of features needed for the task of MNIST classification) with a superposition layer dimension of 2. The model was optimized using AdamW with a learning rate of $1e-3$ and betas $(0.9, 0.999)$, for 5 epochs of the training dataset.

This loss function optimised for task performance through both model branches, as opposed to reconstruction accuracy of features, in order to provide more freedom for the model to organise the superposition latent space in the way most effective for the task, rather than strictly adhering to the instrumental goal of feature reconstruction.

Empirically, we note that for in-distribution samples, the effect of superposition on model accuracy is modest; after training, our test split accuracy was 95.81% on the standard branch, versus 94.34% on the predictions after superposition.

TKTK Add a graph of loss / accuracy between the two branches.

## 3.2 Interpreting the model

We first looked at the feature activations for a batch of 64 samples from the test set. Figure 2 shows features along x-axis and classes up y-axis. Each cell shows the mean activation of the feature for all samples in the batch from a given class. We observe that most features were "dead features" which were not used by the model, with zero activations across the entire batch. This confirmed that the model has an excess of space to learn useful features in high dimension, and was constrained by the capacity of the superposition space. Of the active features, most only activate for a single class, and the remaining mostly have a single class that they activate much more strongly for. Where this is not the case, for example for feature 34, the strong activations are in class 4 and 9, with the feature generally looking for a loop in the upper portion of the image, which understandably would trigger for classes 4 and 9.


Figure 2: Feature activations by class

TKTK add x tick for 34

For the rest of the analysis, we filter the features to only consider those that are active for at least one class.

### 3.2.1 Exploring the superposition latent space

We then plotted the vectors of each feature's row in the encoder matrix on a 2-D plane, to show its direction in superposition space. The colour for each feature is determined by the class with the highest mean activation of the feature. As we do not include any normalisation within our model,

we equally here do not apply any normalisation to the magnitude of vectors, so that we can also assess the magnitude of their influence on superposition activations.

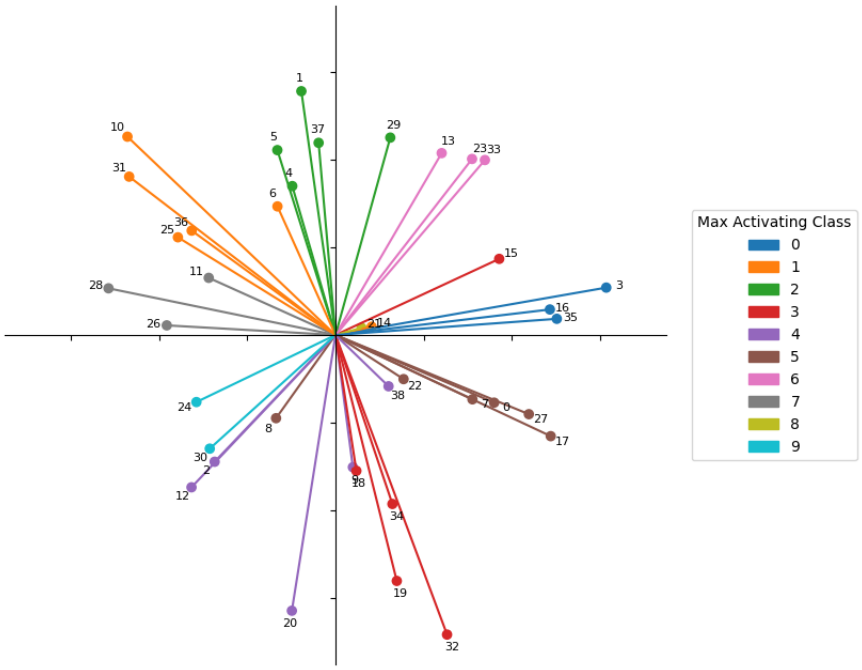


Figure 3: Feature vectors in superposition

TKTK add colouring to demonstrate the angles associated with each class, or convex hulls around the polygons

We observe that nearly 40 features are encoded in the space, which is far higher than Anthropic observed with independent features in 'Toy Models of Superposition', and even than their observations when simulating correlation in the same paper. We hypothesise that this is a consequence of higher rates of correlation in real-world data than randomly sampled independent features. This is discussed further in Section 5.

Critically, we note that the angle between many features is very small, and that the angle between nearly-independent groups of features (i.e. classes) is also substantially less than orthogonal. As a result, activating features from multiple different classes can be expected to have a resultant vector that is aligned with another class.

For example, we can see that class 3 and class 0 enclose class 5, with roughly equal angles between the three classes. Therefore, while the model is able to accurate account for interference between combinations of features that naturally appear in the training dataset, it is fragile to out-of-distribution combinations of features of 3s and 0s.

### 3.2.2 Exploring the features

In most interpretability work, identifying the purpose of features can be a very involved process, finding inputs that maximally activate the feature, and investigating similarities between activating samples. In this work, we take advantage of the simplicity of our model; our features are a single linear projection from the pixel values, followed by a ReLU activation. Therefore, for each feature, we can render the weights of each pixel to visually see the maximally activating image. We can

then use these images to steer the adversarial sample creation by making our samples similar to a candidate feature's maximally activating image.

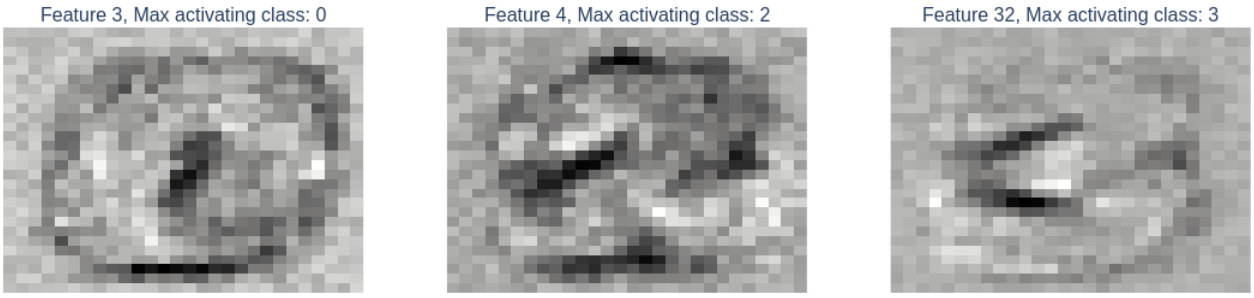


Figure 4: Example maximally activating images for features

### 3.3 Attacking the Interference

In Figure 3, we can see that features 3 and 32 span class 5, and if their vectors are added together, the resultant vector would align with the features for 5s. The most simple attack to exploit this, given our privileged position with maximal images for each feature is to simply add the maximal images for features 3 and 32.

Figure 5: Adversarial sample constructed with simple addition of maximally activating images for two features

This somewhat visually appears to be a combination of a 3 and a 0, although subsequent sections will improve on the interpretability of the adversarial sample.

Figure 6 shows the feature activations when this sample is passed through the model. The blue line shows the feature activations on the sample. We can see that the two highest activating features are features 3 and 32; the two images that we added to make the sample. The superposition layer activations on the sample were $(8.9919, -3.4129)$, which is aligned with the features for class 5. As a result, when considering the red line, which shows the reconstructed feature activations, we can see that features 17, 27, 0, and 7, which form the strong cluster of features for class 5 are all active.

When the output layer then classifies the input, we observe that in the standard branch of the model, none of the classes activate strongly, as the input is not strongly in-distribution of any class, but the highest probability is for class 3. However, in the superposition branch, the model strongly classifies the input as class 5.

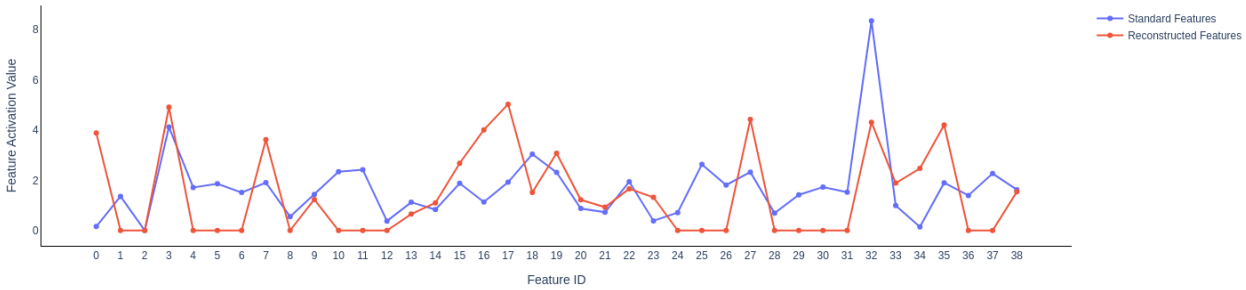


Figure 6: Effect of adversarial sample on feature activations before and after superposition

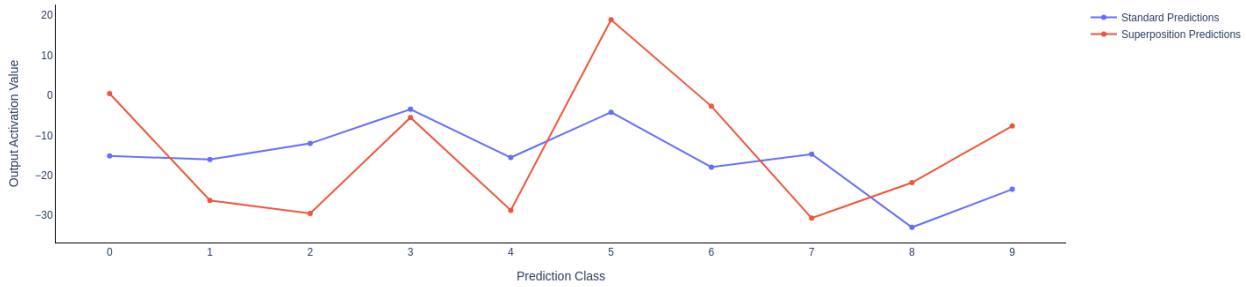


Figure 7: Effect of adversarial sample on class predictions before and after superposition

### 3.3.1 Interpretable adversarial samples

Given that we are exploiting combinations of features, our attacks can be as robust or fragile as the features themselves are. In this model, features are relatively robust, with a significant portion of the classification ability seeming to use negative space in the image (i.e. if there are active pixels away from the central columns of the image, the features for class 1 are inhibited). With knowledge of what each feature is detecting, we can construct adversarial samples by eye, which do not require exact pixel values. Consider the following manually constructed combination of a 3 and a 0, which only uses the minimum and maximum pixel values.

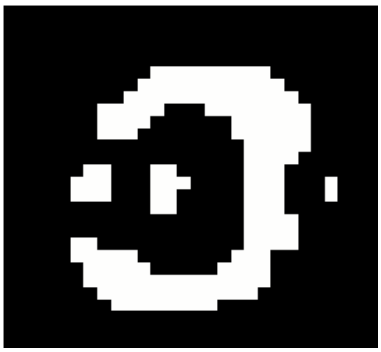

Figure 8: Manually constructed adversarial sample

This image is closer to the training distribution, which leads to less noise among other unrelated features. In the feature activation plot for this sample, we can see features 3 (class 0), 18 (class 3) and 32 (class 3) activating, with all other features inactive. The superposition activations for

this sample are $(1.3838, -1.2240)$. Again, this coordinate is closest aligned to the class 5 features, resulting in our target features (17, 27, 0, 7) activating after feature reconstruction.

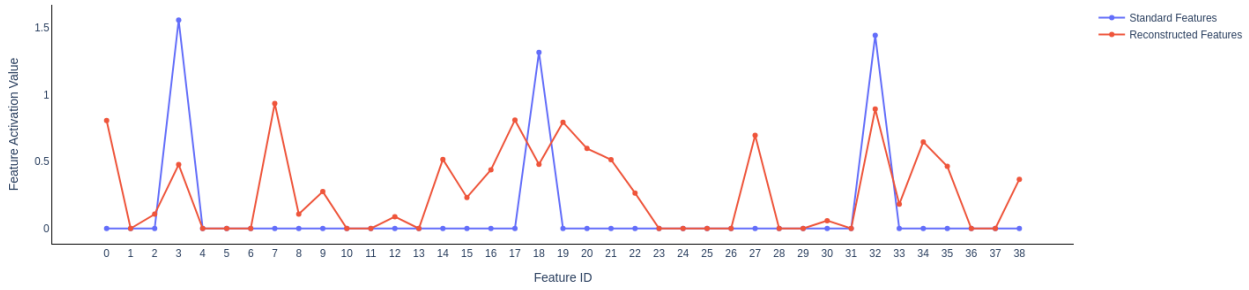


Figure 9: Effect of manual adversarial sample on feature activations before and after superposition

Again, this causes the model to output a class prediction of 5 after superposition, however now that the input is closer to the training distribution, the standard model does output a positive value for a class 3 prediction.

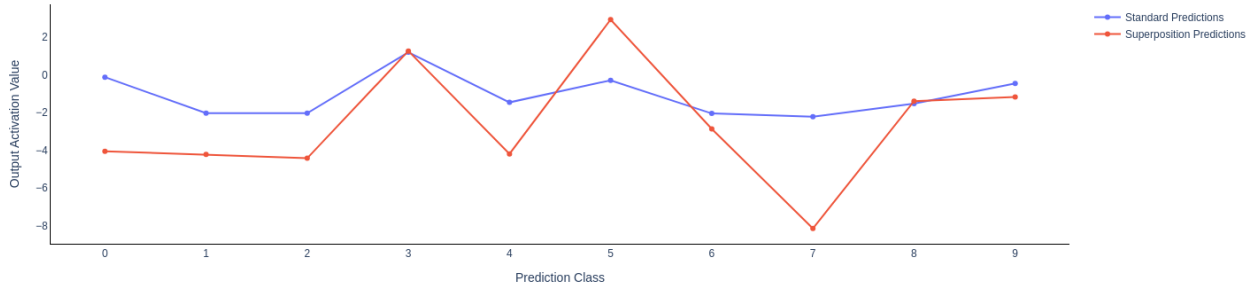


Figure 10: Effect of manual adversarial sample on class predictions before and after superposition

# 4 Applicability and Impact

## 4.1 Applicability

Having established that the class of vulnerable models exists, the next most pertinent question is how large is the class? This attack relies on a number of pre-requisites being true:

1. The model is under-parameterised or otherwise exhibits superposition
2. The angle between unrelated features is suitably off-orthogonal such that a combination of a tractable number of other features could constructively interfere to falsely trick the model into thinking that some target feature is activating
3. The attacker is able to map the activation space of the model and identify interpretable features
4. The attacker is able to combine arbitrary features within the input in a manner that they are active within the same layer of the model.

On the first pre-requisite, frontier models in image and language modelling have been shown to exhibit signs of under-parameterisation.[1][2] While to our knowledge, no definitive evidence of superposition within convolutional neural networks (CNNs) has been demonstrated, Olah et al. noted their discovery of polysemantic neurons within GoogLeNet.[11] Coupled with Anthropic's results showing that superposition tends to emerge in underparameterised contexts, this may be early evidence of superposition in CNNs. Future work investigating whether CNNs do in fact use superposition across or within feature-maps would provide further insight into their susceptability to our proposed method.

On the second, further work is required to investigate the effects of superposition as the number of dimensions in the activation space increases. As the number of dimensions in a space grows, the number of nearly-orthogonal vectors that can be packed into the space grows exponentially (as described by the Johnson-Lindenstrauss Lemma). In machine learning tasks with large datasets, such as language modelling or image generation, the number of features that would be useful to encode intuitively appears to be practically unbounded.[1] Therefore, while any given number of features in superposition in a low dimensional space could be encoded with more orthogonality in a higher dimensional space, there is pressure during training to optimise for the maximum number of features that can be safely encoded in the space while balancing the total cumulative interference as a result of encoding more features. Elhage et al. observe that the growth of the number of independent features learned grows linearly with the dimension of the superposition space, and that feature activation sparsity, and feature correlation can further increase the number of features learned. We build on their work in Section 5, and our results suggest that at the scale of current frontier models, superposition interference angles may still be suitable for our attack to work. Additionally, our experiment in Section 3 shows that real-world correlations between features behave in more complex ways than are easily represented in toy models, which can lead to significantly greater numbers of encoded features than would be predicted by toy models trained on independent features (or features using tractable simulations of feature correlation).

The third pre-requisite, mapping the model's activation space, is far from a trivial task. However, recent advancements in mechanistic interpretability; notably Anthropic's work on Sparse Auto-Encoders, and Feature Circuits, are paving the way to identifying interpretable feature directions in activation space. As mechanistic interpretability techniques improve, so too will our ability to accurately map feature vectors.[2][4]

The final pre-requisite is that an attacker must be able to combine features in a single input in order for their vectors to interfere with each other. The complexity of this task is expected to vary massively depending on model architecture and task. While an MLP trained on images, as used in this work, is an ideal candidate for this attack, other image models, such as convolutional neural networks with a structure similar to AlexNet (where a final classifier works on features extracted through convolutions) could similarly be susceptible. Transformer models, such as those typically used in Large Language Models, would likely pose a more difficult attack surface, since collecting features in any given forward pass of the model would require the attention mechanism to fetch the adversarial feature vectors and collect them in the current residual stream. Given that this attack

[1]Natural language can be thought of as a structure of complex abstractions built upon combinations or priors of other features. For example, having concepts for "parent" and "brother", we can build the concept of "uncle", which is used regularly as a single conceptual abstraction, rather than having to consider parent and brother for each mention of uncle. As the number of features increases, we get to concepts like "weltschmerz" (the feeling of sadness and lack of hope about the state of the world compared to an ideal state), which is used less often, but saves many more tokens when it does come in useful. Similarly, a wealth of information about complex relationships is encoded in the shared understanding of online meme templates, allowing high-bitrate communication with just a meme template ID, and a few words of caption text. Regardless of level of abstraction, there will likely always be more complex features that could be encoded that would be useful as high-information concepts in some scenarios.

requires out-of-distribution combinations of features, it is not clear how an attention mechanism could be tricked into collecting the target features. Future work may be required to investigate whether this is feasible.

### 4.2 Impact

Techniques to create adversarial samples for image classifiers have existed for over a decade.[6] These techniques typically involve creating a loss value based on some target mis-classified output, and backpropagating through to the input space, where adversarial noise can be optimised to incite the model to misbehave while minimising the visible effect on the input that humans would notice. These attacks typically target some desired output of the model.

If extended to be applicable to larger models, then this attack could be used to obtain more fine-grained control over model actions, beyond known target output replication. In the context of self-driving cars, attaching seemingly unrelated objects to a stop sign may constructively interfere to cause the model to view the stop sign as a green traffic light, with an impact similar to Eykholt et al's attack with apparent graffiti on stop signs.[7] In the context of language models, assuming difficulties with feature collection through attention are overcome, then perhaps Golden Gate Claude could be invoked from a crafted input. More nefariously, if the circuitry for logic learned during safety fine-tuning is found to primarily take place in early layers of a model, then it may be possible to construct an input that interferes to add the concept of "a bomb" into the residual stream after the safety logic. Alternatively, it may be possible to destructively interfere with the refusal direction, preventing the model from denying a request.

Since this attack does not rely directly on fragility of features, but rather on the geometry of feature space, it is expected to be by default more tolerant to real-world natural perturbations than the high-frequency adversarial perturbations that tend to be elicited by gradient-based methods. As a result, transference to real-world systems may be more straightforward once the basic attack method has been refined.

Most attacks in traditional cyber security rely on a thorough understanding of the target system, and as more is known, it becomes possible to craft intricate payloads to achieve complex goals. In the area of industrial control systems, safety systems have been refined over many decades, so that engineers are able to accurately model the probability of events causing safety incidents, and appropriate steps could be taken to mitigate those risks. The interplay between safety and security has only much more recently become a focus; if a safety risk has been accepted as it would only be triggered if a number of highly improbable events were to happen simultaneously, but now an adversary were able to architect a situation where those events were much more likely, then security becomes a safety issue. In a similar vein, in the field of AI safety, if we are able to show that a model is safe under normal conditions but do not consider how out-of-distribution inputs affect the features and circuits that we base our safety claims on, then we risk security issues critically exacerbating safety issues that have been accepted.

## 5 On Interference Angles

The number of features that are encoded by a model, which we will refer to as its capacity, has been observed by Anthropic to be dependent on the number of dimensions in the hidden space, the sparsity and importance of features, and correlation between features. They find that model capacity increases where features are correlated, with a 2D hidden space model learning 6 features in correlated pairs, instead of the maximum of 5 for independent features. They also discuss other

phenomena, such as local almost-orthogonal bases where the number of features to encode is less than the ultimate capacity of the model.

We extend Anthropic's work with a slight change of objective: this work introduces an attack method to control behaviour of the model, which is dependent on the angle between a hyperplane orthogonal to a target feature vector, and other feature vectors within the model, which we call the "interference angle" between the vectors. Therefore, our investigation aims to determine what happens to the worst-case interference angle between feature vectors as we vary the size of the hidden space, and correlation between features.

Following Anthropic's experimental setup, we use a vector of randomly generated feature activations as input, with values set to zero based on random samples compared to a sparsity threshold. We use their model architecture using a linear projection to compress features into a superposition space, and using the same projection matrix with a bias and ReLU activation to recover the input. We also use the same loss function, as a weighted mean squared error function weighted by feature importance.

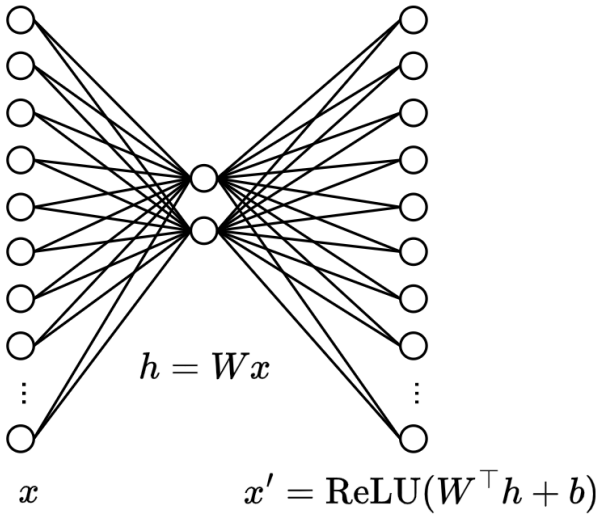


Figure 11: Toy Models of Superposition Model Diagram

### 5.1 Effect of Hidden Dimension Size on Interference Angle

We first consider how the worst-case interference angle would change as the size of the hidden space increases. In this experiment, for each feature, we find the maximum angle between all other features and the hyperplane orthogonal to the current feature as its worst-case interference angle. We then take the mean of the worst-case interference angles across all features.

$$\text{Interference Angle} = \frac{1}{|F|} \sum_{f_i \in F} \left[\max(90 - \cos^{-1}(\langle f_i, f_j \rangle)) \forall f_j \in F, i \neq j)\right]$$

$\langle x, y \rangle$ denotes cosine similarity between $x$ and $y$.

We use a constant feature probability of $20^{-1}$, constant feature importance, and an excess of input features ($n = 20 + 9 \times$ hidden_dim, found empirically to exceed the growth rate of learned features).

In line with Anthropic's observation, we find that the number of features learned in the superposition state grows linearly with the size of the superposition dimension, up to our maximum size of 1000 dimensions. However, while we initially notice an exponential decay in interference angle, levelling off around 8°, between 256 and 512 dimensions, we see an increase in mean worst-case interference angle, rising again to $\approx 12°$. We have no compelling explanation for this observation, other than speculating some form of phase change in the optimal feature packing strategy that model training tends towards.

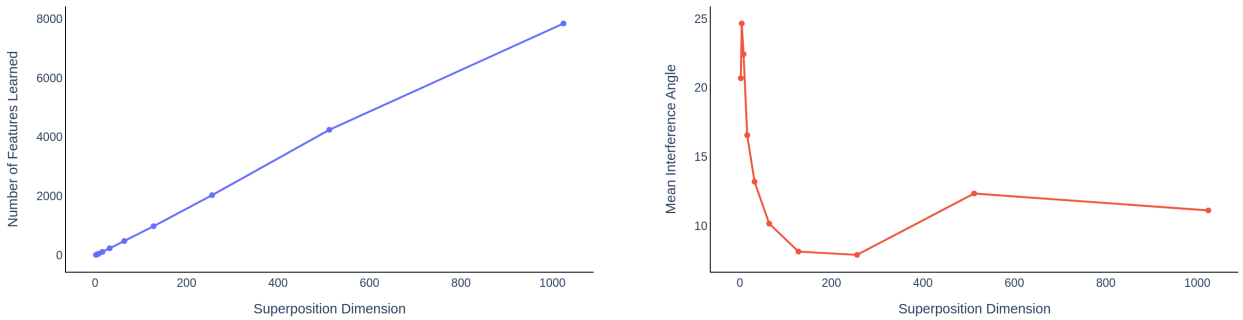


Figure 12: Linear growth rate of number of learned features with superposition dimension, and the effect on the mean worst-case interference angle between feature vectors

During training, the number of features that are learnt by a model is a balance between the value of the feature towards the task that the model is being optimised for, compared to the additional interference that could be incurred by packing an additional feature in to the activation space. As the number of dimensions in a space grows, the number of nearly orthogonal vectors that can be packed into the space grows exponentially. However, as the number of encoded features $n$ rises, the total interference grows proportional to $n \cdot \cos(\theta)$ for interference angle $\theta$. These effects appear cancel to result in a linear growth rate for the number of features encoded in superposition. See Appendix A for further details on this relationship.

This experiment was conducted against independent features, with uniform importance, and resulted in features with non-negligible interference angles in a 1024-dimension space. However, in practice, features of real-world datasets are unlikely to have exclusively independent features or uniform importance for task performance. We next turn to consider the effect of these considerations on interference angles.

## 5.2 Effect of Correlation on Interference Angle

In order to better understand the interference angles between real-world combinations of features, which have noisy correlations, we investigate the effect of imperfect correlation on features. In Toy Models of Superposition, correlation is simulated by using the same sparsity sample to determine if pairs of correlated features should be active or not. Correlated pairs of features will always both be non-zero or both be zero (unless their randomly sampled values happen to be zero). We instead implement another layer of probability to simulate imperfect correlations between features, where correlated groups of features will be probabilistically tied together during sparsity filtering. We consider the simple case with a single independent feature (yellow), and a single correlated pair of features (turquoise and purple) and observe how their feature vectors change as the correlation probability between the correlated pair increases.

We can see that initially, where all features are independent, purple and turquoise are opposite,

Figure 13: Feature vectors in activation space with a single independent feature (yellow), and a pair of probabilistically correlated features (purple and turquoise). From left to right, correlation probability increases from 0 (uncorrelated) to 1 (correlated) in increments of 0.1

with yellow orthogonal to both. As correlation probability rises, we immediate note that the correlated pair will always be adjacent so as to be as close to orthogonal to each other as possible. As correlation probability increased until the features are always zero or non-zero together, the model appears to move from prioritising orthogonality between the feature vectors, to prioritising orthogonality between the independent feature vector, and the mean vector between the correlated pair. This makes some intuitive sense if we consider the principal components in the reconstruction as whether the independent feature is active, and whether the correlated pair is active. Assigning attribution to each of the features in the correlated pair becomes a secondary objective.

From this result, and Anthropic's observations on model capacity increasing with feature correlation, we can infer that the mean feature vector for correlated groups becomes an important factor in the feature packing problem that model training optimises. However, for outliers within any correlated feature group, the worst-case interference angle between it and outliers from unrelated correlated feature groups may be worse than a model would typically accommodate with independent features.

### 5.3 Investigation of Gemma Scope

The above experiments provide speculative evidence that the interference angles in activation spaces as large as those used in even frontier LLMs may be non-negligible, leading to the possibility of this attack being effective against them, if the remaining problems with feature accumulation through attention can be defeated. We therefore turn to Gemma Scope,[12] a popular set of open weight sparse autoencoders trained on the Gemma 2 series of models. We aim to determine how susceptible Gemma may be to our attack by considering the trends in feature vector interference angles in the features learned by the SAE.

We consider the `layer_20/width_65k/average_l0_20` model, and take the weight matrix of the decoder, `W_dec`. Gemma Scope forces weight normalization, meaning that the feature vectors are all already unit length. We therefore find the cosine similarity between these feature vectors by multiplying the weight matrix by itself transposed, and zero the diagonal in the resultant matrix. Next, we sort the values along one dimension, such that the first row of the matrix shows the worst-case interference cosine similarity for each feature. Finally, we find the mean across the other dimension, so that we can plot the mean drop-off in interference across all features. The results are shown in Figure 14.

From this model, the mean worst-case interference cosine similarity across all features is 0.49. Having sorted by cosine similarity, for each feature the 27th worst interfering other feature has mean cosine similarity 0.25. As a result, any combination of 4 of the strongest interfering 27 features can be expected to cause at least equal activation of a target feature compared to a legitimate activation of that feature. Further investigation is required to attribute how much of this interference can be explained by legitimate semantic links between features, and how much is a pure symptom of packing unrelated features in superposition. However, this result does suggest that this research

direction would be worthy of further investigation.

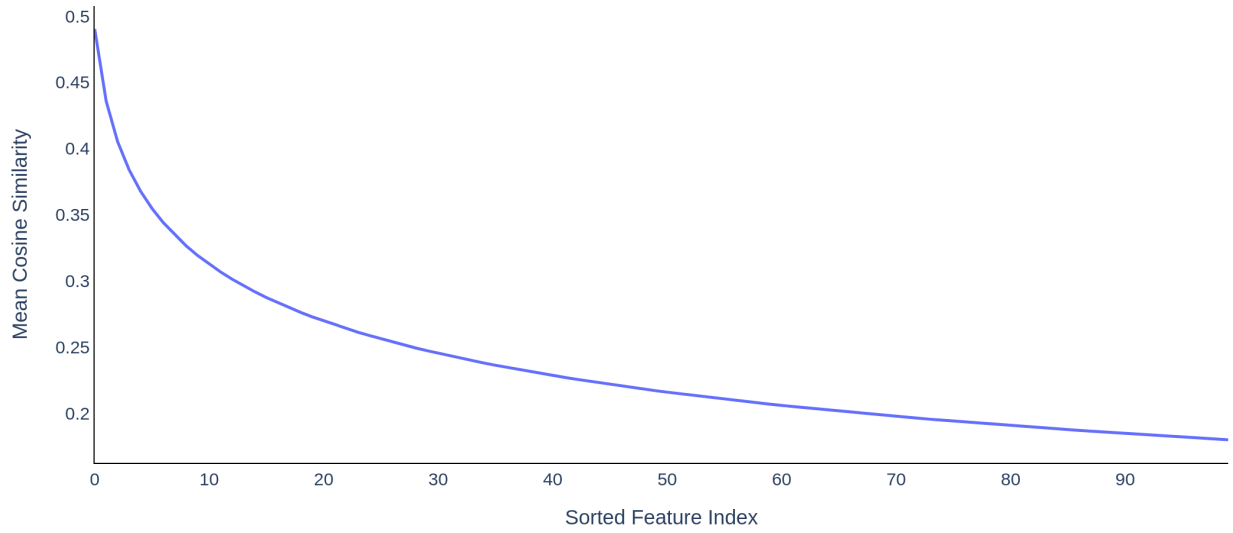


Figure 14: Gemma Scope mean sorted interference angles

## 6 Conclusion

In this paper we have introduced a new vector for creating adversarial samples, which cause a target feature to become active in the model despite that feature not being present in the input. The method builds upon recent work in mechanistic interpretability, utilising the knowledge of the inner workings of the model to generate crafted exploits. We demonstrate the effectiveness of the method on a toy model that was designed to be interpretable, and cause it to misclassify an input in a way that can be predicted using mechanistic interpretability techniques.

Additionally, we extend the work of Elhage et al.[1] to investigate whether large models are likely to be susceptible to this attack. We find early evidence that suggests it is possible that the attack could transfer to large models, based on the interference angles between features in toy models with a superposition space up to 1024 dimensions, and considering the feature vectors learned by Gemma Scope.[12] We leave practical exploitation of large models to future work.

If the attack is found to work against large models that exhibit superposition, then there is scope for a wide range of behaviours to be elicited out of these models, despite existing safety fine-tuning techniques.

## 7 Acknowledgements

Daniel Wilkinson is the lead author and researcher on the project. Chris Anley is the Chief Scientist at NCC Group and acted as the supervisor for this project, including many insightful discussions and guidance on experimentation and on the paper. Thanks to NCC Group for a total of 16 days of research time for this project, as well as access to an EC2 instance with an L4 GPU with which to conduct this research. Thanks to Liz James and Andrew Havard for proof-reading the paper.

## References


[1] Nelson Elhage et al. *Toy Models of Superposition*. 2022. arXiv: 2209.10652 [cs.LG]. URL: https://arxiv.org/abs/2209.10652.

[2] Trenton Bricken et al. "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning". In: *Transformer Circuits Thread* (2023). URL: https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[3] Adly Templeton et al. "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet". In: *Transformer Circuits Thread* (2024). URL: https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[4] Jack Lindsey et al. "On the Biology of a Large Language Model". In: *Transformer Circuits Thread* (2025). URL: https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

[5] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[6] Christian Szegedy et al. *Intriguing properties of neural networks*. 2014. arXiv: 1312.6199 [cs.CV]. URL: https://arxiv.org/abs/1312.6199.

[7] Kevin Eykholt et al. *Robust Physical-World Attacks on Deep Learning Models*. 2018. arXiv: 1707.08945 [cs.CR]. URL: https://arxiv.org/abs/1707.08945.

[8] Dimitris Tsipras et al. *Robustness May Be at Odds with Accuracy*. 2019. arXiv: 1805.12152 [stat.ML]. URL: https://arxiv.org/abs/1805.12152.

[9] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. *Jailbroken: How Does LLM Safety Training Fail?* 2023. arXiv: 2307.02483 [cs.LG]. URL: https://arxiv.org/abs/2307.02483.

[10] Andy Zou et al. *Universal and Transferable Adversarial Attacks on Aligned Language Models*. 2023. arXiv: 2307.15043 [cs.CL]. URL: https://arxiv.org/abs/2307.15043.

[11] Chris Olah et al. "The Building Blocks of Interpretability". In: *Distill* (2018). DOI: 10.23915/distill.00010. URL: https://distill.pub/2018/building-blocks.

[12] Tom Lieberum et al. *Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2*. 2024. arXiv: 2408.05147 [cs.LG]. URL: https://arxiv.org/abs/2408.05147.

# A Derivation of Linear Relationship of Number of Features and Superposition Dimension

Let

$$\{v_1, ..., v_n\} \subset \mathbb{R}^D$$

be the learned feature-vectors in the model. Assume all to be unit-norm: $||v_i|| = 1$.
Total interference seen by feature $i$:

$$I_i = \sum_{j \neq i} \langle v_i, v_j \rangle = \sum_{j \neq i} \cos\theta_{ij}$$

In toy models, assume the off-diagonal inner-products are i.i.d. with

$$\mathbb{E}[\langle v_i, v_j \rangle] = 0, \quad \text{Var}(\langle v_i, v_j \rangle) = \sigma^2 = \frac{1}{D}$$

For explanation of inner product variance, see A.1.
Mean interference:

$$\mathbb{E}[I_i] = 0$$

Variance of interference:

$$\text{Var}(I_i) = \sum_{j \neq i} \text{Var}(\langle v_i, v_j \rangle) = (n-1)\frac{1}{D} \approx \frac{n}{D}$$

As a result,

$$|I_i| \approx \sqrt{\text{Var}(I_i)} = \sqrt{\frac{n}{D}}$$

If we assume that there is a fixed allowable interference threshold that a model can accommodate while still being able to accurately decode any given feature, which we will denote $\epsilon$.

$$|I_i| \lesssim \epsilon \implies \sqrt{\frac{n}{D}} \lesssim \epsilon$$

$$n \lesssim \epsilon^2 D$$

Therefore, the optimal maximum number of features, $n$, grows linearly in the superposition dimension, $D$.

## A.1 Explanation for Variance = $\frac{1}{D}$

Let $u, v \in \mathbb{R}^D$ be two independent random unit vectors on the unit hypersphere. We're interested in the distribution of their dot product:

$$z = \langle u, v \rangle = \sum_{i=1}^{D} u_i v_i$$

We can rotate the hypersphere such that one vector is aligned to a basis vector (i.e. $u = (1, 0, 0, ..., 0)$). Then

$$\langle u, v \rangle = v_1$$

So our variance can be simplified to the distribution of the first element of $v$. Since $v$ is sampled from $\mathcal{N}(0, I_d)$ and then normalised, i.e.

$$v := \frac{w}{||w||}, \text{where } w \sim \mathcal{N}(0, I_D)$$

$$\text{Var}(\langle u, v \rangle) = \text{Var}(v_1) \approx \frac{1}{D}$$