THE EXPLAINABILITY ILLUSION: MUST VITS EXHIBIT PATCH-DECODING INTERPRETABILITY?

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers are widely assumed to exhibit token-decoding explainability. In particular, in vision transformer (ViT) classifiers, decoding individual patch tokens has become a common diagnostic tool due to its tendency to yield human-interpretable outputs. In light of the increasing popularity of this approach, an important question is whether token-decoding explainability an inherent property of transformers that should be expected to hold across architectures, scales, and training objectives, or merely an artifact of some implicit bias? In this paper we systematically study token-decoding explainability across diverse ViT settings, spanning architectures, scales, and training objectives, using both controlled experiments and real-world models. We find that while some models indeed exhibit aligned and interpretable token decodings, others exhibit either inverted explainability maps or a complete lack of correlation with the image semantics. As we show, these tendencies toward different explainability behaviors are not intrinsic to the architecture and can vary arbitrarily with changes in the optimization landscape. We provide theoretical support for these observations by demonstrating a setting where a single fixed toy transformer architecture can provably implement the optimal classification rule in multiple ways, which differ in their patch-decoding explainability patterns. Our results show that token-decoding explainability is not an inherent property of trained transformers, but rather an outcome contingent on the problem setting and thus should not be a-priori assumed to hold.

1 INTRODUCTION

Transformers (Vaswani et al., 2017), originally presented in the context of natural language processing, have proven powerful across various domains, including images, audio, and video (Arnab et al., 2021; Dosovitskiy et al., 2021; Gong et al., 2021; Yun et al., 2019), achieving state-of-the-art performance on countless tasks (Kirillov et al., 2020; Li et al., 2019; Peebles & Xie, 2023; Radford et al., 2021; Zheng et al., 2021). Following their growth in popularity, significant attention has been devoted to mechanistically understanding their internal representations.

One of the simplest methods for explainability in transformer language models is Logit Lens (nostalgebraist, 2020). This cornerstone approach interprets each intermediate hidden state of the model by projecting it back into the model's vocabulary space. This is done by multiplying the token's embedding by the unembedding matrix, which is normally applied only to the tokens of the final layer. Logit Lens has been widely adopted for exposing the evolution of predictions across layers in large language models (LLMs) (Li et al., 2024b; Pal et al., 2023; Patil et al., 2024; Prakash & Lee, 2023; Wang, 2025; Wendler et al., 2024; Wu et al., 2026), and inspired numerous derivative approaches for decoding intermediate tokens (Belrose et al., 2023; Din et al., 2024; Sakarvadia et al., 2023). It has also expanded to other domains, including images (Joseph et al., 2025; Neo et al., 2025), audio (Glazer et al., 2026; Yang et al., 2025), and video (Kim et al., 2026; Zhang et al., 2025).

In light of the widespread adoption of these techniques, it seem natural to ask whether we can universally trust the validity of patch-token decoding explainability. Much of the current literature assumes that intermediate representations consistently map directly to interpretable concepts (Jiang et al., 2025; Joseph et al., 2025; Neo et al., 2025; Vilas et al., 2023). For instance, it is commonly expected that when interpreting the classification result of a vision transformer (ViT) on an input image of an elephant, the patch-tokens corresponding to the spatial location of the elephant would

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

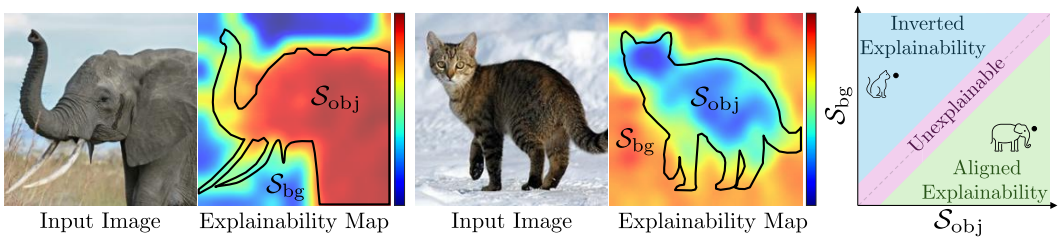


Figure 1: **Explainability regimes.** For an input image, token-decoding of patches produces explainability maps, where each patch-token receives some score $\mathcal{S}$. The relation between the average score of object tokens, $\mathcal{S}_{\text{obj}}$ (tokens inside the black contour), and the average score of background tokens, $\mathcal{S}_{\text{bg}}$ (tokens outside the black contour), determine the explainability of a classified input image in a layer: ◢ aligned, ◖ unexplainable, or ◤ inverted. Here, the elephant is aligned and the cat is inverted.

exhibit a higher logit for the "elephant" class than to any other class. This is illustrated in Fig. 1, which shows a map of the "elephant" logit across all patch tokens of the image. Several recent works noted that this might not always be the case, observing that in some transformers token-decoding maps can look inverted (Bousselham et al., 2024; Kang et al., 2025; Li et al., 2023). Yet, these observations have been treated as a localized effect to be engineered away.

In this paper, we show that there is no guarantee that transformers exhibit token-decoding explainability. We begin by analyzing dozens of pre-trained ViTs, showing they break the common assumption of *aligned*-explainability. Instead, many models exhibit either *inverted*-explainability or completely *unexplainable* intermediate representations. For example, when applying patch-token decoding explainability techniques to prominent contrastive vision-language models (VLMs) such as CLIP (Radford et al., 2021) and SigLIP (Zhai et al., 2023), the explainability maps are typically inverted. Next, through a series of controlled synthetic experiments, we expose how different conditions cause the same architecture to change its explainability behavior. We observe that different tasks and data distributions induce implicit biases toward different explainability regimes. Those biases do not deterministically determine the explainability behavior, however they control the probability of a training run to converge to either an aligned- or an inverted-explainability solution.

Finally, we prove that aligned-explainability cannot be an inherent property of all transformers. Specifically, we present a simple classification task for which the optimal classification rule admits an implementation by a toy, single layer transformer architecture in infinitely many ways. We prove that among the sets of weights that implement the optimal solution, some correspond to an aligned-explainable model and some to an inverted-explainable model.

Our findings demonstrate that the behavior and explainability of decoded representations can vary widely depending on the optimization landscape, from the task, to the training loss, and even the data distribution itself. This implies that token-decoding explainability is not a fundamental property of transformers, but merely a fragile byproduct of their training conditions. As such, when training a transformer on a new task, it cannot be a-priori assumed that zero-shot decoding of its internal states would reveal any meaningful or localized information about the semantics encoded in those tokens.

2 Related Work

Explainability techniques for ViTs. Explainability methods broadly fall into two families. The first attributes the ViT's prediction to the input image patches. Early techniques aggregated attention across layers for standard transformers (Abnar & Zuidema, 2020) and were subsequently adapted to ViTs (Dosovitskiy et al., 2021). Later methods propagated relevance through self-attention in both vision and language models (Achtibat et al., 2024; Ali et al., 2022; Chefer et al., 2021b). Other approaches rely on gradient-based weighting of attention maps or features (Chefer et al., 2021a; Zhao et al., 2024). However, most attribution techniques are computationally intensive, requiring, *e.g.*, heavy backward passes to compute gradients. The second family maps intermediate tokens directly into the model's output space. Logit Lens (nostalgebraist, 2020) popularized this idea, decoding intermediate LLM hidden states with the final unembedding matrix. Tuned Lens (Belrose et al., 2023) refined this by training per-layer translators to correct depth-wise affine drift. Given its

simplicity and gradient-free nature, zero-shot token-decoding became popular for LLM explainability (Li et al., 2024b; Pal et al., 2023; Patil et al., 2024; Prakash & Lee, 2023; Wang, 2025; Wendler et al., 2024; Wu et al., 2026) and was quickly adopted for ViTs by applying the classification head to decode patch tokens (Joseph et al., 2025; Vilas et al., 2023). Since then, it has been extended to multimodal language models (Jiang et al., 2025; Neo et al., 2025), video models (Kim et al., 2026; Zhang et al., 2025), and audio models (Glazer et al., 2026; Yang et al., 2025). Yet, unlike LLM decoders where unembeddings naturally decode individual tokens, ViT encoder heads are trained only on the class-token. Despite this divergence, zero-shot patch decoding is assumed to reliably yield aligned maps. Here, we show this is highly model-dependent and not guaranteed.

Patch misalignment in CLIP. Several previous works observed that CLIP's token decoding can be misleading. Li et al. (2023) identified an "opposite visualization" effect where patch similarity maps highlight the background complement. Similar negative contrast was subsequently reported in open-vocabulary segmentation and dense inference (Bousselham et al., 2024; Lan et al., 2024; Shao et al., 2024; Wang et al., 2024; Zhou et al., 2022), and analyzed geometrically by Kang et al. (2025). However, since prior art focused on downstream dense prediction, they largely approached this from an interventionist perspective, developing test-time remedies like attention surgery or feature pruning to force spatial alignment. Rather than treating inversion as an isolated engineering flaw to be patched, we study it through the lens of representation underdetermination and ask whether alignment should be expected in the first place. By systematically analyzing contrastive models across architectures, objectives, and scales, we show that inverted-explainability is a pervasive implicit bias of zero-shot patch decoding in VLMs, which can be shaped by the data in controlled settings.

3 Explainability of Vision Transformers

3.1 Definitions

Consider an image I of an object located inside a spatial mask $\mathcal{M}$. The image is given as input to a ViT with N patch-tokens and one class-token. Let $\{T[I]_i\}_{i=1}^{N}$ denote the patch tokens in some layer within the ViT. A common approach to interpreting these tokens is to treat each of them as if it were the class token at the final layer. Crucially, this zero-shot approach examines native representations, rather than training auxiliary probes. For example, in ViTs for classification, we can apply to each of these tokens the classification head, which is normally applied only to the class-token of the last layer. We can then inspect the logit value corresponding to the true object class, $\mathcal{C}$, and treat it as the patch's score $\mathcal{S}$. Similarly, in image encoders of contrastive VLMs like CLIP, we can apply the projection head to each patch token and compute its cosine similarity with the text embedding of a template sentence of the form "`a photo of a` $\mathcal{C}$" obtained from the text encoder. This provides a score $\mathcal{S}$ for each patch. The map of scores obtained for all patches in a given layer is often assumed to visualize how the model interprets each spatial location of the image, as shown in Fig. 1.

To quantitatively measure the quality of an explainability map for image I, we calculate the average score of the object's tokens, $\mathcal{S}_{\text{obj}}$, and the average score of the background tokens, $\mathcal{S}_{\text{bg}}$, as

$$\mathcal{S}_{\text{obj}}[I] = \frac{1}{|\mathcal{M}|}\sum_{j\in\mathcal{M}}\mathcal{S}(T[I]_j), \quad \mathcal{S}_{\text{bg}}[I] = \frac{1}{|\mathcal{M}^c|}\sum_{j\notin\mathcal{M}}\mathcal{S}(T[I]_j), \tag{1}$$

where $\mathcal{M}^c$ is the complement of $\mathcal{M}$. The relation between these two mean scores can be used to define the explainability behavior for that image. The common interpretation is that if $\mathcal{S}_{\text{obj}} > \mathcal{S}_{\text{bg}}$ (green triangle in Fig. 1) then the object patches better correlate with the object's label than the background tokens, meaning the object tokens are directly aligned with the expected classification. However, if $\mathcal{S}_{\text{bg}} > \mathcal{S}_{\text{obj}}$ (blue triangle in Fig. 1), the background tokens have higher scores than the object tokens, which according to the common interpretation means the model's knowledge of the class is localized in opposition to the object's tokens. Lastly, if $\mathcal{S}_{\text{bg}} \approx \mathcal{S}_{\text{obj}}$ (purple band in Fig. 1), the token-decoding approach does not separate between background and object tokens. We term these three regimes *aligned-explainability*, *inverted-explainability*, and *unexplainability*, respectively.

The above definitions apply to a single image. To obtain a global measure of a model's explainability, we can examine its behavior over a large set of input images. Specifically, we propose to measure the fraction $\mathcal{A}$ of images within a large set $\mathcal{I}$, which fall in the aligned-explainability regime, *i.e.*,

$$\mathcal{A} = \frac{|\{I \in \mathcal{I} : \mathcal{S}_{\text{obj}}[I] > \mathcal{S}_{\text{bg}}[I]\}|}{|\mathcal{I}|}. \tag{2}$$

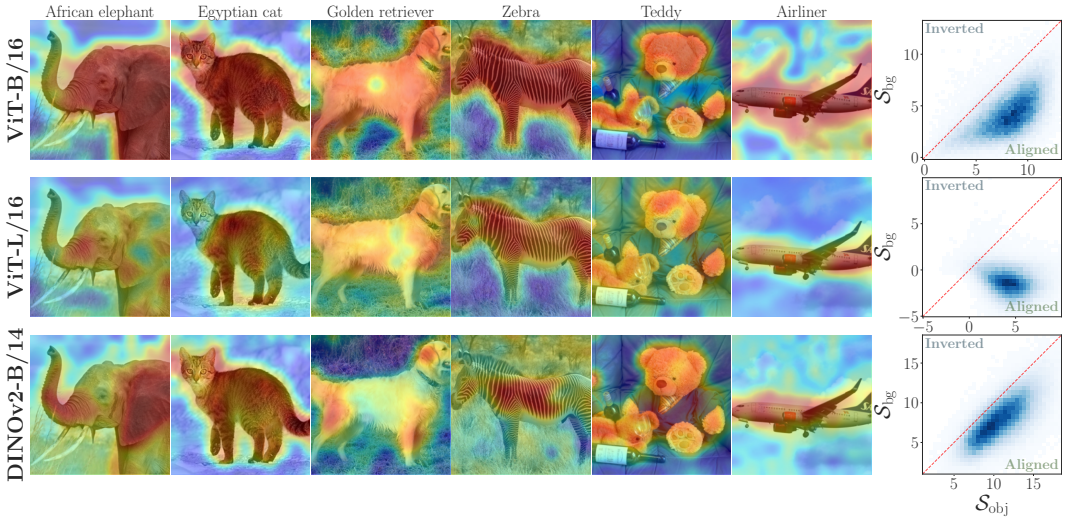


Figure 2: **Aligned-explainability bias of real-world ViT classifiers.** 2D histograms of $\mathcal{S}_{bg}$ and $\mathcal{S}_{obj}$ scores computed over all of PartImageNet++ (right) and explainability maps of six image inputs (left), on ViT-B/16, ViT-L/16, and DINOv2. All classifiers exhibit an aligned-explainability bias.

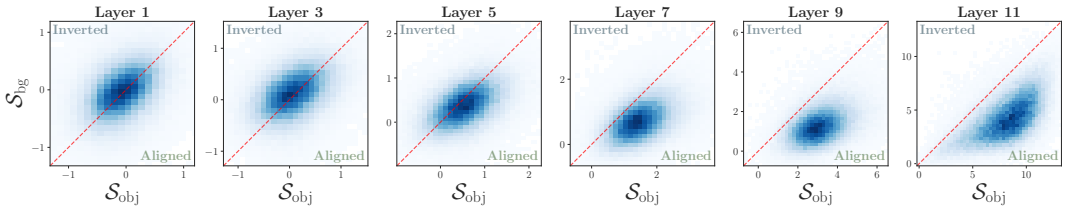


Figure 3: **Evolution of aligned-explainability in ViT-B/16.** 2D histograms of $\mathcal{S}_{bg}$ and $\mathcal{S}_{obj}$ scores computed over all of PartImageNet++ as they evolve through the model's 12 layers. While the early layers are unexplainable, as the features evolve they are more aligned.

This metric captures the bias of a model toward aligned explainability over the set $\mathcal{I}$. We consider a model aligned (◢) if $66.7\% \leq \mathcal{A}$, unexplainable (⬬) if $33.3\% \leq \mathcal{A} < 66.7\%$, and inverted (◤) if $\mathcal{A} < 33.3\%$, making for equal regimes.

Explainability analysis can be applied to any layer. However, for conciseness, all experiments use the following convention unless specified otherwise. For models in which the patch tokens of the last layer were not affected by the loss during training, we always use the layer before the last. This includes, *e.g.*, most ViT classifiers, in which the classification head operates only on the class token at the last layer, so that the patch tokens of that layer are not affected by the loss during training. For models in which the last layer's patch tokens are affected by the loss, we always show the last layer. This includes, *e.g.*, DINOv2, whose pretrained classification head expects as input the class token concatenated to the mean of the last layer's patch tokens. Further details are in Apps. A and B.3.

3.2 Aligned-explainability of real-world vision transformers

It has been well known that the feature maps of the last layers in convolutional neural network based classifiers align with the model's prediction (Selvaraju et al., 2017; Zhou et al., 2016). Therefore, it is natural to assume that the same alignment transfers to transformer-based classifiers.

Figure 2 shows the token-decoding explainability behavior of ViT-B/16, ViT-L/14 (Dosovitskiy et al., 2021), and DINOv2 (Oquab et al., 2024). We plot explainability maps for six varied image inputs, as well as 2D-histograms of $\mathcal{S}_{obj}$ and $\mathcal{S}_{bg}$ calculated over all of PartImageNet++ (Li et al., 2024a). As can be seen, all models are biased toward the aligned-explainability regime. This is true even for DINOv2, which was initially trained in a self-supervised manner and only later

Table 1: **Explainability of real-world ViT classifiers and contrastive VLMs.** Scores are computed over all of PartImageNet++. Explainability regimes are marked according to Sec. 3.1, where dashed rules mark the boundary between them. Contrastive VLMs exhibit a clear bias toward inverted-explainability, whereas classifiers are largely aligned-explainable. OpenCLIP's parentheses indicate dataset size and number of samples seen, and "+reg" marks models trained with register tokens.

Classifier	$\mathcal{A}$ (%)	Exp.
ViT-B/16 (in1k)	98.7	◢
ViT-B/16 (MIIL)	98.6	◢
ViT-S/16	98.4	◢
ViT-B/16 (AugReg2)	98.4	◢
ViT-B/16	98.4	◢
ViT-S/16 (in1k)	98.4	◢
ViT-B/16 (orig)	98.2	◢
ViT-L/16 (orig)	98.1	◢
DINOv2-B/14+reg	97.9	◢
ViT-L/16	97.8	◢
DINOv2-s/14+reg	97.7	◢
ViT-B/32 (in1k)	97.7	◢
ViT-B/32	97.5	◢
DINOv2-g/14+reg	96.8	◢
DINOv2-s/14	96.7	◢
DINOv2-B/14	94.4	◢
DeiT-B/16	92.2	◢
DINOv2-g/14	75.6	◢
ViT-Ti/16	70.5	◢
DINOv2-L/14	63.5	◖
DINOv2-L/14+reg	41.1	◖
ViT-B/32 (SAM)	14.4	◤
ViT-B/16 (SAM)	6.4	◤
DeiT-Ti/16	4.1	◤
DeiT-S/16	1.4	◤

Contrastive VLM	$\mathcal{A}$ (%)	Exp.
SigLIP-B/32	22.5	◤
OpenCLIP-B/16 (LAION80M-3B)	13.8	◤
OpenCLIP-L/14 (LAION80M-3B)	12.1	◤
OpenCLIP-B/16 (LAION400M-3B)	11.5	◤
OpenCLIP-B/16 (LAION80M-13B)	8.5	◤
OpenCLIP-L/14 (LAION80M-34B)	7.0	◤
OpenCLIP-B/16 (LAION80M-34B)	7.0	◤
OpenCLIP-B/32 (LAION2B-34B)	6.3	◤
MetaCLIP-H/14	4.6	◤
CLIP-L/14	4.3	◤
OpenCLIP-B/16 (LAION400M-34B)	3.5	◤
OpenCLIP-B/16 (LAION2B-3B)	3.5	◤
OpenCLIP-B/16 (LAION2B-13B)	3.5	◤
OpenCLIP-L/14 (LAION2B-3B)	3.4	◤
OpenCLIP-B/16 (LAION400M-13B)	3.3	◤
MetaCLIP-L/14	3.2	◤
MetaCLIP-G/14	3.0	◤
OpenCLIP-H/14 (LAION2B-34B)	2.9	◤
OpenCLIP-L/14 (LAION2B-34B)	2.8	◤
MetaCLIP-B/32	2.6	◤
CLIP-B/32	2.3	◤
OpenCLIP-g/14 (LAION2B-34B)	2.2	◤
OpenCLIP-B/16 (LAION2B-34B)	2.2	◤
OpenCLIP-g/14 (LAION2B-13B)	2.0	◤
CLIP-B/16	2.0	◤
MetaCLIP-B/16	1.5	◤

supplemented with a simple classifying head. Notably, deeper layers are more explainable than shallower ones (see Fig. 3). Histograms of additional classifiers are in App. B.1.

However, while most ViT classifiers have a strong aligned-explainability bias, it is not universal. Table 1 compares explainability scores on PartImageNet++ over a large set of ViT classifiers, varying in architecture, training loss, data augmentation, and optimizer. Indeed, ViTs trained with sharpness-aware minimization (SAM) (Foret et al., 2021) and some DeiTs (Touvron et al., 2021) exhibit completely inverted explainability. Importantly, this inversion is strictly directional and not an activation-norm artifact, and is also unrelated to register tokens (Darcet et al., 2024) (details in Apps. A.1 and B.1). We emphasize that it is also not related to some classifier failure, as all models reach a high top-1 accuracy, with almost all exceeding 88% (See App. B.1).

A more fine-grained analysis reveals that the explainability patterns of a model can differ across classes. For example, Fig. 4 depicts the explainability histograms of DINOv2/L-14 over two distinct subsets: images of animals and images that belong to the non-animal classes. Surprisingly, animals are inverted in the penultimate layer and lean toward unexplainability in the last layer, whereas non-animals are unexplainable in the penultimate layer and aligned in the last layer.

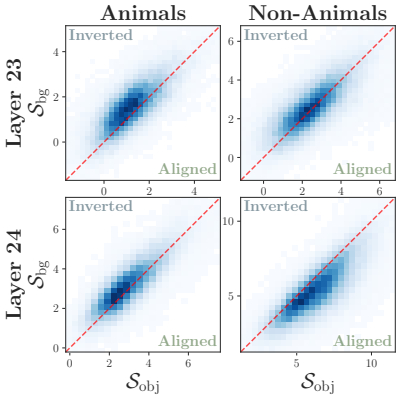


Figure 4: **Different subsets can lie in different regimes.** 2D histograms of $\mathcal{S}_{\text{bg}}$ and $\mathcal{S}_{\text{obj}}$ scores computed separately for animal and non-animal classes in PartImageNet++ on DINOv2-L/14. In the penultimate layer, animals have inverted-explainability and non-animals are unexplainable. In the last layer, animals are unexplainable and non-animals have aligned-explainability.

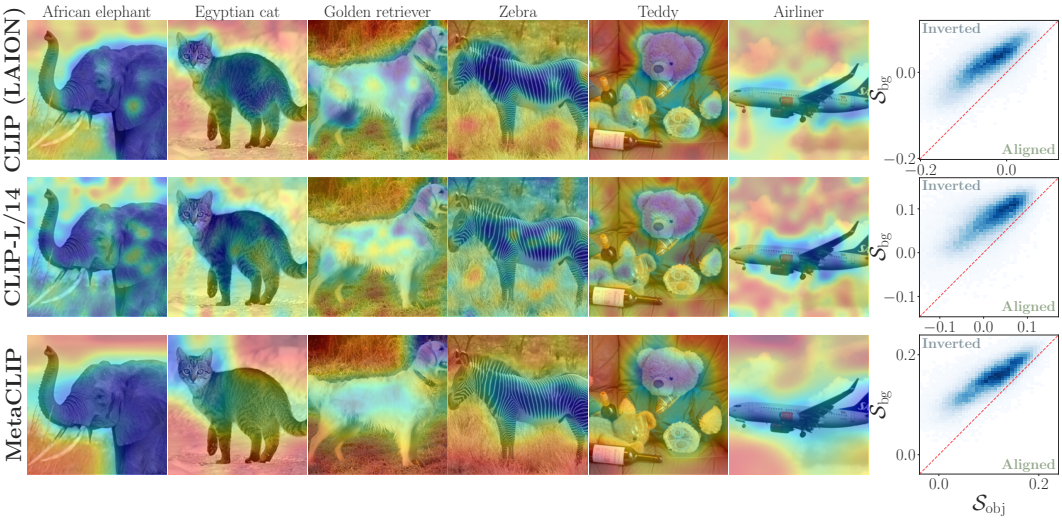


Figure 5: **Inverted-explainability bias of real-world ViT-based contrastive VLMs.** 2D histograms of $\mathcal{S}_{bg}$ and $\mathcal{S}_{obj}$ scores calculated over PartImageNet++ (right), as well as explainability maps of six image inputs (left), on three models: CLIP-B/16 trained by OpenCLIP on LAION, CLIP-L/14, and MetaCLIP. All contrastive VLMs exhibit an inverted explainability bias.

These observations suggest that different parameters of the optimization and the loss landscape bias a model toward different explainability regimes, and that this behavior is independent of its accuracy. Aligned explainability is therefore a property that cannot be assumed a-priori for ViT classifiers.

3.3 Inverted-explainability of real-world vision-language transformers

It is tempting to assume that the bias toward aligned-explainability of ViT classifiers would be seen also in contrastive VLMs like CLIP (Radford et al., 2021). However, consistent with prior localized observations (Bousselham et al., 2024; Li et al., 2023), it turns out this is not the case. Figure 5 shows the explainability behavior of two CLIP models as well as of MetaCLIP (Xu et al., 2024). In all three models the patch tokens of the object consistently score lower than those of the background, *i.e.*, the models display inverted-explainability.

As seen in Tab. 1, contrastive VLMs are consistently biased toward inverted-explainability. A natural suspicion is that the inversion originates in the contrastive objective itself, whose softmax ties every image to every caption in the batch through a single shared normalization. To examine this, we analyze a SigLIP (Zhai et al., 2023) model, which opts for a sigmoid loss applied independently to each image-caption pair. Since the original pre-trained SigLIP image encoders do not employ a class token, we train a class-token ViT-B/32 SigLIP on LAION-400M (Schuhmann et al., 2021) (see App. A.2 for the full setup). Nonetheless, this SigLIP exhibits inverted-explainability as well. We include in App. B.2 explainability histograms of additional contrastive VLMs, as well as analyses on additional data scales and training budgets of OpenCLIP models (Cherti et al., 2023). All these models lie in the inverted-explainability regime, even at the largest training scales. Similarly to the the case in ViT classifiers, the explainability evolves throughout the layers (see Fig. S3).

Several previous works have noted related phenomena in CLIP. However, this was usually taken as a flaw of the model in need of "fixing", by realigning the final attention block (Bousselham et al., 2024; Li et al., 2023; Shao et al., 2024; Wang et al., 2024; Zhou et al., 2022). For example, Li et al. (2023) noticed inverted similarity map visualizations, and proposed this occurs due to self-attention relations between inconsistent semantic regions. They concluded that this is a negative phenomenon that arises due to a lack of local supervision on the features. Other papers followed a similar logic. In particular, all these works inherently assume that the representations of contrastive VLMs *should* have aligned-explainability. Yet, there is no apparent reason to assume these representations should exhibit any explainability at all. Perhaps the representations do not require "fixing", and their inverted explainability is merely an artifact of the task over which the model is trained.

4 CONTROLLED EXPLAINABILITY ANALYSIS

In Secs. 3.2 and 3.3 we demonstrated that real-world ViT classifiers show a bias toward aligned-explainability, whereas contrastive VLMs exhibit inverted-explainability. To test whether it is indeed the training task that ultimately governs the explainability regimes of ViTs, we conduct a large-scale analysis spanning different design choices over synthetic datasets. Specifically, we construct two types of synthetic datasets, fit for classification and for contrastive learning, and train ViTs with varying dataset configurations. We find that for both classifiers and contrastive VLMs, the training can converge to either aligned or inverted explainability. This demonstrates that beyond the loss, the data distribution also affects the explainability of ViTs. However, the effect is not deterministic. We show that for a fixed task and data distribution, different training seeds can converge to different explainability regimes. This shows that the training loss and the data distribution do not impose a strict regime, but rather promote different explainability biases.

4.1 SETTING

Data construction. We construct both datasets to model the compositional nature of real-world images, requiring a ViT to localize and semantically understand a distinct object embedded within a background. Accordingly, both datasets consist of $D \times D \times 3$ sized images, where a small, randomly sampled patch $\mathbf{x}$ of size $p \times p \times 3$, representing the object, is implanted at a uniformly random spatial location within a background of random noise $\mathbf{n}$. For a supervised dataset with $\mathcal{C}$ classes, each object patch $\mathbf{x}_c$ of class c is sampled from a random multivariate Gaussian mixture model with $\mathcal{C}$ equal-weight Gaussians. Gaussians have distinct means, $\boldsymbol{\mu}_c \in \mathbb{R}^{3p^2}$, but share the same r-rank covariance matrix, $\boldsymbol{A}\boldsymbol{A}^\top$, where $\boldsymbol{A} \in \mathbb{R}^{3p^2 \times r}$. Thus, for the cth Gaussian, the data is generated as

$$\mathbf{x}_c = \boldsymbol{\mu}_c + \gamma\, \boldsymbol{A}\mathbf{z} + \boldsymbol{\epsilon}_{\mathbf{x}}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_r), \quad \boldsymbol{\epsilon}_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{3p^2}), \tag{3}$$

where γ serves as a hyper-parameter that controls the scale of the covariance matrix. Contrastive training requires more delicate modeling to simulate paired multi-modal data (*e.g.*, image-text pairs). Instead of a label, the visual object patch $\mathbf{x}$ and its continuous paired representation $\mathbf{y} \in \mathbb{R}^d$ are jointly sampled from a shared latent variable $\mathbf{w}$, and multiplied by scalar factors $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}$ as

$$\mathbf{x} = \alpha_{\mathbf{x}}\mathbf{w} + \boldsymbol{\epsilon}_{\mathbf{x}}, \quad \mathbf{y} = \alpha_{\mathbf{y}}\boldsymbol{V}\mathbf{w} + \boldsymbol{\epsilon}_{\mathbf{y}}, \qquad \mathbf{w}, \boldsymbol{\epsilon}_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{3p^2}), \quad \boldsymbol{\epsilon}_{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d). \tag{4}$$

Here, $\boldsymbol{V} \in \mathbb{R}^{d \times 3p^2}$ has orthonormal rows. In both datasets, $\mathbf{n}$ is sampled from an isotropic Gaussian distribution and scaled to control the *object to background-noise ratio* (OBNR). The OBNR is defined as $\text{OBNR} = \text{Tr}\{\text{Cov}(\mathbf{x})\}/\text{Tr}\{\text{Cov}(\mathbf{n})\}$. Specifically, when the OBNR is set to 1, the background noise variance is the most similar to the variance of the object's patch, leading to a harder separation of the object from its background. Implementation details of the background scaling appear in App. C.1. In all supervised experiments, $\boldsymbol{\mu}_c$ and $\boldsymbol{A}$ are sampled once by drawing i.i.d. standard normal entries, ensuring that different training seeds of the same dataset configuration are compared on the same data distribution. The matrix $\boldsymbol{V}$ is constructed by sampling once from the standard normal distribution and orthogonalizing the rows via QR decomposition, setting a single $\boldsymbol{V}$ for all contrastive experiments. In all experiments, we set $D = 20, p = 8$ for both datasets and $r = 38$ or $d = 16$ for supervised and contrastive training, respectively.

This construction allows us to explicitly control task difficulty by adjusting the OBNR, $\alpha_{\mathbf{x}}$, $\alpha_{\mathbf{y}}$, or γ. We sweep $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}} \in \{1, 2, 4\}$, $\mathcal{C} \in \{10, 20\}$ and $\gamma \in \{0.5, 0.75, 1\}$ at varying OBNRs, giving 81 contrastive and 66 supervised configurations, each trained under three seeds. Appendix D.2 includes additional results for a supervised configuration where $\boldsymbol{A}$ differs per class.

Architecture. In both types of experiments, we opt for small ViTs that consist of 12 layers. For the contrastive multi-modal case, we do not include tokenization and embedding matrices and instead operate on the sampled $\mathbf{y}$ directly. Additionally, we use $\mathbf{y}$ as a single token. More details on the architecture and training settings appear in App. C.2.

4.2 EXPLAINABILITY IN SYNTHETIC EXPERIMENTS

Among all hyper-parameters, we find the OBNR to have the most prominent effect on the explainability regime to which the ViTs converge. We plot in Fig. 6 the distribution of $\mathcal{A}$ across our trained models for varying OBNR values. Each model performs similarly well on its task (see App. D.3).

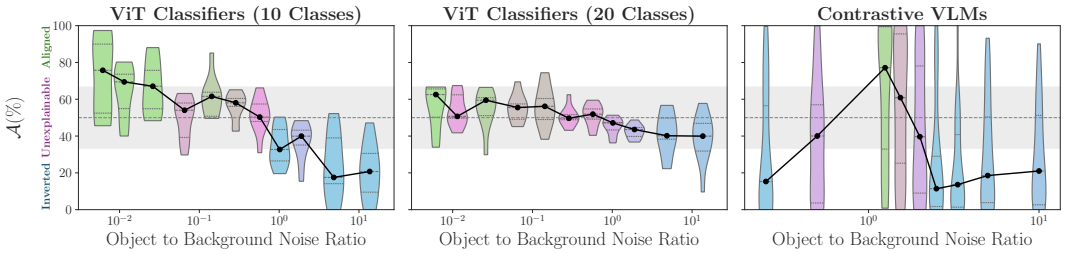


Figure 6: **Explainability trends of controlled ViT experiments.** Violin plots of $\mathcal{A}$ scores of 99 classifiers with 10 (left) or 20 (middle) classes, and 232 contrastive models (right), trained on our synthetic datasets, for different OBNR values. The violins are colored gradually according to their medians (indicated with black dots), with blue, green, and purple denoting inverted-explainability, aligned-explainability and unexplainability, respectively. The black lines connect the medians, the dashed rules mark $\mathcal{A} = 50\%$, and the gray background indicates the unexplainability regime. In ViTs, low or high OBNR values induce aligned or inverted explainability biases, respectively, while OBNR ≈ 1 promote unexplainability. In contrastive VLMs, low and high OBNR values promote inverted explainability, while OBNR ≈ 1 yields aligned-explainability. Despite these biases, ViTs occasionally converge outside the dominant regime for the same data distribution.

For ViT-classifiers, very high or very low OBNR values induce some sort of explainability: lower OBNR levels cause alignment bias, whereas higher OBNR levels tend to lead to inversion. OBNR values that are close to 1, where the object is similar in variance to the background and the classification task is harder, tend to lie in the unexplainable regime. Additionally, a smaller number of classes leads to a sharper explainability trend.

For contrastive VLMs, we see a slightly different behavior. While high OBNR values promote an inverted-explainability bias similarly to ViT classifiers, low values also induce inverted-explainability. Interestingly, aligned-explainability emerges when the OBNR is close to 1, and unexplainability emerges in intermediate OBNR values.

We note that in both cases, while models are biased to converge to some regime, as indicated by the width of the violins and their medians (connected by the black line), some can still converge outside the biased regime for the same data distribution (the violins' tail). This aligns with the spurious inverted-explainable ViTs samples in Tab. 1. In fact, different regimes can occur for the same setting using different training seeds (see App. D.1). Further experiments in App. D.2, including different data distributions and smaller ViTs, show the same trend: explainability is an implicit bias determined by the problem setting and not a fixed property of the training task or data distribution.

5 Theoretical Analysis of a Toy Model

Previous sections showed that the explainability regime changes across models, objectives, and training seeds. However, they leave a core question open: can a model be optimal yet fail to be explainable? To answer this, we study a problem simple enough to solve analytically. We derive a closed form expression for the optimal classifier, identify a toy attention architecture that precisely implements it, and show that its token-decoding map can be made aligned, unexplainable, or inverted. All three versions produce identical logits, making them indistinguishable under any logit-based loss.

The classification task. We consider a one-dimensional image $\mathbf{z} = (\mathrm{z}_1, \ldots, \mathrm{z}_N) \in \mathbb{R}^N$, a vector of length N whose entry z_n is the content of position n. One position n_{obj}, drawn uniformly at random, holds an object, and every other position holds background $\mathrm{z}_n \sim \mathcal{N}(0, \sigma^2)$ drawn independently of the class. The object belongs to a class $\mathrm{c} \in \{1, \ldots, K\}$, which equals i with prior probability π_i, and its content follows a Gaussian mixture over the K classes: given $\mathrm{c} = i$, $\mathrm{z}_{n_{\text{obj}}} \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

Proposition 1 (Optimal classifier)**.** *The optimal classifier for this data distribution is*

$$p_{\mathrm{c}|\mathbf{z}}(i|\boldsymbol{z}) = \mathrm{softmax}_i\left(\log\left(\sum_{n=1}^{N} e^{\psi_{i,n}}\right) + \beta_i\right), \qquad \psi_{i,n} = \frac{z_n^2}{2\sigma^2} - \frac{(z_n - \mu_i)^2}{2\sigma_i^2}, \tag{5}$$

where $\beta_i = \log \pi_i - \log \sigma_i$.

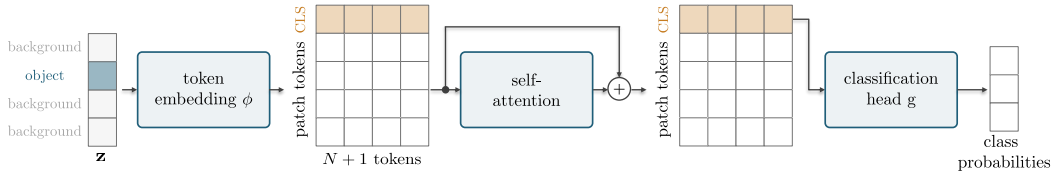


Figure 7: **The toy transformer model.** Each token in the 1D $\mathbf{z}$ image input is embedded using ϕ. After prepending a class token, a single self-attention block with a residual connection produces representations. The classification head g is applied to the class token to output predictions.

The toy model. Figure 7 shows the model's architecture: token embedding, followed by a single-head self-attention, a residual connection, and finally a classification head. It treats every entry of $\mathbf{z}$ as a token, and prepends a learnable class token, giving $N+1$ tokens in total. At a high level, every patch token carries a score for each class, and that is what a token-level explanation reads.

Theorem 1 (One optimum, three regimes)**.** *There are weights for the toy model, with a self-attention value matrix $\boldsymbol{W}_V = (N+1)\alpha\boldsymbol{I}$ for a scalar α, such that for every $\alpha > 0$ and every $\alpha \leq -1$ the class token computes the optimal classifier of Proposition 1 exactly, with the same logits for every α. For these weights and every class i, the scores $\mathcal{S}_{\text{obj}}$ and $\mathcal{S}_{\text{bg}}$ of class i satisfy:*

1. *if $\alpha > 0$, then $\mathbb{E}[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i] > 0$, so the map is aligned-explainable in expectation;*
2. *if $\alpha \leq -1$, then $\mathbb{E}[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i] < 0$, so the map is inverted-explainable in expectation;*
3. *if $|\alpha| \to \infty$, then $\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} \to 0$ in every image, so the map becomes unexplainable.*

Items 1 and 2 hold when the distribution of class i differs from the background ($\mu_i \neq 0$ or $\sigma_i \neq \sigma$).

The weights and the proofs are given in App. E. Importantly, when the same logits are produced using different weights, the training loss alone cannot tell the weight settings apart.

Theorem 1 yields two key conclusions that challenge the assumption of inherent explainability. First, it shows that an inverted token-decoding map or an unexplainable map incurs no loss penalty. Thus, these non-aligned maps do not inherently indicate a defective or suboptimal model. Second, since the same loss value can be reached with various weights, it does not dictate the convergence regime. Therefore, the token-decoding explainability bias of a model must stem from other factors, such as the architecture, the optimization, or the data distribution.

Naturally, our toy model has limitations. The setting is simplified, using one dimension, a single implanted object patch, and one attention block. It also decodes patch tokens unaffected by the loss, and requires an embedding dimension of at least $K+1$, while real ViTs are often narrower relative to the number of classes. Furthermore, while it demonstrates that an optimal model can converge to any of the three regimes, it does not explain the specific regime adopted by any real-world model. Nevertheless, we verify in App. B.3 that decoding patch tokens unaffected by the loss of real-world ViTs also exhibits the model's regime. We leave the study of narrower embeddings to future work.

6 Conclusion

We examined whether token-decoding explainability is an inherent property of ViTs, and found that it is not. Most real-world ViT classifiers are aligned-explainable, yet some are inverted while still classifying accurately. Furthermore, contrastive vision-language models are consistently inverted. We presented controlled experiments which showed that the explainability regime of a model shifts with its data and training setting. Finally, we presented a toy problem that illustrated that even an optimal transformer can be aligned, unexplainable, or inverted. While our analysis spans many real-world models, it focuses on gradient-free decoding of patch-tokens. Extending this study to gradient-based explainability techniques is an interesting direction for future work. Our synthetic models are deliberately small, and our toy model establishes that every regime is attainable at optimality without predicting which regime a given model reaches. Nevertheless, our work exposes an important observation, which is that aligned-explainability in patch-token decoding is a property to verify for each model, not one to assume.

AI USE STATEMENT

During the writing of this paper, generative AI tools (Claude 5 (Anthropic, 2026) and Gemini 3 (Google, 2025)) were mainly used to polish the text, and refine its clarity and fluency, including assisting with translation. In addition, they were used to generate drafts of some sections in the paper (specifically, sections relating to the proofs in Sec. 5 and App. E), as well as to assist in formulating our mathematical claims and in writing proofs. They also assisted when creating figures in this paper and when coding our experiments. Generative AI tools were not used for retrieval or discovery of related work, for research ideation, for generating synthetic data sets, for developing the theoretical models or conceptual frameworks of our work or providing the critical ingredients in the proofs, for designing or providing feedback on research methodology or experiments, for proposing or refining hypotheses, for implementing methods, for cleaning and reformatting datasets, for supporting qualitative and thematic data analysis, or for interpreting results. We have reviewed all AI-assisted work, including all proofs, code, and figures they helped in creating, using manual verification by two authors. We take full responsibility for the content of this work, including all text, claims, or artifacts generated with the help of generative AI.

ETHICS STATEMENT

In this work, we analyze pre-trained, real-world ViTs and demonstrate that common assumptions regarding their explainability do not hold, both theoretically and in practice. Specifically, we show that token-decoding explainability methods are not guaranteed to provide meaningful semantics, so reliance on such explainability maps to audit model behavior can be misleading. By exposing and disproving these assumptions, we aim to prevent potentially harmful reliance on flawed explainability metrics or misplaced trust in AI tools. Additionally, all our experiments rely only on publicly available data. We use PartImageNet++ for our analysis of real-world models and synthetic datasets for our controlled experiments, and to train SigLIP with a class-token we used the publicly available LAION-400M dataset. Lastly, while our theoretical model in Sec. 5 uses a shallow toy network, which is non-practical for real-world models, we hope even this relatively modest setting can serve as a basis for future work extending our results to deeper networks.

REPRODUCIBILITY STATEMENT

As explainability of ViTs helps create transparency of these models, so do we aim to have such transparency with regard to our paper. Therefore, we disclose all information needed for reproducing our results and analysis. Specifically, we include full and detailed proofs for all theoretical claims in App. E. For our real-world experiments, we use only publicly available checkpoints, except for the SigLIP model we trained on a publicly available dataset, for which we will release weights upon acceptance. For our controlled experiments, all details needed to recreate the experiments appear in Sec. 4.1 and App. C and we will release the code needed to train all models on acceptance. The supplementary material includes code that computes $\mathcal{S}_{\text{obj}}$, $\mathcal{S}_{\text{bg}}$, and $\mathcal{A}$ on PartImageNet++ for any `timm` ViT, DINOv2, or OpenCLIP model.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, 2020. 2

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In *International Conference on Machine Learning*, 2024. 2

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. XAI for transformers: Better explanations through conservative propagation. In *International Conference on Machine Learning*, 2022. 2

Anthropic. Introducing claude opus 5. https://www.anthropic.com/news/claude-opus-5, 2026. 6

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 6816–6826. IEEE, 2021. 1

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023. 1, 2

Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3828–3837. IEEE, 2024. 1, 2, 3.3, 3.3

Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 397–406, 2021a. 2

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, 2021b. 2

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023. 3.3, A.3, B.2, S6

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International conference on learning representations*, volume 2024, pp. 2632–2652, 2024. 3.2, A.1

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009. A.1

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. Jump to conclusions: Short-cutting transformers with linear transformations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 9615–9625, 2024. 1

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy. 1, 2, 3.2, S2

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM. 3.2

Neta Glazer, Yael Segal-Feldman, Hilit Segev, Aviv Shamsian, Asaf Buchnick, Gill Hetz, Ethan Fetaya, Joseph Keshet, and Aviv Navon. Beyond transcription: Mechanistic interpretability in ASR. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40(44), pp. 37407–37416, 2026. 1, 2

Yuan Gong, Yu-An Chung, and James Glass. AST: Audio spectrogram transformer. In *Proc. Interspeech 2021*, pp. 571–575, 2021. doi: 10.21437/Interspeech.2021-698. 1

Google. A new era of intelligence with gemini 3. https://blog.google/products-and-platforms/products/gemini/gemini-3/, 2025. 6

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, July 2021. URL https://doi.org/10.5281/zenodo.5143773. A.3, S3, C.2

Nick Jiang, Anish Kachinthaya, Suzie Petryk, and Yossi Gandelsman. Interpreting and editing vision-language representations to mitigate hallucinations. In *International Conference on Learning Representations*, 2025. 1, 2

Sonia Joseph, Praneet Suresh, Lorenz Hufe, Edward Stevinson, Robert Graham, Yash Vadi, Danilo Bzdok, Sebastian Lapuschkin, Lee Sharkey, and Blake Aaron Richards. Prisma: An open source toolkit for mechanistic interpretability in vision and video. *arXiv preprint arXiv:2504.19475*, 2025. 1, 2

Raphi Kang, Yue Song, Georgia Gkioxari, and Pietro Perona. Is CLIP ideal? no. can we fix it? yes! In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 1, 2

Minji Kim, Taekyung Kim, and Bohyung Han. Map the flow: Revealing hidden pathways of information in videollms. In *International Conference on Learning Representations*, volume 2026, pp. 50839–50870, 2026. 1, 2

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. C.2

Alexander Kirillov, N Usunier, N Carion, S Zagoruyko, G Synnaeve, F Massa, N Carion, F Massa, G Synnaeve, N Usunier, et al. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision, Glasgow, UK*, pp. 23–28, 2020. 1

Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. ClearCLIP: Decomposing clip representations for dense vision-language inference. In *European Conference on Computer Vision*, pp. 143–160. Springer, 2024. 2

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33(01), pp. 6706–6713, 2019. 1

Xiao Li, Yining Liu, Na Dong, Sitian Qin, and Xiaolin Hu. PartImageNet++ dataset: Scaling up part-based models for robust recognition. In *European Conference on Computer Vision*, pp. 396–414. Springer, 2024a. 3.2, S1, B.1, B.1, S4, S6, B.2, S2, S3

Yi Li, Hualiang Wang, Yiqun Duan, Jiheng Zhang, and Xiaomeng Li. A closer look at the explainability of contrastive language-image pre-training. *arXiv preprint arXiv:2304.05653*, 2023. 1, 2, 3.3, 3.3

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. Understanding and patching compositional reasoning in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9668–9688, 2024b. 1, 2

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. C.2

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *International Conference on Learning Representations*, volume 2025, pp. 57172–57189, 2025. 1, 2

nostalgebraist. Interpreting GPT: The logit lens. LessWrong, August 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. 1, 2

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification. 3.2, A.1

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 548–560, 2023. 1, 2

Vaidehi Ramesh Patil, Peter Hase, and Mohit Bansal. Can sensitive information be deleted from LLMs? objectives for defending against extraction attacks. In *International Conference on Learning Representations*, volume 2024, pp. 45497–45514, 2024. 1, 2

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023. 1

Nirmalendu Prakash and Roy Ka-Wei Lee. Layered bias: Interpreting bias in pretrained large language models. In *Proceedings of the 6th BlackboxNLP workshop: Analyzing and interpreting neural networks for NLP*, pp. 284–295, 2023. 1, 2

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021. 1, 3.3, A.3, D.3

Mansi Sakarvadia, Arham Khan, Aswathy Ajith, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. Attention lens: A tool for mechanistically interpreting the attention head information retrieval mechanism. *arXiv preprint arXiv:2310.16270*, 2023. Will appear in NeurIPS Attributing Model Behavior at Scale workshop. 1

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. In *NeurIPS Workshop on Data Centric AI*, 2021. 3.3, A.2

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017. 3.2

Tong Shao, Zhuotao Tian, Hang Zhao, and Jingyong Su. Explore the potential of CLIP for training-free open vocabulary semantic segmentation. In *European Conference on Computer Vision (ECCV)*. Springer, 2024. 2, 3.3

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021. 3.2, A.3

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

Martina G Vilas, Timothy Schaumlöffel, and Gemma Roig. Analyzing vision transformers for image classification in class embedding space. *Advances in neural information processing systems*, 36: 40030–40041, 2023. 1, 2

Feng Wang, Jieru Mei, and Alan Yuille. SCLIP: Rethinking self-attention for dense vision-language inference. In *European Conference on Computer Vision (ECCV)*, pp. 315–332. Springer, 2024. 2, 3.3

Zhenyu Wang. LogitLens4LLMs: Extending logit lens analysis to modern large language models. *arXiv preprint arXiv:2503.11667*, 2025. 1, 2

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15366–15394, 2024. 1, 2

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. A.3, S2

Songhao Wu, Zhongxin Chen, Yuxuan Liu, Heng Cui, Cong Li, and Rui Yan. Your unembedding matrix is secretly a feature lens for text embeddings. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5454–5463, 2026. 1, 2

Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *International Conference on Learning Representations*, volume 2024, pp. 47812–47831, 2024. 3.3, A.3

Chih-Kai Yang, Neo Ho, Yi-Jyun Lee, and Hung-yi Lee. AudioLens: A closer look at auditory attribute perception of large audio-language models. In *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2025. 1, 2

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019. 1

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)*, pp. 11941–11952. IEEE, 2023. 1, 3.3, A.2

Yiming Zhang, Zhuokai Zhao, Chengzhang Yu, Kun Wang, Zhendong Chu, Qiankun Li, Zihan Chen, Yang Liu, Zenghui Ding, Yining Sun, et al. CircuitProbe: Tracing visual temporal evidence flow in video language models. *arXiv preprint arXiv:2507.19420*, 2025. 1, 2

Chenyang Zhao, Kun Wang, Xingyu Zeng, Rui Zhao, and Antoni B. Chan. Gradient-based visual explanation for transformer-based CLIP. In *International Conference on Machine Learning*, pp. 61072–61091, 2024. 2

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 6877–6886. IEEE, 2021. 1

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016. 3.2

Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *European Conference on Computer Vision (ECCV)*, pp. 696–712. Springer, 2022. 2, 3.3

A TECHNICAL DETAILS OF ANALYZING REAL-WORLD MODELS

Token grid. In all experiments, since the object is positioned at pixel resolution it generally straddles token boundaries. Therefore, before computing $\mathcal{S}_{\text{obj}}$ and $\mathcal{S}_{\text{bg}}$, we assign each token to $\mathcal{M}$ if more than half of its pixels lie inside the object mask, and to $\mathcal{M}^c$ otherwise. Thus, every token contributes to exactly one of $\mathcal{S}_{\text{obj}}$ and $\mathcal{S}_{\text{bg}}$.

Applying normalization layers. In Sec. 3.1 we describe how the score $\mathcal{S}$ of a patch is computed by the token-decoding approach of passing it through the model's classification head. We note that some models are trained such that the class-token of the final layer is first normalized via some normalization layer before being passed to the classification head. In all experiments, whenever a model includes such a normalization layer, we always apply the same processing to the patch tokens before propagating them through the classification head. This is also the case for contrastive VLMs, where usually the class-token is passed through a normalization layer before being projected onto the common image-text space. Therefore, we always pass the patch tokens through the final normalization layer and only then apply the classification head.

A.1 EXPLAINABILITY ANALYSIS OF DINOV2

Classification head. DINOv2 for classification is trained in two stages. First, it is pretrained in a self-supervised manner, and later supervised training is used to train its classification head. We use this classifying head in all DINOv2 experiments.

In standard ViTs, the classification head operates on the class-token alone, and therefore to analyze the explainability of a model using token-decoding, we can reuse the classification head directly on patch tokens. However, in DINOv2, the classification head expects as input a concatenation of the class-token and the mean of all patch tokens. Therefore, to apply token-decoding on DINOv2, we replace the class-token input of the classification head with the currently analyzed patch-token, and keep the concatenated patch-tokens mean untouched.

To verify the validity of this approach, we train for each DINOv2 model variant a second classification head that operates on the class-token alone and apply token-decoding to it exactly as in standard ViTs. Specifically, we keep each pretrained DINOv2 backbone frozen and train for it a linear classification head on its normalized final-layer class-token over the ImageNet-1k training set (Deng et al., 2009). We follow the linear evaluation protocol of Oquab et al. (2024), and train for 10 epochs using SGD with momentum 0.9, no weight decay, a batch size of 1,024, and a cosine learning-rate schedule. As in their protocol, we train the head with 13 learning rates between 10^{-5} and 0.1, and keep the one with the highest top-1 accuracy on the ImageNet-1k validation set. Training images are resized to 256 pixels and center-cropped to 224, with no further augmentation.

Table S1 compares the two heads. All new heads, which accept only a class-token, are within one point of the top-1 accuracy of the official head for the corresponding model. Importantly, every DINOv2 model remains in the same explainability regime when using the new head and $\mathcal{A}$ changes by at most 5.1 points. In particular, DINOv2-L/14 and DINOv2-L/14+reg remain unexplainable. Therefore, their unexplainability does not stem from our patch-token substitution.

Register tokens. Darcet et al. (2024) have shown that ViTs can recycle background tokens as registers that store global image information. Specifically, they showed that without adding explicit training register tokens, background patches tend to have high-norm activations. Therefore, considering how inverted-explainability is affected by the relation between $\mathcal{S}_{\text{bg}}$ and $\mathcal{S}_{\text{obj}}$, one might expect all models trained with registers to exhibit aligned-explainability and all models trained without them to exhibit inverted-explainability. However, we demonstrate in Tab. 1 that models trained without registers exhibit a clear bias towards aligned-explainability, despite background patches having higher activation norms. In fact, it seems that the existence of registers does not dictate the regime of a model at all. We include in Tabs. 1 and S4 DINOv2 models trained with and without token registers over the same architectures. Indeed, the results clearly indicate that models trained with or without token register can still exhibit aligned-explainability or unexplainability, irrespective of the existence of the registers.

Table S1: **Comparing the official classification heads of DINOv2 with a class-token head.** We compare the original DINOv2 heads used in Tab. 1 ("Official") with new trained versions of these heads ("CLS-only"). Specifically, the original heads expect as input a patch-token concatenated to the mean value of all patch-tokens, and the new heads are trained on a single input of the class-token. In both, we apply the token-decoding approach of replacing the class-token with patch-tokens of the final layer. Top-1 accuracy is computed over the ImageNet-1k validation set, and $\mathcal{A}$ over PartImageNet++ at the last layer. Explainability regimes are marked according to Sec. 3.1. Both heads yield the same regime for every model.

	Top-1 (%)		$\mathcal{A}$ (%)		**Exp.**	
Classifier	Official	CLS-only	Official	CLS-only	Official	CLS-only
DINOv2-s/14	81.1	80.3	96.7	97.2	◢	◢
DINOv2-s/14+reg	80.8	80.2	97.7	97.8	◢	◢
DINOv2-B/14	84.3	83.8	94.4	95.9	◢	◢
DINOv2-B/14+reg	84.4	83.8	97.9	97.7	◢	◢
DINOv2-L/14	86.1	85.6	63.5	65.0	◢	◢
DINOv2-L/14+reg	86.5	85.9	41.1	39.2	◢	◢
DINOv2-g/14	86.6	86.1	75.6	70.5	◢	◢
DINOv2-g/14+reg	87.0	86.6	96.8	96.5	◢	◢

A.2 Training details of the class-token SigLIP

To analyze whether the inverted-explainability bias in contrastive VLMs comes from the softmax in the contrastive objective, we opted to analyze a SigLIP (Zhai et al., 2023) model, as discussed in Sec. 3.3. However, the original pre-trained SigLIP encoders do not employ a class token, which is required for token-decoding explainability methods. We therefore train OpenCLIP's ViT-B/32 image encoder, which uses a class-token, on LAION-400M (Schuhmann et al., 2021) under the sigmoid loss, and report explainability of the resulting model as SigLIP-B/32.

The architecture of this SigLIP method is thus identical to the OpenCLIP ViT-B/32 models we analyze in Sec. 3.3 and Tabs. 1 and S6, and only the loss differs. We retain the same text tower from OpenCLIP, including the tokenizer. Any difference from the CLIP models of Sec. 3.3 and Tabs. 1 and S6 is therefore attributable to the objective rather than to the architecture. We train for 32 epochs on 32 NVIDIA B200 GPUs across four nodes, with a batch size of $4{,}096$ per device. We set the learning rate to 10^{-3}, and use $2{,}000$ warm-up steps and bfloat16 mixed precision. Following Zhai et al. (2023), we also set weight decay to 10^{-4} and Adam's β_2 to 0.95, in order to suppress the loss spikes that appear at this batch size.

The resulting trained model achieves 57.45% top-1 and 82.92% top-5 zero-shot accuracy on the ImageNet-1k validation set.

A.3 Source of pretrained checkpoints

In Secs. 3.2 and 3.3 we analyze pretrained ViT-based classifiers and contrastive VLMs and report their results in Tab. 1. The DINOv2 classifiers are the official backbones with ImageNet-1k linear heads (e.g., `dinov2_vitb14_lc` and `dinov2_vitb14_reg_lc` for DINOv2-B/14 and DINOv2-B/14+reg respectively), loaded through `torch.hub`. All other ViT-based classifiers were obtained from the `timm` library (Wightman, 2019), with the identifiers listed in Tab. S2. For DeiT (Touvron et al., 2021) we use the non-distilled checkpoints, which carry no distillation token and whose only classification head operates on the class-token. For contrastive VLMs, we use the official checkpoints of CLIP (Radford et al., 2021) and MetaCLIP (Xu et al., 2024), as well as one official OpenCLIP checkpoint (Ilharco et al., 2021), all loaded through the OpenCLIP library with the identifiers listed in Tab. S3. All other OpenCLIP models are the scaling-law checkpoints of Cherti et al. (2023), for which we use the final checkpoint of each run from their Hugging Face repository.

Table S2: **`timm` checkpoints.** Names used in Tabs. 1 and S4 and the corresponding `timm` (Wightman, 2019) model identifiers (`timm` v1.0.22). †The original ViT-L/16 weights (Dosovitskiy et al., 2021) have no identifier in v1.0.22, so we use the file `jx_vit_large_p16_224-4ee7a4dc.pth` distributed by early `timm` versions.

Name	`timm` identifier
ViT-Ti/16	`vit_tiny_patch16_224.augreg_in21k_ft_in1k`
ViT-S/16	`vit_small_patch16_224.augreg_in21k_ft_in1k`
ViT-B/16	`vit_base_patch16_224.augreg_in21k_ft_in1k`
ViT-L/16	`vit_large_patch16_224.augreg_in21k_ft_in1k`
ViT-B/16 (orig)	`vit_base_patch16_224.orig_in21k_ft_in1k`
ViT-B/16 (AugReg2)	`vit_base_patch16_224.augreg2_in21k_ft_in1k`
ViT-B/16 (MIIL)	`vit_base_patch16_224_miil.in21k_ft_in1k`
ViT-L/16 (orig)†	–
ViT-S/16 (in1k)	`vit_small_patch16_224.augreg_in1k`
ViT-B/16 (in1k)	`vit_base_patch16_224.augreg_in1k`
ViT-B/16 (SAM)	`vit_base_patch16_224.sam_in1k`
ViT-B/32	`vit_base_patch32_224.augreg_in21k_ft_in1k`
ViT-B/32 (in1k)	`vit_base_patch32_224.augreg_in1k`
ViT-B/32 (SAM)	`vit_base_patch32_224.sam_in1k`
DeiT-Ti/16	`deit_tiny_patch16_224.fb_in1k`
DeiT-S/16	`deit_small_patch16_224.fb_in1k`
DeiT-B/16	`deit_base_patch16_224.fb_in1k`

Table S3: **OpenCLIP checkpoints.** Names used in Tab. 1 and the corresponding OpenCLIP (Ilharco et al., 2021) model names and pretrained tags.

Name	Model	Pretrained tag
CLIP-B/32	`ViT-B-32`	`openai`
CLIP-B/16	`ViT-B-16`	`openai`
CLIP-L/14	`ViT-L-14`	`openai`
MetaCLIP-B/32	`ViT-B-32`	`metaclip_fullcc`
MetaCLIP-B/16	`ViT-B-16`	`metaclip_fullcc`
MetaCLIP-L/14	`ViT-L-14`	`metaclip_fullcc`
MetaCLIP-H/14	`ViT-H-14`	`metaclip_fullcc`
MetaCLIP-G/14	`ViT-bigG-14`	`metaclip_fullcc`
OpenCLIP-g/14 (LAION2B-34B)	`ViT-g-14`	`laion2b_s34b_b88k`

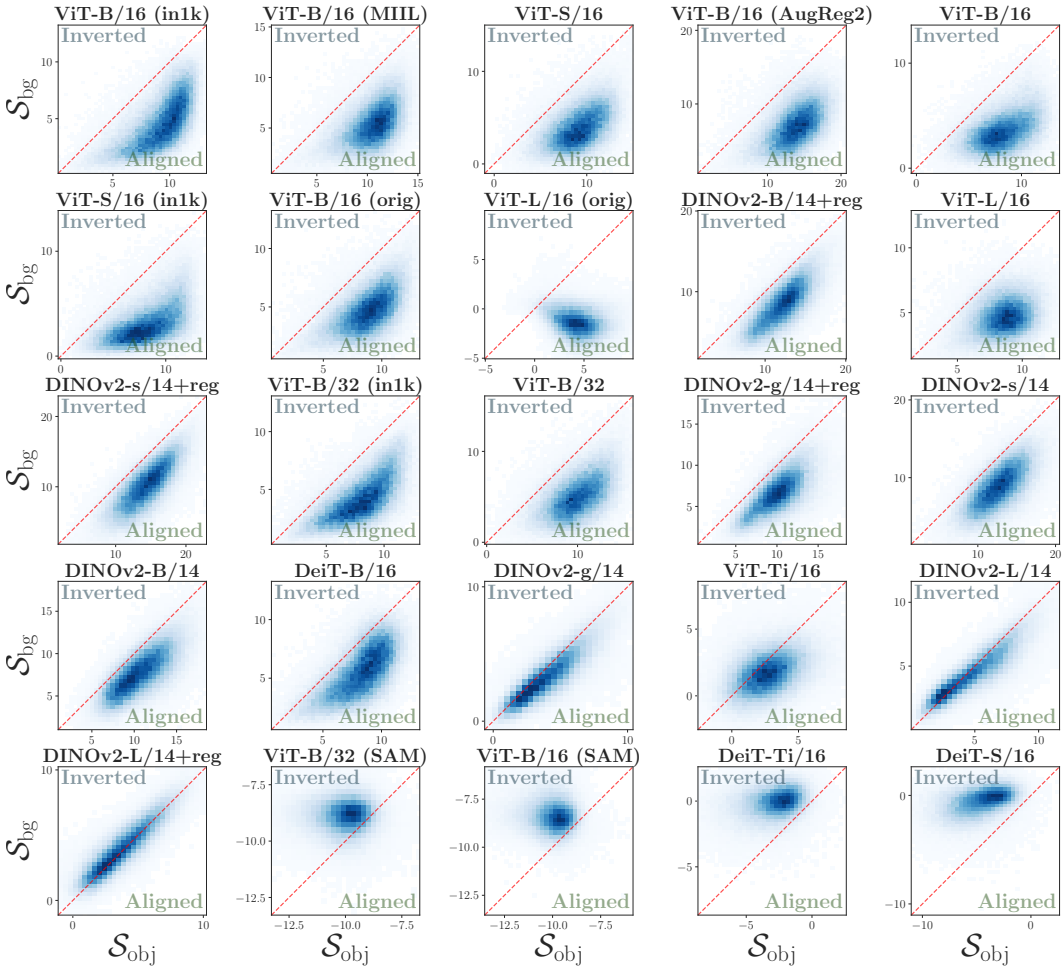


Figure S1: **Aligned-explainability bias of real-world ViT-based classifiers.** We plot 2D histograms of $\mathcal{S}_{\text{bg}}$ and $\mathcal{S}_{\text{obj}}$ scores calculated over all of PartImageNet++ (Li et al., 2024a) for 25 ViT classifiers of different architectures. ViT classifiers exhibit a strong bias toward aligned-explainability. Some models still converge to the unexplainability or even the inverted-explainability regime, irrespective of accuracy or scale.

B Additional results of real-world models

B.1 ViT classifiers

Additional 2D histograms. We include in Fig. S1 2D-histograms of $\mathcal{S}_{\text{obj}}$ and $\mathcal{S}_{\text{bg}}$ calculated over all of PartImageNet++ (Li et al., 2024a) for all ViT-based classifiers that appear in Tab. 1. As can be seen, ViTs have a strong aligned-explainability bias. Still, some models, such as DINOv2-L/14, ViT-B/16 trained with sharpness-aware minimization or some DeiT configurations exhibit non-aligned explainability.

Accuracy of ViTs. Furthermore, we include in Tab. S4 the top-1 classification accuracy over PartImageNet++ (Li et al., 2024a) of the ViTs from Tab. 1. All models have high top-1 accuracy, even in the inverted-explainability regime or the unexplainable regime.

Effect of activation norms on inverted classifiers. For contrastive VLMs, our explainability scores are computed by applying the projection matrix directly on the patch tokens (after the normalization layer, as described in App. A) and computing the cosine similarity to the embedded text prompt. However, for classifiers, after applying the classification head on the patch tokens we use

Table S4: **Top-1 accuracy of real-world ViT classifiers across explainability regimes.** Scores and accuracy values are computed over PartImageNet++ (Li et al., 2024a). Explainability regimes are marked according to Sec. 3.1. Dashed rules mark the boundary between explainability regimes. The aligned-explainability bias of ViT classifiers is unrelated to classification accuracy, as all models achieve high top-1 accuracy values.

Classifier	Top-1 Accuracy (%)	$\mathcal{A}$ (%)	Exp.
ViT-B/16 (in1k)	91.9	98.7	
ViT-B/16 (MIIL)	93.5	98.6	
ViT-S/16	88.3	98.4	
ViT-B/16 (AugReg2)	93.2	98.4	
ViT-B/16	91.2	98.4	
ViT-S/16 (in1k)	89.7	98.4	
ViT-B/16 (orig)	90.3	98.2	
ViT-L/16 (orig)	93.2	98.1	
DINOv2-B/14+reg	90.9	97.9	
ViT-L/16	93.2	97.8	
DINOv2-s/14+reg	88.1	97.7	
ViT-B/32 (in1k)	87.3	97.7	
ViT-B/32	89.1	97.5	
DINOv2-g/14+reg	93.5	96.8	
DINOv2-s/14	88.4	96.7	
DINOv2-B/14	92.1	94.4	
DeiT-B/16	95.6	92.2	
DINOv2-g/14	93.1	75.6	
ViT-Ti/16	82.2	70.5	
DINOv2-L/14	92.6	63.5	
DINOv2-L/14+reg	93.2	41.1	
ViT-B/32 (SAM)	89.1	14.4	
ViT-B/16 (SAM)	92.4	6.4	
DeiT-Ti/16	78.9	4.1	
DeiT-S/16	90.6	1.4	

the logits directly as our scores, as is standard for token-decoding explainability. Thus, one might suspect that the inverted-explainability results observed in Tab. 1 stem from activation norm irregularities. Specifically, when logits are negative, as observed in SAM-trained ViTs and DeiT-Ti/S in Fig. S1, lower $\mathcal{A}$ scores could hypothetically be caused by higher activation norms of object patches, instead of due to true directional misalignment.

To rule this out, we evaluate normalized cosine similarity scores for these checkpoints. Specifically, we compute the cosine similarity $\mathrm{w}_c^\top \boldsymbol{x}/(\|\mathrm{w}_c\|\,\|\boldsymbol{x}\|)$ between the patch tokens (after the normalization layer) and the label class weight vector, over all of PartImageNet++. This measures only the angular alignment of the classification head to the token representation. We note that here we omit the scalar bias b_c to isolate directional alignment, and that all DeiT models, as described in App. A.3, are non-distilled. The results in Tab. S5 show that all four models remain strongly inverted-explainable under cosine similarity. Particularly, the mean norms of both object and background patches are very similar in all models. Therefore, the observed explainability regime of these models is indeed directional, rather than an artifact of token norm scaling.

B.2 Contrastive VLMs

Scaling analysis. In Sec. 3.3 and App. A.2 we verified that inverted-explainability is not a simple consequence of the loss. Here, we further check whether explainability might be a product of scale. Specifically, we analyze the 29 pre-trained checkpoints of Cherti et al. (2023). These checkpoints were trained from scratch across varying training-set sizes, different numbers of samples seen during training (equivalent to different numbers of epochs), and various architecture sizes. We report the

Table S5: **Effect of activation norm on inverted ViT-based classifiers.** The mean activation norms for both object and background patch tokens are very similar, which means our $\mathcal{S}$ computation approach of directly using the logits leads to similar results as using normalized cosine similarity decoding. In particular, both exhibit the same inverted-explainability, which rules out norm-scaling artifacts.

Model	$\mathcal{S}(\cdot)$ **Approach**	**Mean** $\mathcal{S}_{\text{obj}}$	**Mean** $\mathcal{S}_{\text{bg}}$	$\mathcal{A}$ **(%)**	**Mean Norm** ($\|\boldsymbol{x}_{\text{obj}}\|/\|\boldsymbol{x}_{\text{bg}}\|$)
ViT-B/16 (SAM)	Logits	-10.07	-8.33	6.4%	$10.99/10.77$
	Cosine Similarity	-0.35	-0.30	8.4%	
ViT-B/32 (SAM)	Logits	-10.09	-8.87	14.4%	$13.22/13.16$
	Cosine Similarity	-0.33	-0.29	15.2%	
DeiT-Ti/16	Logits	-2.99	-0.13	4.1%	$25.88/24.44$
	Cosine Similarity	-0.17	-0.002	4.2%	
DeiT-S/16	Logits	-4.73	-0.57	1.4%	$23.06/21.54$
	Cosine Similarity	-0.23	-0.02	1.3%	

Table S6: **Inverted-explainability across the CLIP scaling grid.** We compute $\mathcal{A}$ scores for all models of Cherti et al. (2023) over all of PartImageNet++ (Li et al., 2024a). The models vary in architecture size and training-set size (rows) and in the number of samples seen during training (columns), which is equivalent to varying training epochs. † marks the seven runs trained at 5×10^{-4} rather than the 10^{-3} used by the other 22 runs, and with a smaller batch-size. All 29 models exhibit inverted-explainability. Moreover, for a specific architecture, larger training sets and longer training promote a stronger inverted-explainability bias.

		Samples seen		
Arch.	Data	3B	13B	34B
B-32	80M	9.7†	8.3†	9.1
	400M	9.4	5.7	6.2†
	2B	11.3	3.4†	6.3
B-16	80M	13.8	8.5	7.0
	400M	11.5	3.3†	3.5
	2B	3.5	3.5	2.2
L-14	80M	12.1	2.5	7.0
	400M	6.1	5.2	7.7
	2B	3.4	3.5	2.8
H-14	2B	–	–	2.9†
g-14	2B	–	2.0†	–

results of $\mathcal{A}$ in Tab. S6. Indeed, every one of the models exhibits inverted-explainability, with $\mathcal{A}$ ranging from 2.0% to 13.8%. More striking is the apparent direction of the inverted-explainability trend: within a specific architecture, larger training sets and longer training promote lower $\mathcal{A}$, which means a stronger bias toward inverted-explainability in the model. We note that models marked with a dagger (†) used smaller learning rate and batch sizes in their training. These runs diverge from the clear scaling trend by a small margin. For example, in B-16, the single model trained with a smaller learning rate and batch size has lower $\mathcal{A}$ than its counterparts that were trained with more seen samples, or with a larger training set. The explainability 2D histograms of all models appear in Fig. S2.

Additional 2D histograms. We include in Fig. S2 2D-histograms of $\mathcal{S}_{\text{obj}}$ and $\mathcal{S}_{\text{bg}}$ calculated over all of PartImageNet++ (Li et al., 2024a) for all contrastive VLMs that appear in Tabs. 1 and S6. As can be seen, contrastive VLMs all exhibit inverted-explainability, even at the largest training scales.

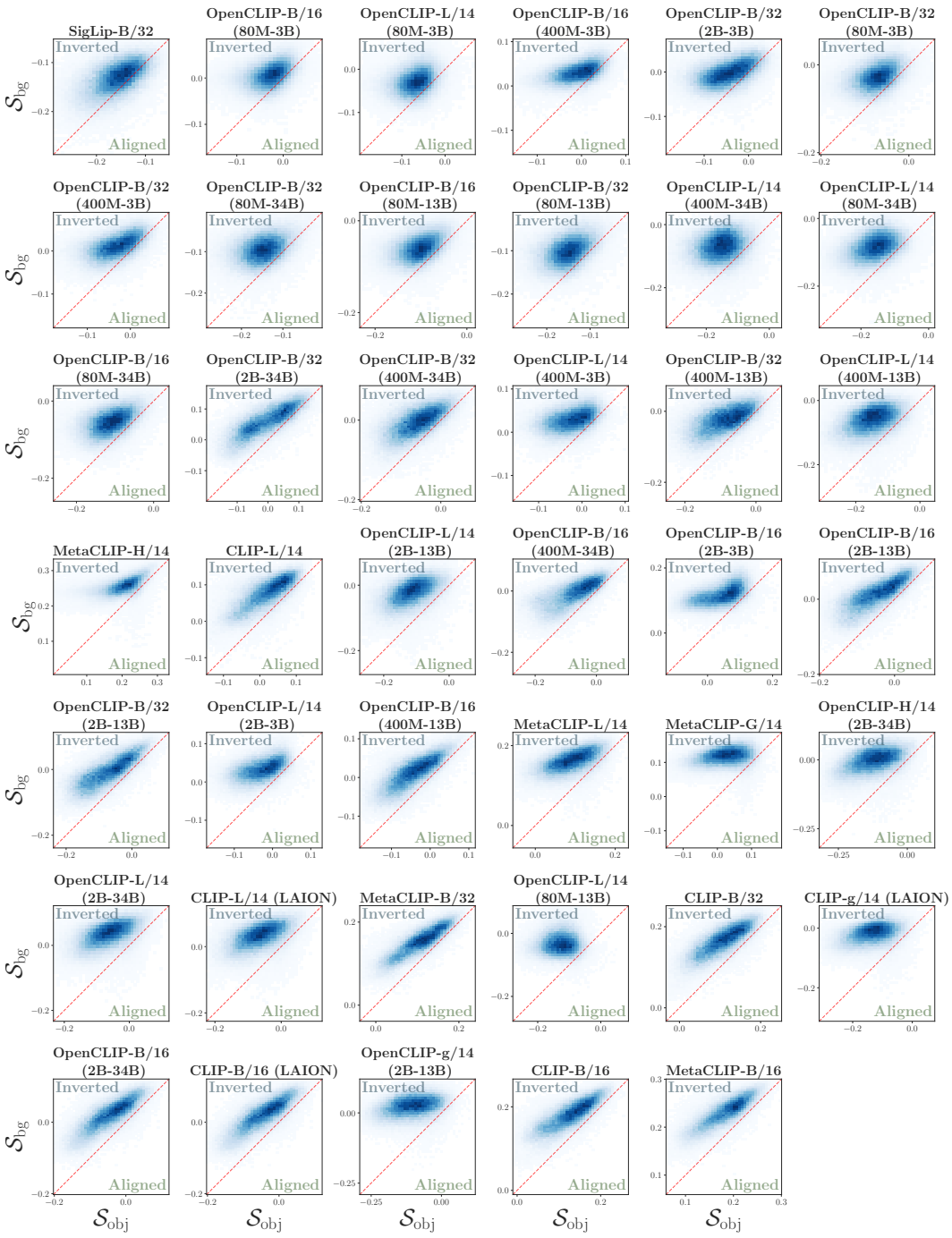


Figure S2: **Inverted-explainability bias of real-world contrastive VLMs.** We plot 2D histograms of $\mathcal{S}_{\text{bg}}$ and $\mathcal{S}_{\text{obj}}$ scores calculated over all of PartImageNet++ (Li et al., 2024a) for 41 contrastive VLMs of different architectures. Contrastive VLMs exhibit a clear bias toward inverted-explainability, irrespective of accuracy or scale.

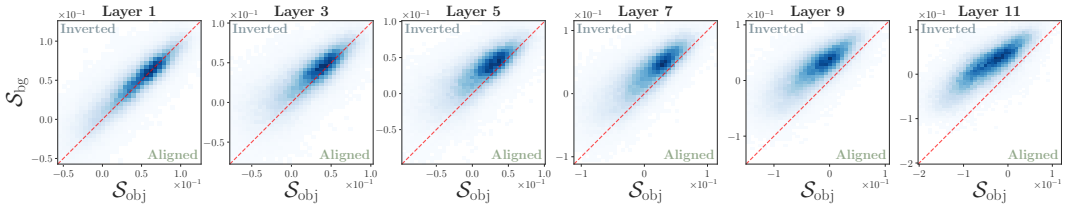


Figure S3: **Evolution of inverted explainability in CLIP.** We plot 2D histograms of $\mathcal{S}_{\text{bg}}$ and $\mathcal{S}_{\text{obj}}$ scores calculated over all of PartImageNet++ (Li et al., 2024a) as they evolve through the model's 12 layers. While the early layers are unexplainable, as the features evolve they exhibit more inversion.

Evolution of inverted explainability. We include in Fig. S3 a plot depicting how inverted explainability evolves through the layers of a CLIP model. Deeper layers are more inverted-explainable than shallower ones, similarly to the phenomenon seen in ViT classifiers in Fig. 3.

B.3 Explainability of the last layer

For models whose last-layer patch tokens are not affected by the loss, all our analyses use the penultimate layer (Sec. 3.1). In these models, the classification head or the contrastive projection, for classifiers and contrastive models respectively, usually reads only the class token of the last layer. Thus, the training loss places no constraint on the last-layer patch tokens. Specifically, the patch tokens at the end of the last layer receive no gradient from the loss. They are computed by the last layer, whose attention and MLP weights are trained only through the class token. In contrast, the patch tokens at the end of the penultimate layer are optimized directly, since the class token attends to them in the last layer. However, the decoded tokens of our toy model in Sec. 5 are of the same kind, as its head is also applied only to the class token. Therefore, for completeness, we examine here the explainability regimes of the last layer patch tokens.

Table S7 compares $\mathcal{A}$ at the penultimate and last layers for all ViT classifiers and contrastive VLMs of Tab. 1 whose last-layer patch tokens are not affected by the loss. Specifically, all models from Tab. 1 are included with the exception of DINOv2, whose head reads the last layer patch tokens and for which we already reported the last layer in Tab. 1. Indeed, the aligned-explainability and inverted-explainability exhibited in all models remain the same when examining the last layer. This indicates that the explainability regime is a property of the representation that carries through to the final representations in the last layer, even though these representations do not participate in the loss. It also supports our analysis of tokens outside the loss path in the toy model, since real-world models exhibit a clear regime on such tokens as well.

Table S7: **Explainability at the penultimate and last layers.** $\mathcal{A}$ is computed over all of PartImageNet++ at the penultimate layer (Penult.), which we use throughout the paper, and at the last layer (Last), whose patch tokens are not affected by the loss. Explainability regimes are marked according to Sec. 3.1. Each model is in the same regime at both layers. Dashed rules mark the boundary between regimes. DINOv2 is excluded, as its head reads the last-layer patch tokens, and thus our results in Tab. 1 already use its last layer decoding.

	$\mathcal{A}$ (%)		
Classifier	Penult.	Last	**Exp.**
ViT-B/16 (in1k)	98.7	98.8	◢
ViT-B/16 (MIIL)	98.6	96.3	◢
ViT-S/16	98.4	95.5	◢
ViT-B/16 (AugReg2)	98.4	88.0	◢
ViT-B/16	98.4	93.8	◢
ViT-S/16 (in1k)	98.4	98.3	◢
ViT-B/16 (orig)	98.2	92.0	◢
ViT-L/16 (orig)	98.1	97.8	◢
ViT-L/16	97.8	98.2	◢
ViT-B/32 (in1k)	97.7	97.7	◢
ViT-B/32	97.5	94.5	◢
DeiT-B/16	92.2	95.1	◢
ViT-Ti/16	70.5	70.5	◢
ViT-B/32 (SAM)	14.4	9.8	◤
ViT-B/16 (SAM)	6.4	11.2	◤
DeiT-Ti/16	4.1	4.5	◤
DeiT-S/16	1.4	2.2	◤

	$\mathcal{A}$ (%)		
Contrastive VLM	Penult.	Last	**Exp.**
SigLIP-B/32	22.5	14.6	◤
OpenCLIP-B/16 (LAION80M-3B)	13.8	16.5	◤
OpenCLIP-L/14 (LAION80M-3B)	12.1	12.2	◤
OpenCLIP-B/16 (LAION400M-3B)	11.5	13.0	◤
OpenCLIP-B/16 (LAION80M-13B)	8.5	11.0	◤
OpenCLIP-L/14 (LAION80M-34B)	7.0	11.9	◤
OpenCLIP-B/16 (LAION80M-34B)	7.0	16.8	◤
OpenCLIP-B/32 (LAION2B-34B)	6.3	12.5	◤
MetaCLIP-H/14	4.6	4.1	◤
CLIP-L/14	4.3	10.6	◤
OpenCLIP-B/16 (LAION400M-34B)	3.5	8.9	◤
OpenCLIP-B/16 (LAION2B-3B)	3.5	4.4	◤
OpenCLIP-B/16 (LAION2B-13B)	3.5	15.0	◤
OpenCLIP-L/14 (LAION2B-3B)	3.4	3.8	◤
OpenCLIP-B/16 (LAION400M-13B)	3.3	15.7	◤
MetaCLIP-L/14	3.2	2.6	◤
MetaCLIP-G/14	3.0	3.0	◤
OpenCLIP-H/14 (LAION2B-34B)	2.9	4.3	◤
OpenCLIP-L/14 (LAION2B-34B)	2.8	13.3	◤
MetaCLIP-B/32	2.6	6.4	◤
CLIP-B/32	2.3	9.9	◤
OpenCLIP-g/14 (LAION2B-34B)	2.2	3.1	◤
OpenCLIP-B/16 (LAION2B-34B)	2.2	15.2	◤
OpenCLIP-g/14 (LAION2B-13B)	2.0	1.9	◤
CLIP-B/16	2.0	14.3	◤
MetaCLIP-B/16	1.5	8.5	◤

C TECHNICAL DETAILS OF THE CONTROLLED EXPERIMENTS

C.1 DATA CONSTRUCTION DETAILS

OBNR details. As described in Sec. 4.1, the background noise sampled in both datasets is scaled to control the OBNR. Specifically, in each case we first normalize the background noise according to the covariance of the object patch. This way, the OBNR is equal to 1. Then, we can control the OBNR simply by multiplying it by constant factors, $\alpha_{\mathbf{n}}$.

For the supervised case (Eq. (3)), the initial scaling of the background noise is $\mathrm{Tr}\{\mathrm{Cov}(\mathbf{x}_c)\} = \sqrt{\gamma^2 r + 1}$, since $\boldsymbol{A}$ is a randomly sampled r-rank matrix. Then, we multiply the noise by $\alpha_{\mathbf{n}} \in [0.05, 0.1, 0.3, 0.6, 1, 2, 4, 8, 20, 50, 100]$. In the contrastive case (Eq. (4)), the background noise is similarly normalized to $\mathrm{Tr}\{\mathrm{Cov}(\mathbf{x})\} = \alpha_{\mathbf{x}}^2 + 1$. Here, we multiply by $\alpha_{\mathbf{n}} \in [0.1, 0.2, 0.3, 0.4, 0.5, 0.65, 0.8, 2, 4]$.

C.2 SYNTHETIC ARCHITECTURE DETAILS

In both experiments, we use small ViTs with 12 layers and a patch size of 2. As the input images are of size 20×20, this results in a 10×10 token grid. Additionally, the model always uses an embedding dimension of 24, with 3 heads.

ViT classifiers. All classifiers were trained using Adam (Kingma & Ba, 2015) with a learning rate of 0.02, $\beta_1 = 0.9, \beta_2 = 0.999$, no weight decay, and a cosine learning rate scheduler. Additionally, all models were trained with a batch size of 2,048 for 300 epochs. As the dataset samples a completely new sample at each training step, we set 100 training steps per epoch, which means an entire training run covers over 61 million samples.

Contrastive VLMs. All contrastive runs were trained using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 0.01, $\beta_1 = 0.9, \beta_2 = 0.98$, a weight decay of 0.025, and a cosine learning rate scheduler. Additionally, all models were trained with a batch size of 50,000 for 2,000 epochs, where 100 steps served as warm-up. The dataset samples new samples at each training step here as well, so we set 20 training steps per epoch, meaning the entire training run covers 2 billion samples. For the text tower, we use 4 layers with 2 heads, and an embedding dimension of 16. The contrastive loss, the projections, and the learned logit scale are unmodified from OpenCLIP (Ilharco et al., 2021).

D CONTROLLED EXPERIMENTS DETAIL

D.1 EXPLAINABILITY REGIME VARIANCE FOR DIFFERENT TRAINING SEEDS

Section 4 demonstrates that in our controlled experiments, the bias toward different explainability regimes is heavily dictated by the OBNR. However, we note that this is a bias and not a strict deterministic outcome. In Fig. S4 we include 2D histograms of 6 different contrastive configurations, each for 3 different seeds. While, as we showed in Fig. 6, different OBNR values lead to a bias of some explainability regime, different training seeds of the exact same data distribution can still yield very different explainability regimes. For example, for contrastive models, when the OBNR is close to 1 (*e.g.*, when $\alpha_{\mathbf{n}} = 0.8$) models have a bias toward aligned-explainability. However, when training on three different seeds, one converged to an inverted-explainable solution (Fig. S4c). In the opposite case, when the OBNR is high (*e.g.* when $\alpha_{\mathbf{n}} = 0.2$) the natural bias is toward inverted-explainability. Still, one of the training seeds for such a configuration causes the model to converge to aligned-explainability (Fig. S4b). In fact, we can even find configurations such as Fig. S4a where the three seeds lead to all three possible regimes: inverted-explainability, aligned-explainability, and unexplainability.

In addition, in Fig. S5 we include 2D histograms of 2 different ViT-based classifiers, each for 3 different seeds. Here, as well, different training seeds can lead to very different and interesting explainability patterns. For instance, Fig. S5b demonstrates a similar phenomenon as seen in DINOv2-L/14 (Fig. 4). Specifically, different class clusters can converge to have different explainability regime, similar to the animals and non-animals clusters of DINOv2. This reinforces our claim on the elusive nature of explainability. It is not some inherent property of transformers, but an outcome that is not to be assumed a-prior.

D.2 ADDITIONAL RESULTS

We conduct additional experiments on more complicated tasks for the ViTs, both in the supervised and the contrastive configurations. For ViT classifiers, we repeat our experiment of Sec. 4.1, while this time also randomly sampling the r-rank $\boldsymbol{A}$ matrix per class. As the covariances of the objects can vary widely between each class, we limit γ to $[0.5, 0.75]$, yielding a total of 44 supervised configurations, each trained under three seeds. For the self-supervised experiments, we reduce the number of layers in the ViT to 6, and test on $\alpha_{\mathbf{n}} \in [0.05, 0.1, 0.2, 0.3, 0.5, 0.65, 0.8, 4.0]$, which yields 72 configurations, each trained under three seeds.

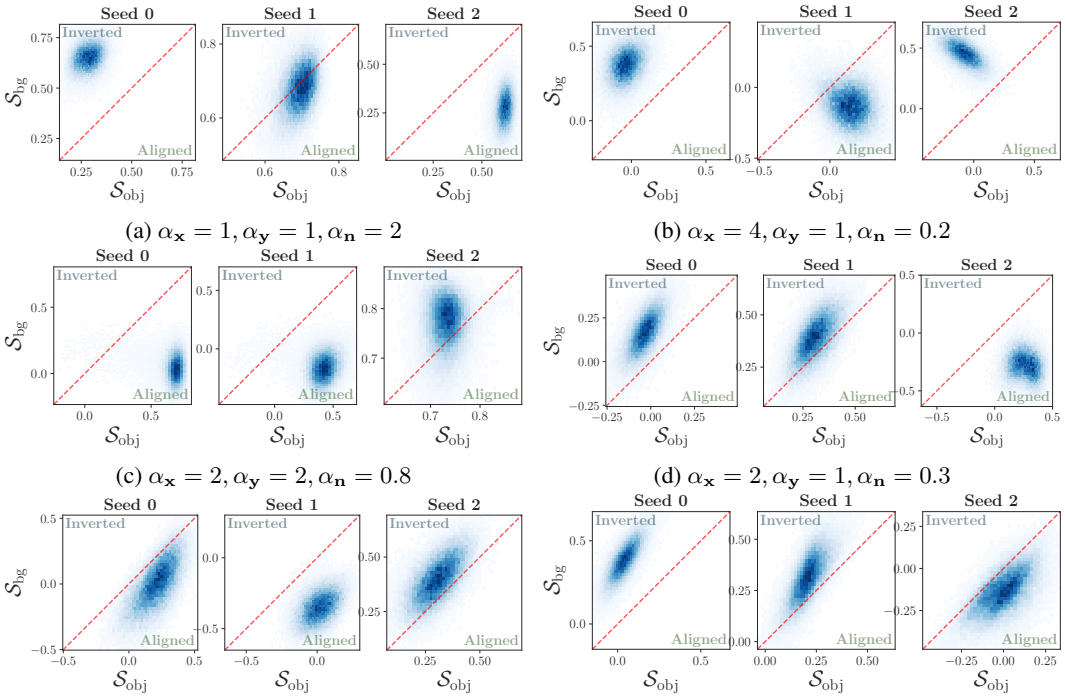
In Fig. S6 we plot the distribution of $\mathcal{A}$ across these trained models for varying OBNR values. The explainability trends are exactly as in Fig. 6. Specifically, in ViT-classifiers very high or very low OBNR values induce explainability, with higher background noise levels causing alignment bias and lower background noise levels leading to inversion, while OBNR ≈ 1 lie in the unexplainable regime. Here, as well a smaller choice classes amount leads to a sharper explainability trend.

The explainability behavior seen in Fig. 6 seen for contrastive VLMs, also remains the same for a smaller ViT. Low and high OBNR values promote an inverted-explainability bias, and OBNR ≈ 1 promotes aligned-explainability.

D.3 PERFORMANCE OF TRAINED MODELS

In Sec. 4.2 we discuss the results of our controlled experiments, and demonstrate that the bias toward different explainability regimes can be controlled in our setting via adjusting the OBNR. However, as the OBNR is also dominant in setting the complexity of the problem, it is important to validate that all compared runs converge and each perform well on its respective task. We therefore include in Figs. S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17 and S18 plots depicting the performance of all trained models shown in Fig. 6, separated according to their $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}, \gamma, \alpha_{\mathbf{n}}$ configurations.

Contrastive VLMs. For contrastive VLMs we plot the training loss and the text-to-image retrieval recall at 5 (R@5) during training of all models in Figs. S7, S8, S9 and S10. R@5 is measured on a held-out set of 10,000 pairs, where each text embedding ranks all 10,000 image embeddings by cosine similarity, and it is the fraction of texts whose paired image ranks among the top 5 (Radford et al., 2021). The image-to-text R@5 is nearly identical in all runs, so we report only the text-to-



(a) $\alpha_{\mathbf{x}} = 1, \alpha_{\mathbf{y}} = 1, \alpha_{\mathbf{n}} = 2$

(b) $\alpha_{\mathbf{x}} = 4, \alpha_{\mathbf{y}} = 1, \alpha_{\mathbf{n}} = 0.2$

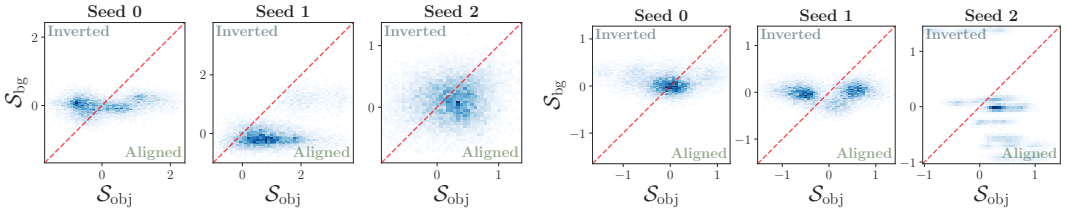
(c) $\alpha_{\mathbf{x}} = 2, \alpha_{\mathbf{y}} = 2, \alpha_{\mathbf{n}} = 0.8$

(d) $\alpha_{\mathbf{x}} = 2, \alpha_{\mathbf{y}} = 1, \alpha_{\mathbf{n}} = 0.3$

(e) $\alpha_{\mathbf{x}} = 4, \alpha_{\mathbf{y}} = 4, \alpha_{\mathbf{n}} = 0.65$

(f) $\alpha_{\mathbf{x}} = 1, \alpha_{\mathbf{y}} = 4, \alpha_{\mathbf{n}} = 0.4$

Figure S4: **Different training seeds of the exact same configuration exhibiting different explainability regimes.** 2D histograms of $\mathcal{S}_{\text{bg}}$ and $\mathcal{S}_{\text{obj}}$ scores of 6 different contrastive configurations, spanning different $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}, \alpha_{\mathbf{n}}$ (and thus different OBNRs), trained over our synthetic dataset. While all configurations induce a bias toward some explainability regime, as seen in Fig. 6, different training seeds of the exact same data distribution can still yield very different explainability regimes. For example, when $\alpha_{\mathbf{n}} = 0.8$ then OBNR ≈ 1 and the contrastive models have a bias toward aligned-explainability, yet one of the three training seeds of Fig. S4c leads to inverted-explainability. Additionally, when $\alpha_{\mathbf{n}} = 0.2$ the OBNR is high and the natural bias is toward inverted-explainability, however one of the seeds in Fig. S4b cause the model to converge to aligned-explainability. In some configurations, such as in Fig. S4a, the three seeds even led to the three possible regimes: inverted-explainability, aligned-explainability, and unexplainability. Indeed, explainability is not an inherent property of transformers, but an outcome contigent on the problem setting and optimization.



(a) $\mathcal{C} = 10, \gamma = 0.75, \alpha_{\mathbf{n}} = 100$

(b) $\mathcal{C} = 10, \gamma = 0.5, \alpha_{\mathbf{n}} = 50$

Figure S5: **Different training seeds of the exact same configuration exhibiting different explainability regimes.** 2D histograms of $\mathcal{S}_{\text{bg}}$ and $\mathcal{S}_{\text{obj}}$ scores of 2 different ViT-classifier configurations, spanning different $\mathcal{C}, \gamma, \alpha_{\mathbf{n}}$ (and thus different OBNRs), trained over our synthetic dataset. While a configuration induces a bias toward some explainability regime, as seen in Fig. 6, different training seeds of the exact same data distribution can still yield very different explainability patterns. For example, in Fig. S5a, as the OBNR is low the bias is toward aligned-explainability (Fig. 6). However, one seed exhibits at least partial inverted-explainability. Additionally, in Fig. S5b we can see some seeds exhibit different explainability for different classes of the model, similar to the phenomenon in DINOv2-L/14 (Fig. 4). Here, the model might cluster classes, and converge to have aligned or inverted explainability depending on the classes cluster, similarly to the animals and non-animals clusters of DINOv2. This reinforces how explainability is not a static property of transformers.

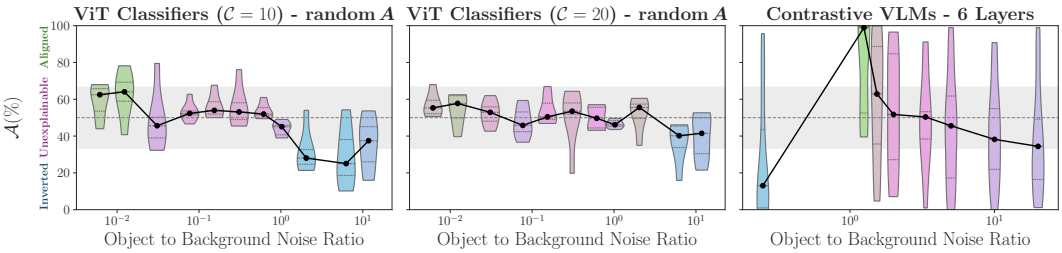


Figure S6: **Explainability trends of additional controlled ViT experiments.** Violin plots of $\mathcal{A}$ scores of 66 classifiers with 10 (left) or 20 (middle) classes, and 144 contrastive models (right), trained on our synthetic datasets, for different OBNR values. The ViT classifiers are trained on a synthetic dataset variation where $\boldsymbol{A}$ is sampled differently per class. The contrastive models are trained using ViTs with 6 layers. Both make the task harder for the ViT to accomplish. The violins are colored gradually according to their medians (indicated with black dots), with blue, green, and purple denoting inverted-explainability, aligned-explainability and unexplainability, respectively. The black lines connect the medians, the dashed rules mark $\mathcal{A} = 50\%$, and the gray background indicates the unexplainability regime. The trends of explainability biases seen in Fig. 6 are present here as well. Specifically, in ViTs, low or high OBNR values induce aligned or inverted explainability biases, respectively, while OBNR ≈ 1 promotes unexplainability. Contrastive VLMs instead converge to inverted-explainability at both extremes, and to aligned-explainability at OBNR ≈ 1. Despite these biases, ViTs occasionally converge outside the dominant regime for the same data distribution.

image direction. It is important to note that each combination of $\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}}$ and OBNR value constitutes a different data distribution of different difficulty. For instance, when the OBNR is close to 1 and $\alpha_{\mathbf{x}} = \alpha_{\mathbf{y}} = 1$, then the object patch is much less separable from the background than in the case when the OBNR is close to 10 and $\alpha_{\mathbf{x}} = \alpha_{\mathbf{y}} = 4$. This means that even an optimal model will achieve a higher training loss and lower R@5 for the more difficult case. Therefore, we separate the different configurations in our plots. As can be seen, all reported models converge and perform similarly well when trained on the same task. Therefore, the trends of explainability biases we see in Fig. 6 are indeed not a product of some failure of the model, but an implicit bias of the problem setting.

We note that we did not include in Figs. 6, S7, S8, S9 and S10 11 trained models which did not converge during training, and report only the remaining 232 runs of the 243 trained models for the 81 configurations in the paper.

ViT-based classifiers. For the ViT classifiers, we plot the training losses and the top-1 classification accuracy during training of all models in Figs. S11, S12, S15 and S16 for $\mathcal{C} = 10$ and in Figs. S13, S14, S17 and S18 for $\mathcal{C} = 20$. Here, similarly to the contrastive case, each combination of γ and OBNR value constitutes a different data distribution with different difficulty. Therefore, we separate the different configurations in our plots. As can be seen, all reported models converge, and perform similarly well when trained on the same task. In fact, almost all classifiers converge to a top-1 accuracy close to 100% when the explainability regimes are clearest, that is, when the OBNR is much larger or smaller than 1. Thus, the trends of explainability biases we see in Fig. 6 are indeed not a product of some failure of the classifier, but, here as well, an implicit bias of the problem setting.

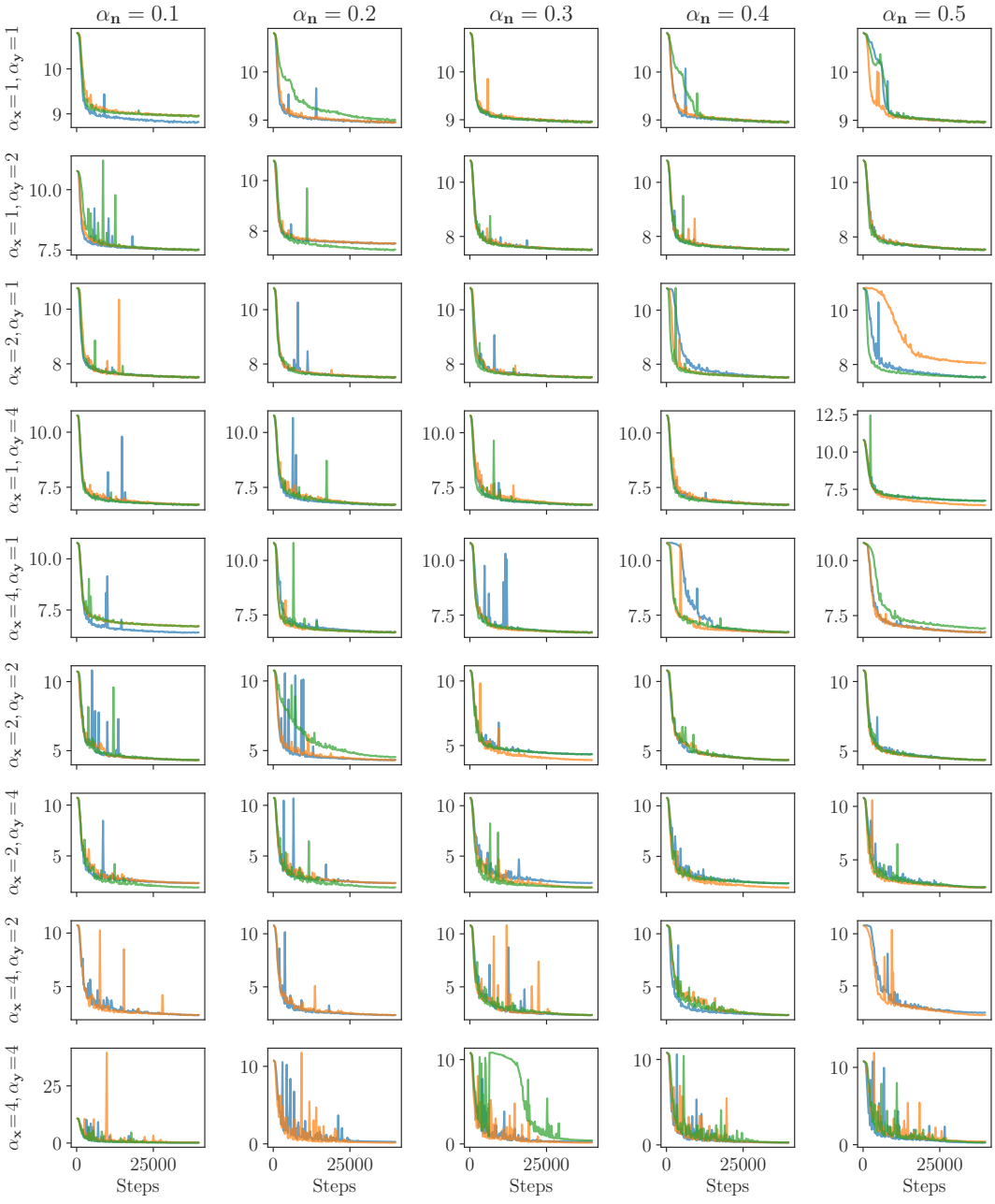


Figure S7: **Training loss plots of the contrastive VLMs from Fig. 6, for $\alpha_n \in [0.1, 0.5]$.** All reported VLMs converge similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

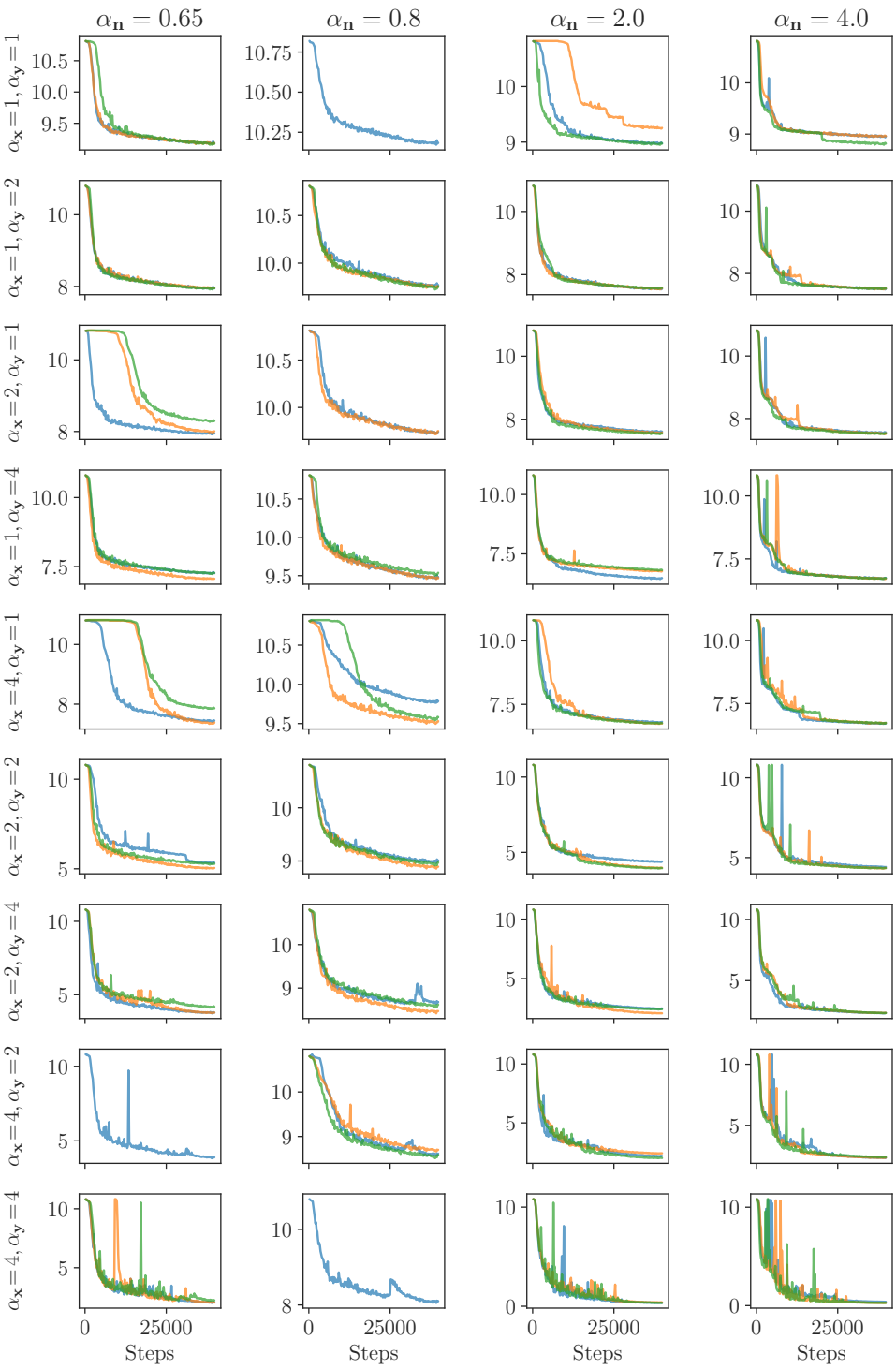


Figure S8: **Training loss plots of the contrastive VLMs from Fig. 6, for $\alpha_{\mathbf{n}} \in [0.65, 4.0]$.** All reported VLMs converge similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

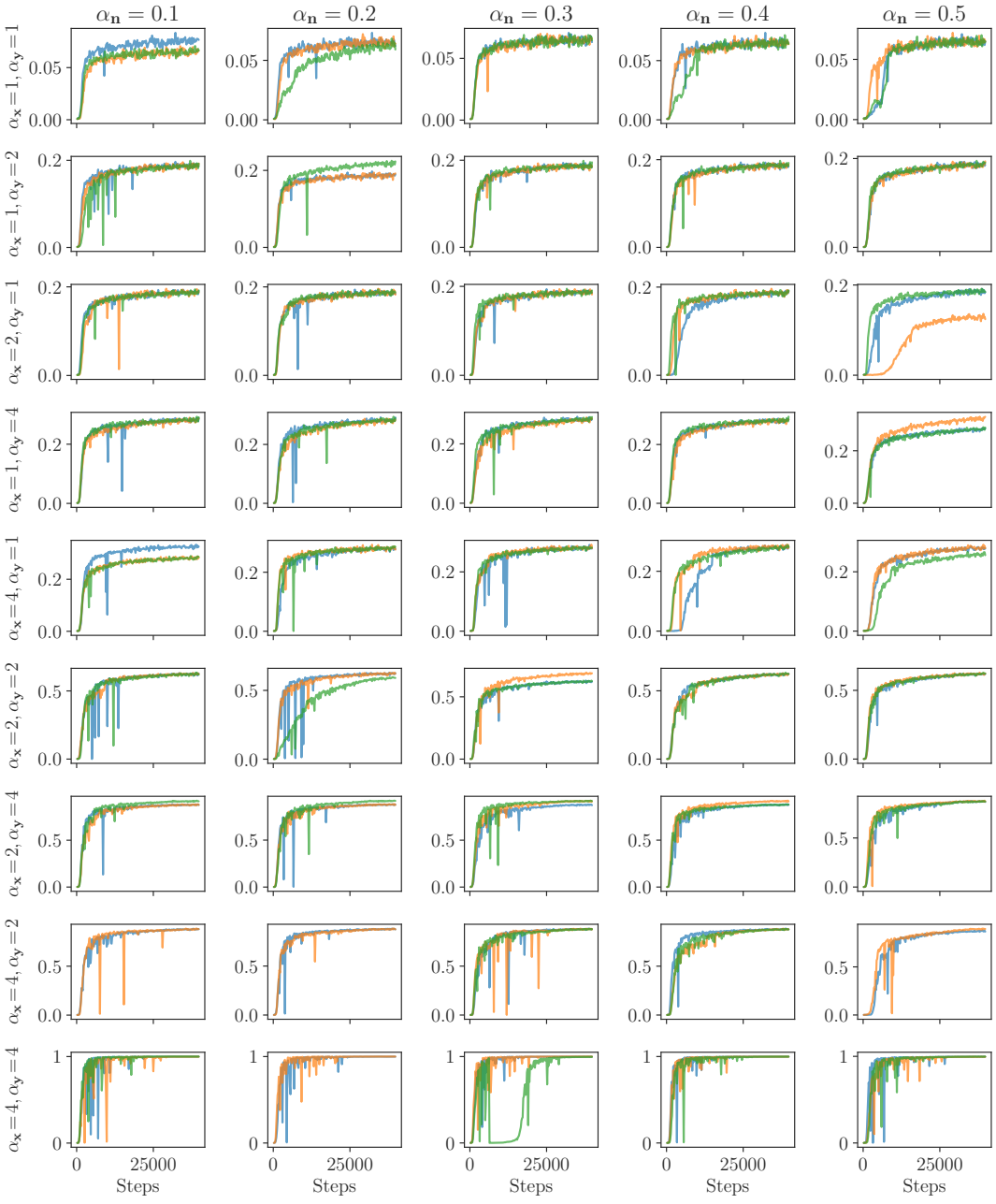


Figure S9: **Text-to-image retrieval recall at 5 (R@5) plots of the contrastive VLMs from Fig. 6, for $\alpha_{\mathbf{n}} \in [0.1, 0.5]$, during training.** All reported VLMs converge to perform similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

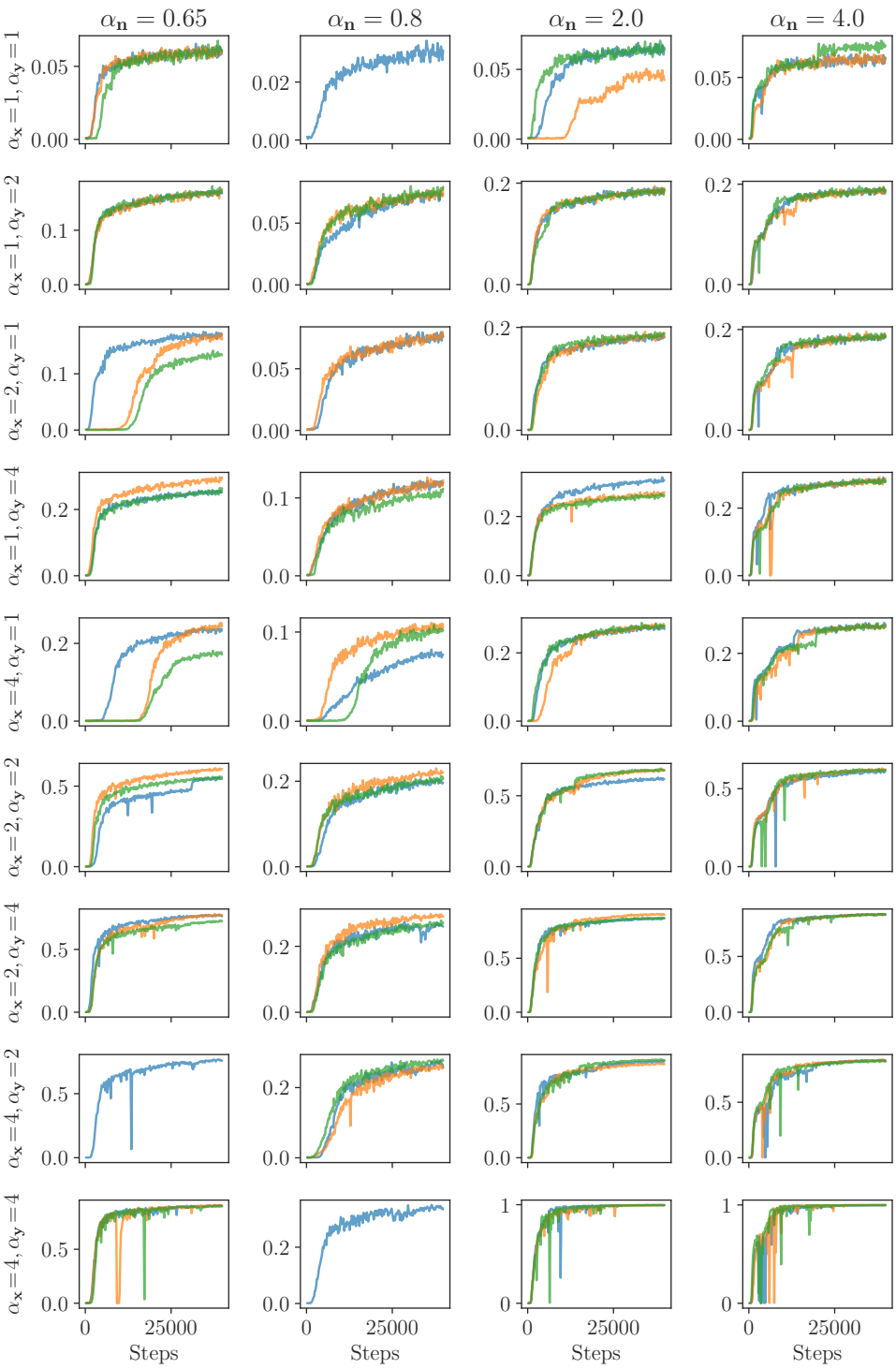


Figure S10: **Text-to-image retrieval recall at 5 (R@5) plots of the contrastive VLMs from Fig. 6, for $\alpha_n \in [0.65, 4.0]$, during training.** All reported VLMs converge to perform similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

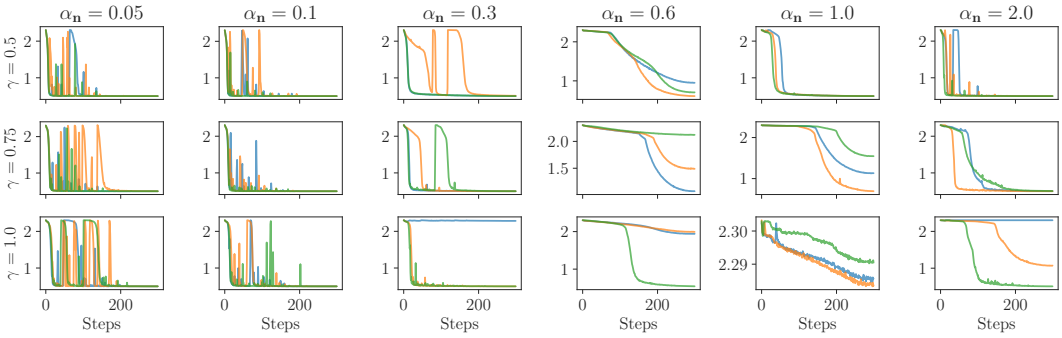


Figure S11: **Training loss plots of the ViT classifiers with** 10 **classes from Fig. 6, for** $\alpha_n \in [0.05, 2.0]$**.** All reported ViT classifiers converge similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

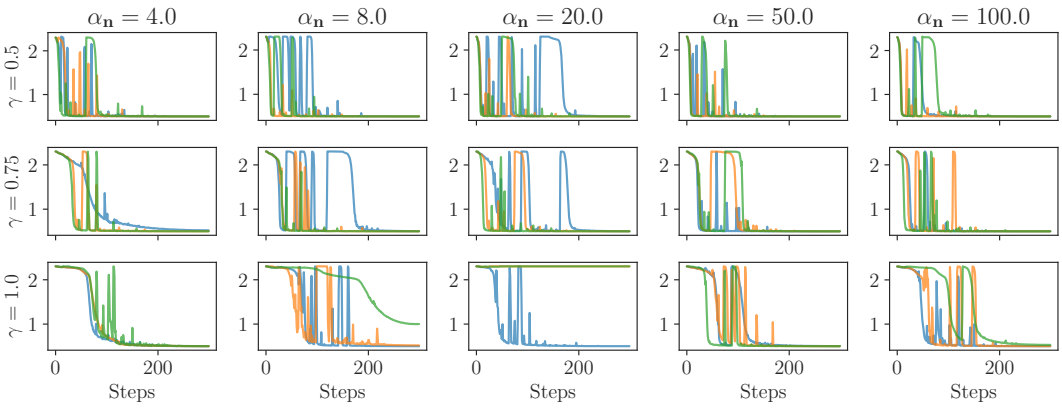


Figure S12: **Training loss plots of the ViT classifiers with** 10 **classes from Fig. 6, for** $\alpha_n \in [4, 100]$**.** All reported ViT classifiers converge similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

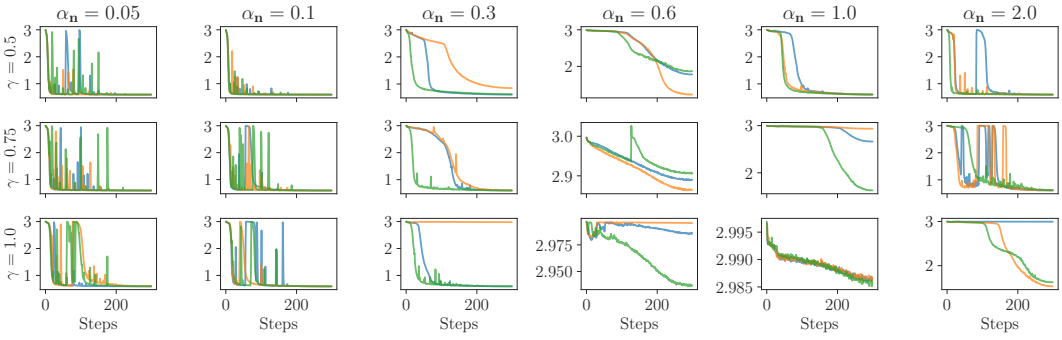


Figure S13: **Training loss plots of the ViT classifiers with** 20 **classes from Fig. 6, for** $\alpha_n \in [0.05, 2.0]$**.** All reported ViT classifiers converge similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

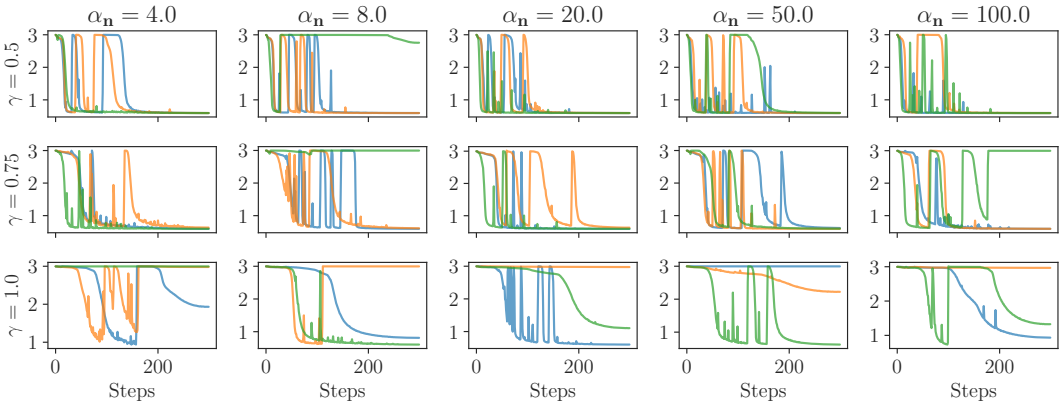


Figure S14: **Training loss plots of the ViT classifiers with** 20 **classes from Fig. 6, for** $\alpha_n \in [4, 100]$**.** All reported ViT classifiers converge similarly when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

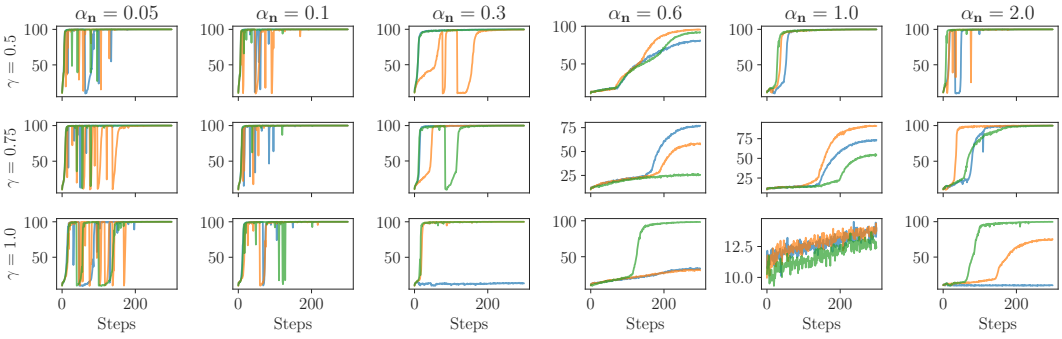


Figure S15: **Top-1 accuracy plots of the ViT classifiers with** 10 **classes from Fig. 6, for** $\alpha_n \in [0.05, 2.0]$**, during training.** All reported ViT classifiers converge similarly to high accuracy when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

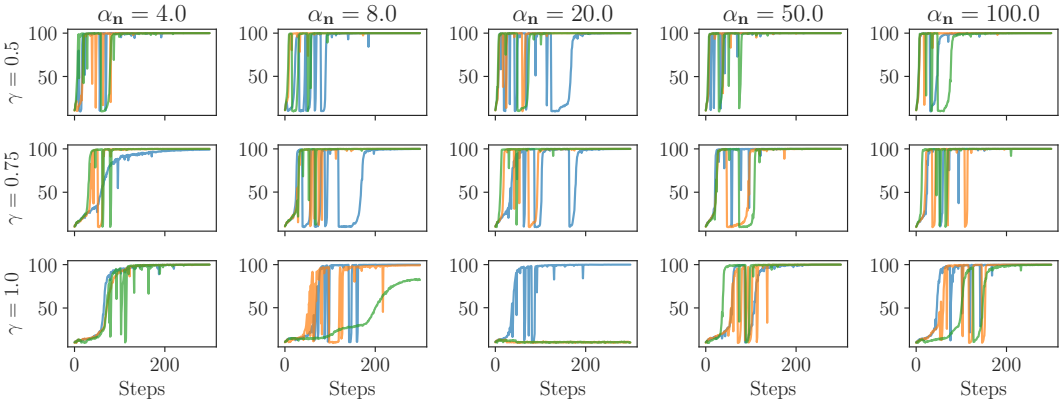


Figure S16: **Training loss plots of the ViT classifiers with** 10 **classes from Fig. 6, for** $\alpha_n \in [4, 100]$**, during training.** All reported ViT classifiers converge similarly to high accuracy when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

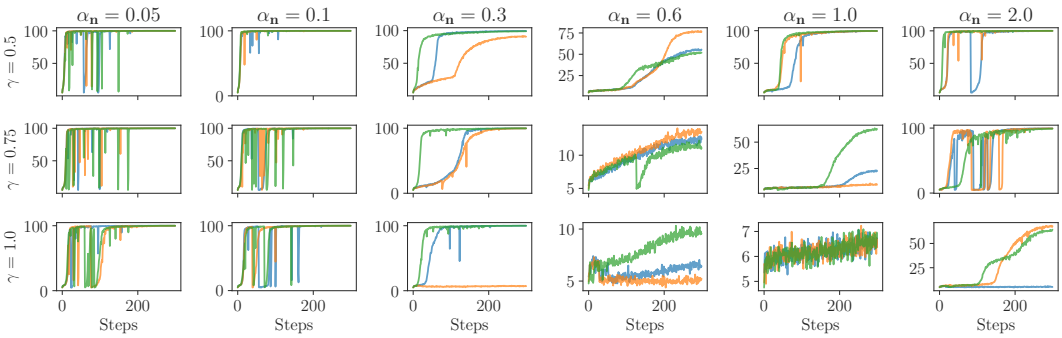


Figure S17: **Training loss plots of the ViT classifiers with** 20 **classes from Fig. 6, for** $\alpha_n \in [0.05, 2.0]$**, during training.** All reported ViT classifiers converge similarly to high accuracy when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

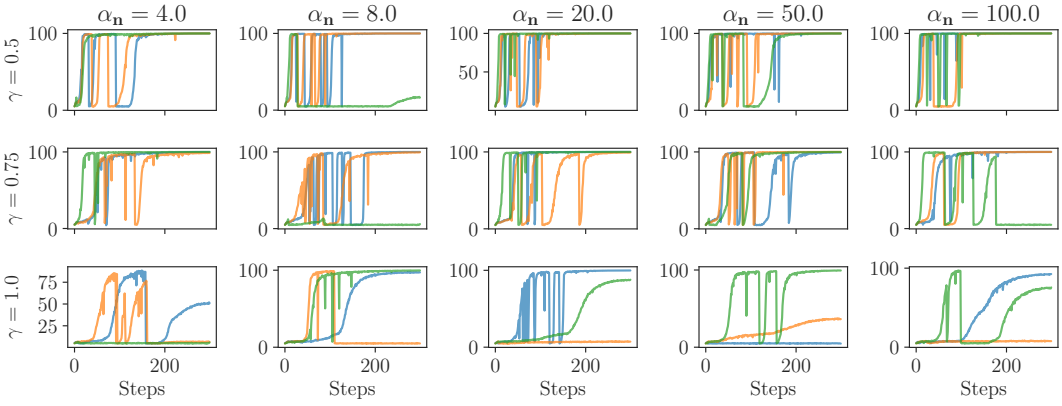


Figure S18: **Training loss plots of the ViT classifiers with** 20 **classes from Fig. 6, for** $\alpha_n \in [4, 100]$**, during training.** All reported ViT classifiers converge similarly to high accuracy when trained on the same task. Therefore, the trends of explainability biases we see are not a product of some failure of the model, but rather an implicit bias of the problem setting.

E Details on the theoretical analysis of a toy model

This appendix covers the setting of the toy model of Sec. 5 in full and provides the proofs for Proposition 1 and Theorem 1. We begin with describing the synthetic data distribution of the problem in App. E.1. Then we derive in App. E.2 the optimal classifier for this distribution in closed form, as summarized in Proposition 1. Next, we show that the optimum has the structure of a transformer forward pass and identify weights for which a single-layer transformer implements the optimal solution precisely in App. E.3, which proves the first part of Theorem 1. Finally, we provide proof for the specific α values for which the weights achieve the three regimes of Theorem 1 in Apps. E.4 and E.5.

We note that the classification head of our toy ViT applies a logarithm, whereas standard, real-world ViTs use a linear head. Therefore, we verify that the explainability regimes do not stem from this logarithm in App. E.6. Specifically, we show that dropping it leaves a linear head that changes neither the predicted class nor the regime, and costs only the exactness of the predicted probabilities obtained by applying softmax to the logits In addition, for completeness, we derive in App. E.7 a closed form for the expected difference of the scores, $\mathbb{E}[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i]$, in the aligned regime as $\alpha \to 0$.

E.1 Problem Setup

Notation. For a discrete random variable c, p denotes its probability mass function, so $p_{\mathrm{c}}(i)$ is the probability that $\mathrm{c} = i$. For a continuous random variable $\mathbf{z}$, f denotes its probability density function, so $f_{\mathbf{z}}(\boldsymbol{z})$ is the density of $\mathbf{z}$ at $\boldsymbol{z}$.

Inputs. We consider a one-dimensional image $\mathbf{z} = (\mathrm{z}_1, \ldots, \mathrm{z}_N) \in \mathbb{R}^N$, a vector of length N whose entry z_n is the content of position n. As depicted in the left part of Fig. 7, exactly one position n_{obj} holds an object, while the remaining positions in the image hold background. The object's distribution depends on which of the K classes it belongs to:

- Class: $p_{\mathrm{c}}(i) = \pi_i$, for $i = 1, \ldots, K$.
- Object position: $\mathrm{n}_{\text{obj}} \sim U(\{1, \ldots, N\})$, statistically independent of the class.
- Object content: $f_{\mathrm{z}_{n_{\text{obj}}}|\mathrm{c}}(z|i) = \mathcal{N}(z; \mu_i, \sigma_i^2)$.
- Background: $f_{\mathrm{z}_n}(z) = \mathcal{N}(z; 0, \sigma^2)$ independently for every $n \neq n_{\text{obj}}$.

The probability density of an image vector $\mathbf{z}$ conditioned on the case where an object of class i is located in the nth element is given by the Gaussian distribution:

$$f_{\mathbf{z}|\mathrm{c},\mathrm{n}_{\text{obj}}}(\boldsymbol{z}|i,n) \;=\; \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{ni}, \boldsymbol{\Sigma}_{ni}), \quad \boldsymbol{\mu}_{ni} = \mu_i \boldsymbol{e}^{(n)}, \quad \boldsymbol{\Sigma}_{ni} = \sigma^2 \boldsymbol{I} + (\sigma_i^2 - \sigma^2)\boldsymbol{e}^{(n)}\boldsymbol{e}^{(n)\top}, \tag{S1}$$

where $\boldsymbol{e}^{(n)}$ is the standard basis vector with a one in coordinate n and zeros elsewhere. The covariance is a scaled identity matrix, perturbed in the single entry that carries the object.

Here, we assume the background content is drawn independently of the class. Thus, all information about the expected classification of the image resides in the $n_{\text{obj}}{}^{\text{th}}$ element. Specifically, a token-decoding approach must highlight this exact token for an aligned-explainable solution.

E.2 Proof of Proposition 1 (The optimal classifier)

We now compute the posterior $p_{\mathrm{c}|\mathbf{z}}(i|\boldsymbol{z})$ in closed form. First, we apply Bayes' rule and the law of total probability:

$$\begin{aligned} p_{\mathrm{c}|\mathbf{z}}(i|\boldsymbol{z}) &= \frac{f_{\mathbf{z}|\mathrm{c}}(\boldsymbol{z}|i)\, p_{\mathrm{c}}(i)}{f_{\mathbf{z}}(\boldsymbol{z})} \\ &= \frac{\pi_i\, f_{\mathbf{z}|\mathrm{c}}(\boldsymbol{z}|i)}{\sum_{j=1}^{K} \pi_j\, f_{\mathbf{z}|\mathrm{c}}(\boldsymbol{z}|j)}. \end{aligned} \tag{S2}$$

Then, we use the law of total probability over the unknown object position for $f_{\mathbf{z}|\mathrm{c}}(\boldsymbol{z}|i)$:

$$\begin{aligned} f_{\mathbf{z}|\mathrm{c}}(\boldsymbol{z}|i) &= \sum_{n=1}^{N} f_{\mathbf{z}|\mathrm{c},\mathrm{n_{obj}}}(\boldsymbol{z}|i,n)\, p_{\mathrm{n_{obj}}}(n) \\ &= \frac{1}{N}\sum_{n=1}^{N} \mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ni},\boldsymbol{\Sigma}_{ni}). \end{aligned} \tag{S3}$$

The second step in the last equality uses $p_{\mathrm{n_{obj}}}(n) = 1/N$, since the object position is uniform. When substituting Eq. (S3) into Eq. (S2), the factor $1/N$ cancels and we obtain:

$$p_{\mathrm{c}|\mathbf{z}}(i|\boldsymbol{z}) \;=\; \frac{\pi_i \sum_{n=1}^{N} \mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ni},\boldsymbol{\Sigma}_{ni})}{\sum_{j=1}^{K}\pi_j \sum_{n=1}^{N} \mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{nj},\boldsymbol{\Sigma}_{nj})}. \tag{S4}$$

The expressions in the numerator and the denominator are given by the multivariate Gaussian distribution,

$$\mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ni},\boldsymbol{\Sigma}_{ni}) \;=\; \frac{1}{\sqrt{(2\pi)^N|\boldsymbol{\Sigma}_{ni}|}} \exp\!\left(-\tfrac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}_{ni})^\top \boldsymbol{\Sigma}_{ni}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_{ni})\right). \tag{S5}$$

The covariance $\boldsymbol{\Sigma}_{ni}$ is diagonal, with σ^2 in every diagonal entry except the nth, which is σ_i^2. Its inverse is therefore also diagonal, with the reciprocal of each entry,

$$\begin{aligned} \boldsymbol{\Sigma}_{ni}^{-1} &= \frac{1}{\sigma^2}\boldsymbol{I} + \frac{1}{\sigma_i^2}\boldsymbol{e}^{(n)}\boldsymbol{e}^{(n)\top} - \frac{1}{\sigma^2}\boldsymbol{e}^{(n)}\boldsymbol{e}^{(n)\top} \\ &= \frac{1}{\sigma^2}\boldsymbol{I} + \left(\frac{1}{\sigma_i^2} - \frac{1}{\sigma^2}\right)\boldsymbol{e}^{(n)}\boldsymbol{e}^{(n)\top}. \end{aligned} \tag{S6}$$

Substituting this and $\boldsymbol{\mu}_{ni} = \mu_i \boldsymbol{e}^{(n)}$ in the quadratic form in the exponent, it expands as

$$\begin{aligned} (\boldsymbol{z}-\boldsymbol{\mu}_{ni})^\top \boldsymbol{\Sigma}_{ni}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_{ni}) &= \frac{1}{\sigma^2}(\boldsymbol{z}-\mu_i\boldsymbol{e}^{(n)})^\top(\boldsymbol{z}-\mu_i\boldsymbol{e}^{(n)}) \\ &\quad + \left(\frac{1}{\sigma_i^2}-\frac{1}{\sigma^2}\right)(\boldsymbol{z}-\mu_i\boldsymbol{e}^{(n)})^\top \boldsymbol{e}^{(n)}\boldsymbol{e}^{(n)\top}(\boldsymbol{z}-\mu_i\boldsymbol{e}^{(n)}) \\ &= \frac{1}{\sigma^2}\|\boldsymbol{z}-\mu_i\boldsymbol{e}^{(n)}\|^2 + \left(\frac{1}{\sigma_i^2}-\frac{1}{\sigma^2}\right)(z_n-\mu_i)^2 \\ &= \frac{1}{\sigma^2}\left(\|\boldsymbol{z}\|^2 - 2\mu_i z_n + \mu_i^2\right) + \left(\frac{1}{\sigma_i^2}-\frac{1}{\sigma^2}\right)(z_n-\mu_i)^2 \\ &= \frac{1}{\sigma^2}\left(\|\boldsymbol{z}\|^2 - z_n^2 + (z_n-\mu_i)^2\right) + \left(\frac{1}{\sigma_i^2}-\frac{1}{\sigma^2}\right)(z_n-\mu_i)^2 \\ &= \frac{1}{\sigma^2}\left(\|\boldsymbol{z}\|^2 - z_n^2\right) + \frac{1}{\sigma_i^2}(z_n-\mu_i)^2, \end{aligned} \tag{S7}$$

where the second line uses $\boldsymbol{e}^{(n)\top}(\boldsymbol{z}-\mu_i\boldsymbol{e}^{(n)}) = z_n - \mu_i$, the fourth uses $-2\mu_i z_n + \mu_i^2 = -z_n^2 + (z_n-\mu_i)^2$, and the fifth collects the two copies of $(z_n-\mu_i)^2$.

Finally, since $\boldsymbol{\Sigma}_{ni}$ is diagonal, its determinant is the product of its diagonal entries, $\sigma^{2(N-1)}\sigma_i^2$, and only the factor σ_i depends on the class.

Substituting the determinant and Eq. (S7) into Eq. (S5) and rearranging gives

$$\begin{aligned} \mathcal{N}(\boldsymbol{z};\boldsymbol{\mu}_{ni},\boldsymbol{\Sigma}_{ni}) &= \frac{1}{\sqrt{(2\pi)^N\,\sigma^{2(N-1)}\sigma_i^2}} \exp\!\left(\frac{z_n^2 - \|\boldsymbol{z}\|^2}{2\sigma^2} - \frac{(z_n-\mu_i)^2}{2\sigma_i^2}\right) \\ &= \underbrace{\frac{e^{-\|\boldsymbol{z}\|^2/2\sigma^2}}{(2\pi)^{N/2}\,\sigma^{N-1}}}_{\text{same for every } i \text{ and } n} \cdot \frac{1}{\sigma_i} \exp\!\left(\frac{z_n^2}{2\sigma^2} - \frac{(z_n-\mu_i)^2}{2\sigma_i^2}\right). \end{aligned} \tag{S8}$$

The first multiplier does not depend on the class i or the object position n, which means when substituting Eq. (S8) back into Eq. (S4) we obtain

$$p_{\mathrm{c}|\mathbf{z}}(i|\boldsymbol{z}) = \frac{\pi_i \sum_{n=1}^{N} \frac{1}{\sigma_i} \exp\left(\frac{z_n^2}{2\sigma^2} - \frac{(z_n-\mu_i)^2}{2\sigma_i^2}\right)}{\sum_{j=1}^{K} \pi_j \sum_{n=1}^{N} \frac{1}{\sigma_j} \exp\left(\frac{z_n^2}{2\sigma^2} - \frac{(z_n-\mu_j)^2}{2\sigma_j^2}\right)}. \tag{S9}$$

The exponent term depends on the image only through z_n, and is the only factor in the expression that couples the class and the object's position. We therefore call it the *class feature* $\psi_{i,n}$, and combine the leftover $1/\sigma_i$ with the prior π_i into a single *class bias*,

$$\psi_i(z) \;=\; \frac{z^2}{2\sigma^2} - \frac{(z-\mu_i)^2}{2\sigma_i^2}, \qquad \psi_{i,n} \;\equiv\; \psi_i(z_n), \qquad \beta_i \;=\; \log \pi_i - \log \sigma_i. \tag{S10}$$

Substituting Eq. (S8) into Eq. (S9),

$$\begin{aligned} p_{\mathrm{c}|\mathbf{z}}(i|\boldsymbol{z}) &= \frac{\frac{\pi_i}{\sigma_i} \sum_{n=1}^{N} e^{\psi_{i,n}}}{\sum_{j=1}^{K} \frac{\pi_j}{\sigma_j} \sum_{n=1}^{N} e^{\psi_{j,n}}} \\ &= \frac{\exp\left(\log\left(\sum_{n=1}^{N} e^{\psi_{i,n}}\right) + \beta_i\right)}{\sum_{j=1}^{K} \exp\left(\log\left(\sum_{n=1}^{N} e^{\psi_{j,n}}\right) + \beta_j\right)} \\ &= \mathrm{softmax}_i\left(\log\left(\sum_{n=1}^{N} e^{\psi_{i,n}}\right) + \beta_i\right). \end{aligned} \tag{S11}$$

This concludes the proof of Proposition 1. This posterior is the optimal classifier: no other classifier achieves a lower cross-entropy over the data distribution, and it is what a model trained on unlimited data converges to. Any weights that make a model compute Eq. (S11) are therefore optimal, and the loss gives no reason to prefer one setting over another.

Importantly, Eq. (S11) has the structure of some forward pass of classifier. First, features are extracted from patches of an image. Then, an aggregation over the features passes through a non-linear layer. Finally, together with an added bias, a softmax over the classes produces a classification.

E.3 A SINGLE-LAYER TRANSFORMER THAT COMPUTES THE OPTIMUM (FIRST PART OF THEOREM 1)

We now show it is possible to choose weights for the toy model of Fig. 7 so that its class-token output equals the optimal classifier of Eq. (S11) exactly, for every input $\boldsymbol{z}$.

Token embedding. We consider the image patches to be already tokenized. Therefore, we only apply token embedding. Our embedding acts on each patch separately, without mixing, and raises the dimension of a scalar patch to a vector of dimension $K+1$:

$$\phi : \mathbb{R} \to \mathbb{R}^{K+1}, \qquad \phi(z) \;=\; \big(\underbrace{1}_{\text{anchor}}, \underbrace{e^{\psi_1(z)}, \ldots, e^{\psi_K(z)}}_{K \text{ class channels}}\big). \tag{S12}$$

For the learnable class token, we choose the learned weights such that the already-embedded class token is $\boldsymbol{e}^{(0)}$. Stacking it on top of the N embedded patches gives the token matrix

$$\boldsymbol{X} \;=\; \begin{pmatrix} \boldsymbol{x}_{\mathrm{cls}}^\top \\ \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_N^\top \end{pmatrix} \;=\; \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & e^{\psi_{1,1}} & \cdots & e^{\psi_{K,1}} \\ \vdots & \vdots & & \vdots \\ 1 & e^{\psi_{1,N}} & \cdots & e^{\psi_{K,N}} \end{pmatrix} \in \mathbb{R}^{(N+1)\times(K+1)}, \tag{S13}$$

whose rows are the tokens $\boldsymbol{x}_m$, with $\boldsymbol{x}_{\mathrm{cls}} = \boldsymbol{e}^{(0)}$ and $\boldsymbol{x}_n = \phi(z_n)$ for $n = 1, \ldots, N$. We term the first coordinate of each token as the anchor, and note it equals 1 in every row.

Self-attention. For the attention matrices, we set as the weights of the self-attention the rank-one choice $\boldsymbol{W}_Q\boldsymbol{W}_K^\top = \boldsymbol{e}^{(0)}\boldsymbol{e}^{(0)\top}$ and $\boldsymbol{W}_V = (N+1)\alpha\boldsymbol{I}_{K+1}$ for a nonzero scalar α. With this choice, the attention score between tokens m and m' before the softmax, $R_{m,m'}$, is the product of their anchors, which are all equal 1:

$$R_{m,m'} \;=\; \boldsymbol{x}_m^\top \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_{m'} \;=\; \big(\boldsymbol{x}_m^\top \boldsymbol{e}^{(0)}\big)\big(\boldsymbol{e}^{(0)\top}\boldsymbol{x}_{m'}\big) \;=\; 1. \tag{S14}$$

Then, the softmax over the $N+1$ tokens equalizes all attention scores:

$$A_{m,m'} \;=\; \frac{e^{R_{m,m'}}}{\sum_{m''} e^{R_{m,m''}}} \;=\; \frac{1}{N+1}. \tag{S15}$$

Notably, the attention scores remain equal under any attention scaling inside the softmax, such as the common choice of $1/\sqrt{d}$.

Therefore, after multiplying the softmax result by $\boldsymbol{W}_V$ and adding the residual connection, we get

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{A}\boldsymbol{X}\boldsymbol{W}_V. \tag{S16}$$

Specifically, for an input token $\boldsymbol{x}_m$ where $m = \text{cls}, 1, \ldots, N$, its output token $\tilde{\boldsymbol{x}}_m$ becomes

$$\begin{aligned} \tilde{\boldsymbol{x}}_m &= \boldsymbol{x}_m + (N+1)\alpha \sum_{m'} A_{m,m'}\boldsymbol{x}_{m'} \\ &= \underbrace{\boldsymbol{x}_m}_{\text{residual}} + \underbrace{\alpha \sum_{m'} \boldsymbol{x}_{m'}}_{\text{attention output} \equiv \alpha\boldsymbol{v}} . \end{aligned} \tag{S17}$$

Importantly, the attention output equals $\alpha\sum_{m'}\boldsymbol{x}_{m'}$ for all tokens, a constant which we denote $\alpha\boldsymbol{v}$. Thus, the residual element keeps the content specific to each token.

For the class token, the residual stream $\boldsymbol{e}^{(0)}$ is zero for all elements except for the first, such that

$$\tilde{\boldsymbol{x}}_{\text{cls}} \;=\; \boldsymbol{e}^{(0)} + \alpha\boldsymbol{v}, \qquad \big(\tilde{\boldsymbol{x}}_{\text{cls}}\big)_0 \;=\; 1 + \alpha(N+1), \qquad \underbrace{\big(\tilde{\boldsymbol{x}}_{\text{cls}}\big)_i \;=\; \alpha \sum_{n=1}^{N} e^{\psi_{i,n}} \;\equiv\; \alpha v_i}_{i=1,\ldots,K}. \tag{S18}$$

Head. For our learned head, we choose weights that ignore the anchor channel of the class-token and act on the remaining K coordinates as

$$g : \mathbb{R}^{K+1} \to \mathbb{R}^K, \qquad g(\boldsymbol{u})_i \;=\; \log\left(\frac{u_i}{\alpha}\right) + \beta_i, \qquad i = 1, \ldots, K, \tag{S19}$$

followed by a softmax over classes. The factor $1/\alpha$ undoes the factor α in the attention output, so by Eq. (S18) the class-token logit is $\log(v_i) + \beta_i$ for every α. This head implementation includes the outer logarithm and bias of Eq. (S11). The argument of the logarithm, which includes summation over exponents, is exactly the result of the self-attention with our chosen weights. Therefore, our chosen embedding, self-attention weights, and the head implementation compute together the optimal classifier exactly,

$$\text{softmax}_i\, g\big(\tilde{\boldsymbol{x}}_{\text{cls}}\big) \;=\; p_{\text{c}|\boldsymbol{z}}(i|\boldsymbol{z}) \qquad \text{for every } \boldsymbol{z} \in \mathbb{R}^N. \tag{S20}$$

Importantly, the class-token logits $\log(v_i) + \beta_i$ do not depend on α, so they will compute the same classification for every chosen nonzero α. This proves the first part of Theorem 1.

E.4 Proofs for the aligned and inverted regimes (items 1 and 2 of Theorem 1)

A common token-decoding explainability approach, which we use throughout the paper, applies the classification head, which is usually reserved for the class token, to the patch tokens as well. We denote the logit output for class i of the head when applied to a patch token as the score $\mathcal{S}(\cdot)_i$ from Sec. 3.1. Here, i is the true label of the object.

From Eq. (S17), for our choice of weights, the class-i channel of the patch token $\tilde{\boldsymbol{x}}_n$, after the attention and the residual connection and before the classification head, is

$$(\tilde{\boldsymbol{x}}_n)_i = e^{\psi_{i,n}} + \alpha v_i. \tag{S21}$$

Therefore, its score after passing through the classification head from Eq. (S19) is

$$\mathcal{S}(\tilde{\boldsymbol{x}}_n)_i = g(\tilde{\boldsymbol{x}}_n)_i = \log\left(\frac{e^{\psi_{i,n}}}{\alpha} + v_i\right) + \beta_i. \tag{S22}$$

We note that $\tilde{\boldsymbol{x}}_n$ is dependent on all the patch-tokens of the original input $\boldsymbol{z}$, as it is the output of the self-attention layer.

We now prove items 1 and 2 of Theorem 1: for every $\alpha > 0$ and every $\alpha \leq -1$, the expectation of the difference of the scores, $\mathbb{E}_{\mathbf{z}}[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \text{c} = i]$, has the sign of α, unless class i has the same distribution as the background. We show this in five steps, which are outlined below. First, we verify that the patch logit is defined and inspect how it changes with α. Next, we reduce the expectation of the difference from being computed over all patches in the image, to a single pair of patches. Then, we prove two properties of that difference of a pair of patches. Following that, we rewrite the expectation of the pair in a symmetric form. Finally, we analyze the symmetric form to understand the sign of the expression.

Step 1: the patch logit. Take any $\alpha > 0$ or any $\alpha \leq -1$, with the head of Eq. (S19). For $\alpha > 0$ the argument of the logarithm in Eq. (S22) is a sum of positive terms. For $\alpha \leq -1$ it stays positive as well, since $1/\alpha \geq -1$ gives

$$v_i + \frac{e^{\psi_{i,n}}}{\alpha} \geq v_i - e^{\psi_{i,n}} = \sum_{m \neq n} e^{\psi_{i,m}} > 0. \tag{S23}$$

Therefore, the patch logit is defined in both cases. Additionally, in both cases the class token still returns the same logit $\log(v_i) + \beta_i$, so by Eq. (S20) the model remains optimal. Crucially, for a specific image $\boldsymbol{z}$, v_i is the same for all patches, so the patch logit increases with $\psi_{i,n}$ when $\alpha > 0$ and decreases when $\alpha < 0$.

Step 2: reduce the expectation of the difference of scores from all patches in the image to one pair of patches. To consider the explainability of the image $\mathbf{z}$, we inspect the difference between $\mathcal{S}_{\text{obj}}$ and $\mathcal{S}_{\text{bg}}$ as defined in Eq. (1), for the object of class i that appears in $\boldsymbol{z}$. Here, since the object occupies a single token, $\mathcal{S}_{\text{obj}}$ is the logit at the position of the object patch n_{obj}, whereas $\mathcal{S}_{\text{bg}}$ is the mean logit over the $N-1$ background patches:

$$\begin{aligned}\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} &= \mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \frac{1}{N-1}\sum_{b \neq n_{\text{obj}}} \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i \\ &= \frac{1}{N-1}\sum_{b \neq n_{\text{obj}}} \left(\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i\right).\end{aligned} \tag{S24}$$

To derive the expectation of these differences, $\mathbb{E}_{\mathbf{z},\text{n}_{\text{obj}}}[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \text{c} = i]$, we first consider the case when we condition on $\text{n}_{\text{obj}} = n_{\text{obj}}$. Now, when the object position is fixed, the background values are independent and identically distributed, and every logit depends on the other patches only through the sum $v_i = \sum_{n=1}^{N} e^{\psi_{i,n}}$ (Eq. (S18)), which does not depend on their order. Exchanging two background patches therefore leaves the joint distribution of the logits unchanged, so $\mathbb{E}_{\mathbf{z}}\left[\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i | \text{c} = i, \text{n}_{\text{obj}} = n_{\text{obj}}\right]$ is the same for every background patch b. By linearity, and since all the terms in the sum are equal, for any single background patch b,

$$\begin{aligned}\mathbb{E}_{\mathbf{z}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \text{c} = i, \text{n}_{\text{obj}} = n_{\text{obj}}\right] &= \mathbb{E}_{\mathbf{z}}\left[\frac{1}{N-1}\sum_{b \neq n_{\text{obj}}} \left(\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i\right) \,\middle|\, \text{c} = i, \text{n}_{\text{obj}} = n_{\text{obj}}\right] \\ &= \frac{1}{N-1}\sum_{b \neq n_{\text{obj}}} \mathbb{E}_{\mathbf{z}}\left[\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i \,\middle|\, \text{c} = i, \text{n}_{\text{obj}} = n_{\text{obj}}\right] \\ &= \mathbb{E}_{\mathbf{z}}\left[\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i \,\middle|\, \text{c} = i, \text{n}_{\text{obj}} = n_{\text{obj}}\right].\end{aligned} \tag{S25}$$

This expectation is the same for every object position n_{obj}, since both the data distribution and the transformer, which has no positional embedding, treat all positions alike. Since n_{obj} is uniform and independent of the class, the law of total expectation over it gives, for any fixed n_{obj} and any background patch $b \neq n_{\text{obj}}$,

$$\begin{aligned}\mathbb{E}_{\mathbf{z},\mathrm{n}_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] &= \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{\mathbf{z}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n\right] \\ &= \mathbb{E}_{\mathbf{z}}\left[\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i \,\middle|\, \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right]. \end{aligned} \tag{S26}$$

In particular, this means we can inspect the expectation of the difference of the object patch and a single background patch, instead of considering all background patches.

Step 3: two properties of the difference of a single pair, $D_i(x, y; r)$. First, $\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i$ from Eq. (S25) satisfies

$$\begin{aligned}\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i &= \log\left(\frac{e^{\psi_i(z_{n_{\text{obj}}})}}{\alpha} + v_i\right) - \log\left(\frac{e^{\psi_i(z_b)}}{\alpha} + v_i\right) \\ &= \log\left(\frac{e^{\psi_i(z_{n_{\text{obj}}})}}{\alpha} + e^{\psi_i(z_{n_{\text{obj}}})} + e^{\psi_i(z_b)} + r\right) - \\ &\qquad \log\left(\frac{e^{\psi_i(z_b)}}{\alpha} + e^{\psi_i(z_{n_{\text{obj}}})} + e^{\psi_i(z_b)} + r\right), \end{aligned} \tag{S27}$$

as β_i cancels. Here, $r = \sum_{m \notin \{n_{\text{obj}}, b\}} e^{\psi_{i,m}}$, collects all $N-2$ background patches apart from z_b.

Now, we construct a function $D_i(x, y; r)$ of real numbers x, y and $r \geq 0$, which we evaluate at the object entry $x = z_{n_{\text{obj}}}$, the entry $y = z_b$ of a background patch b in the same image:

$$D_i(x, y; r) \;=\; \log\left(\frac{e^{\psi_i(x)}}{\alpha} + e^{\psi_i(x)} + e^{\psi_i(y)} + r\right) - \log\left(\frac{e^{\psi_i(y)}}{\alpha} + e^{\psi_i(x)} + e^{\psi_i(y)} + r\right). \tag{S28}$$

Both arguments of the logarithms are positive, as in Step 1, since the first equals $(1 + 1/\alpha)e^{\psi_i(x)} + e^{\psi_i(y)} + r$ with $1 + 1/\alpha \geq 0$, and similarly for the second. Importantly, the score difference is

$$\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i \;=\; D_i(z_{n_{\text{obj}}}, z_b; r), \tag{S29}$$

with r computed from the same image. The function D_i operates on the patch entries directly, whereas the scores operate on the same patches after the embedding and the self-attention.

Given $\mathrm{c} = i$ and $\mathrm{n}_{\text{obj}} = n_{\text{obj}}$, the entries of the image are independent (Eq. (S1)), so $\mathrm{z}_{n_{\text{obj}}}$, z_b and $\mathrm{r} = \sum_{m \notin \{n_{\text{obj}}, b\}} e^{\psi_i(\mathrm{z}_m)}$ are independent, since r is computed from the other $N-2$ background entries. Substituting Eq. (S29) into Eq. (S26) and applying the law of total expectation over r gives

$$\begin{aligned}\mathbb{E}_{\mathbf{z},\mathrm{n}_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] &= \mathbb{E}_{\mathrm{z}_{n_{\text{obj}}}, \mathrm{z}_b, \mathrm{r}}\left[D_i(\mathrm{z}_{n_{\text{obj}}}, \mathrm{z}_b; \mathrm{r}) | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right] \\ &= \mathbb{E}_{\mathrm{r}}\left[\mathbb{E}_{\mathrm{z}_{n_{\text{obj}}}, \mathrm{z}_b}\left[D_i(\mathrm{z}_{n_{\text{obj}}}, \mathrm{z}_b; \mathrm{r}) | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}, \mathrm{r}\right]\right]. \end{aligned} \tag{S30}$$

(a) D_i is an antisymmetric function. Exchanging x and y leaves the sum $e^{\psi_i(x)} + e^{\psi_i(y)} + r$ unchanged and exchanges the two logarithms in Eq. (S28), so

$$D_i(y, x; r) \;=\; -D_i(x, y; r) \quad \text{for every } r. \tag{S31}$$

(b) The sign of D_i follows α and the likelihood ratio. Given $\mathrm{c} = i$ and $\mathrm{n}_{\text{obj}} = n_{\text{obj}}$, the object entry has density $f_{\text{obj}}(z) = \mathcal{N}(z; \mu_i, \sigma_i^2)$ and each background entry has density $f_{\text{bg}}(z) = \mathcal{N}(z; 0, \sigma^2)$ (App. E.1). Let $L(z) = f_{\text{obj}}(z)/f_{\text{bg}}(z)$ be the likelihood ratio of the object to the background. Dividing the two Gaussian densities,

$$L(z) \;=\; \frac{\frac{1}{\sqrt{2\pi}\,\sigma_i}\exp\left(-\frac{(z-\mu_i)^2}{2\sigma_i^2}\right)}{\frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{z^2}{2\sigma^2}\right)} \;=\; \frac{\sigma}{\sigma_i}\exp\left(\frac{z^2}{2\sigma^2} - \frac{(z-\mu_i)^2}{2\sigma_i^2}\right) \;=\; \frac{\sigma}{\sigma_i}\,e^{\psi_i(z)}, \tag{S32}$$

so $L(z)$ equals $e^{\psi_i(z)}$ up to a positive factor, and is positive itself. Next, considering D_i, both logarithms in Eq. (S28) contain the same sum $v_i = e^{\psi_i(x)} + e^{\psi_i(y)} + r$, so

$$\begin{aligned}
D_i(x,y;r) > 0 &\iff \log\left(\frac{e^{\psi_i(x)}}{\alpha} + v_i\right) > \log\left(\frac{e^{\psi_i(y)}}{\alpha} + v_i\right) && \text{(by the definition of } D_i\text{)} \\
&\iff \frac{e^{\psi_i(x)}}{\alpha} + v_i \;>\; \frac{e^{\psi_i(y)}}{\alpha} + v_i && \text{(the logarithm is increasing)} \\
&\iff \frac{e^{\psi_i(x)}}{\alpha} \;>\; \frac{e^{\psi_i(y)}}{\alpha} && \text{(subtract } v_i \text{ from both sides)} \\
&\iff \alpha\big(e^{\psi_i(x)} - e^{\psi_i(y)}\big) \;>\; 0 && \text{(multiply both sides by } \alpha^2 > 0\text{)} \\
&\iff \alpha\big(L(x) - L(y)\big) \;>\; 0 && \text{(by Eq. (S32)).} \qquad \text{(S33)}
\end{aligned}$$

Every step is an equivalence, and the same chain with $<$ or $=$ in place of $>$ covers the other two cases. Together,

$$\operatorname{sgn}\{D_i(x,y;r)\} = \operatorname{sgn}\left\{\alpha\big(L(x) - L(y)\big)\right\}. \qquad \text{(S34)}$$

Since v_i cancels in this chain, Eq. (S34) holds for every value of r.

Step 4: a symmetric form of the expectation. We now compute the inner expectation of the RHS of Eq. (S30) for a fixed value of r. Given $\mathrm{c} = i$ and $\mathrm{n_{obj}} = n_{\mathrm{obj}}$, the random variables $\mathrm{z}_{n_{\mathrm{obj}}}$, z_b and r are mutually independent (Step 3), so fixing $\mathrm{r} = r$ does not change the joint density $f_{\mathrm{obj}}(x) f_{\mathrm{bg}}(y)$ of $\mathrm{z}_{n_{\mathrm{obj}}}$ and z_b. Therefore,

$$\begin{aligned}
&\mathbb{E}\left[D_i(\mathrm{z}_{n_{\mathrm{obj}}}, \mathrm{z}_b; \mathrm{r}) \,|\, \mathrm{c} = i, \mathrm{n_{obj}} = n_{\mathrm{obj}}, \mathrm{r} = r\right] \\
&\quad = \iint f_{\mathrm{obj}}(x)\, f_{\mathrm{bg}}(y)\, D_i(x,y;r)\, dx\, dy && \text{(definition of the expectation)} \\
&\quad = \iint f_{\mathrm{bg}}(x)\, f_{\mathrm{bg}}(y)\, L(x)\, D_i(x,y;r)\, dx\, dy && \text{(since } f_{\mathrm{obj}} = L\, f_{\mathrm{bg}}\text{)} \\
&\quad = \iint f_{\mathrm{bg}}(y)\, f_{\mathrm{bg}}(x)\, L(y)\, D_i(y,x;r)\, dy\, dx && \text{(renaming } x \leftrightarrow y\text{)} \\
&\quad = -\iint f_{\mathrm{bg}}(x)\, f_{\mathrm{bg}}(y)\, L(y)\, D_i(x,y;r)\, dx\, dy && \text{(by antisymmetry).} \qquad \text{(S35)}
\end{aligned}$$

The fourth line only changes the names of the two integration variables, which does not change the value of the integral. The third and fifth lines are two expressions for the same quantity, so adding them and dividing by two gives

$$\begin{aligned}
&\mathbb{E}\left[D_i(\mathrm{z}_{n_{\mathrm{obj}}}, \mathrm{z}_b; \mathrm{r}) \,|\, \mathrm{c} = i, \mathrm{n_{obj}} = n_{\mathrm{obj}}, \mathrm{r} = r\right] \\
&\quad = \frac{1}{2}\iint f_{\mathrm{bg}}(x)\, f_{\mathrm{bg}}(y)\, \big(L(x) - L(y)\big)\, D_i(x,y;r)\, dx\, dy. \qquad \text{(S36)}
\end{aligned}$$

Step 5: the sign. By Eq. (S34), for every x, y and r, the integrand of Eq. (S36) satisfies

$$\operatorname{sgn}\big\{f_{\mathrm{bg}}(x)\, f_{\mathrm{bg}}(y)\big(L(x) - L(y)\big) D_i(x,y;r)\big\} = \operatorname{sgn}\big\{\alpha\big(L(x) - L(y)\big)^2\big\}, \qquad \text{(S37)}$$

since $f_{\mathrm{bg}}(x) > 0$ for Gaussian distributions. The integrand is therefore non-negative when $\alpha > 0$ and non-positive when $\alpha \le -1$, so by Eq. (S36), for every r,

$$\begin{aligned}
\mathbb{E}\left[D_i(\mathrm{z}_{n_{\mathrm{obj}}}, \mathrm{z}_b; \mathrm{r}) \,|\, \mathrm{c} = i, \mathrm{n_{obj}} = n_{\mathrm{obj}}, \mathrm{r} = r\right] &\;\ge\; 0 \quad \text{if } \alpha > 0, \\
\mathbb{E}\left[D_i(\mathrm{z}_{n_{\mathrm{obj}}}, \mathrm{z}_b; \mathrm{r}) \,|\, \mathrm{c} = i, \mathrm{n_{obj}} = n_{\mathrm{obj}}, \mathrm{r} = r\right] &\;\le\; 0 \quad \text{if } \alpha \le -1. \qquad \text{(S38)}
\end{aligned}$$

Specifically, the integrand will be zero only when $L(x) = L(y)$ for all x and y values. Thus, for the inequalities to be strict the integrand of Eq. (S36) must be nonzero over some x and y values.

Lastly, we show that these inequalities are strict. Recall that $L(z)$ is the likelihood ratio of the two distributions $f_{\mathrm{obj}}(z)$ and $f_{\mathrm{bg}}(z)$. Since the distribution of class i differs from that of the background, $f_{\mathrm{obj}}(z) \neq f_{\mathrm{bg}}(z)$, $L(z)$ is not a constant function. As $L(z)$ is also continuous, there exists some area in the (x, y) plane in which $L(x) \neq L(y)$ for every pair (x, y). Thus, in this area

$\alpha\big(L(x) - L(y)\big)^2 \neq 0$. So, by Eq. (S37) the integrand of Eq. (S36) is nonzero, and so is the integral. The inequalities in Eq. (S38) are therefore strict, that is,

$$\begin{aligned} \mathbb{E}\left[D_i(\mathrm{z}_{n_{\text{obj}}}, \mathrm{z}_b; \mathrm{r}) \,|\, \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}, \mathrm{r} = r\right] &> 0 \quad \text{if } \alpha > 0, \\ \mathbb{E}\left[D_i(\mathrm{z}_{n_{\text{obj}}}, \mathrm{z}_b; \mathrm{r}) \,|\, \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}, \mathrm{r} = r\right] &< 0 \quad \text{if } \alpha \leq -1. \end{aligned} \tag{S39}$$

Finally, the outer expectation in Eq. (S30) averages this inner expectation over r, so it has the same sign, which gives

$$\mathbb{E}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] > 0 \quad \text{if } \alpha > 0, \qquad \mathbb{E}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] < 0 \quad \text{if } \alpha \leq -1. \tag{S40}$$

This proves items 1 and 2 of Theorem 1.

E.5 The unexplainable regime (item 3 of Theorem 1)

We now prove item 3 of Theorem 1. Factoring v_i out of the logarithm in Eq. (S22), the logit of patch n is

$$\begin{aligned} \mathcal{S}(\tilde{\boldsymbol{x}}_n)_i &= \log\left(\frac{e^{\psi_{i,n}}}{\alpha} + v_i\right) + \beta_i \\ &= \log(v_i) + \beta_i + \log(1 + t_n), \qquad t_n = \frac{e^{\psi_{i,n}}}{\alpha\, v_i}. \end{aligned} \tag{S41}$$

The first two terms are the same for every patch. The sum v_i contains $e^{\psi_{i,n}}$ and all its terms are positive, so $0 < e^{\psi_{i,n}} \leq v_i$. Therefore, $|t_n| \leq \frac{1}{|\alpha|}$. Thus, when $|\alpha| > 1$ the logarithm is defined.

As $|\alpha| \to \infty$, $t_n \to 0$ for every patch, so $\log(1 + t_n) \to 0$. Thus, every patch logit tends to the same value $\log(v_i) + \beta_i$, *i.e.*, $\mathcal{S}(\tilde{\boldsymbol{x}}_n)_i \to \log(v_i) + \beta_i$. The scores $\mathcal{S}_{\text{obj}}$ and $\mathcal{S}_{\text{bg}}$ are averages of patch logits, so both tend to this value and $\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} \to 0$ in every image. In the limit, all patch scores are the same, which is the unexplainable case of Sec. 3.1, while the classifier stays optimal. This proves item 3 of Theorem 1.

In conclusion, a single scalar α leaves the class-token logits unchanged but can produce all three regimes. In expectation, any $\alpha > 0$ gives an aligned map and any $\alpha \leq -1$ an inverted one, and as $|\alpha|$ grows the map becomes unexplainable in every image.

E.6 The role of the logarithm in the head

In standard, real-world ViTs, the classification head is typically linear, whereas the head used in our toy model, Eq. (S19), is not. Specifically, the optimal logits of Proposition 1 contain a logarithm of a sum over all positions, since the object may lie in any of them. Thus, our toy head implementation applies exactly this logarithm. Since token-decoding applies the same head to the patch tokens, one may ask whether the three regimes of Theorem 1 stem from this logarithm. Therefore, we further examine the case where instead of a logarithmic head, we use a linear head similar to standard ViTs. As outlined below, using a linear head only causes the exactness of the predicted probabilities to be lost. That is, a model can converge to all three regimes while producing the same classification logits for every α, where the resulting classifier is only an approximation of the optimal one, as it induces the optimal decision rule but not the exact posterior probabilities.

Since $\beta_i = \log(\pi_i/\sigma_i)$, the head of Eq. (S19) can also be written as $g(\boldsymbol{u})_i = \log(c_i u_i)$, with the per-class weight $c_i = \pi_i/(\alpha\sigma_i)$. This form is equivalent to a linear layer followed by an elementwise logarithm. Removing this logarithm leaves the linear head $h(\boldsymbol{u})_i = c_i u_i$, so that $g(\boldsymbol{u})_i = \log\big(h(\boldsymbol{u})_i\big)$.

First, we note that the linear head h predicts from the class token the same optimal decision as the optimal logarithmic head g of Proposition 1, and assigns to every patch token the same class as g. This is because the logarithm function is increasing and acts on each class channel separately, so $\arg\max_i g(\boldsymbol{u})_i = \arg\max_i h(\boldsymbol{u})_i$ for every token. For the same reason, both heads rank the patches of an image identically for every class.

Next, we show h has the same three regimes as g, by following the steps of the proof in App. E.4. First, step 1 is not needed once the logarithm is removed, so items 1 and 2 will hold for every $\alpha \neq 0$,

and step 2 does not depend on the head. Next, under h, the patch score of Eq. (S22) is

$$\mathcal{S}(\tilde{\boldsymbol{x}}_n)_i = h(\tilde{\boldsymbol{x}}_n)_i = \frac{\pi_i}{\sigma_i}\left(\frac{e^{\psi_{i,n}}}{\alpha} + v_i\right), \tag{S42}$$

and the class-token logit is $\pi_i v_i/\sigma_i$, which does not change for different α values. Additionally, Eq. (S31) still holds since exchanging the two patches still flips the sign of their difference. Since the first equivalence of Eq. (S33) removes the logarithm for being an increasing function, steps 3 and 4 remain unchanged with the modified patch score, as does step 5, which finalizes the proof for the regimes of items 1 and 2 in Theorem 1. Finally, item 3 holds as well, because the patch scores differ only through $e^{\psi_{i,n}}/\alpha$, which vanishes as $|\alpha| \to \infty$.

What the logarithm provides is therefore the exact posterior of Eq. (S20), as h attains the optimal decision rule but no longer minimizes the cross-entropy. Note that real-world models do not converge to the optimal classifier either, since they are trained on finite data with a finite architecture, so h is in fact closer than g to the models we analyze in Secs. 3.2 and 3.3.

The expected difference in closed form. The linear head is also easier to analyze and yields the expected difference of the scores in closed form for every α, which for the head g we obtain only as $\alpha \to 0$ (App. E.7). The term v_i is shared by both scores, so it cancels in the difference of a single object-background pair. For the object entry $x = z_{n_{\text{obj}}}$ and a background entry $y = z_b$,

$$\begin{aligned}\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i &= \frac{\pi_i}{\sigma_i}\left(\frac{e^{\psi_i(x)}}{\alpha} + v_i\right) - \frac{\pi_i}{\sigma_i}\left(\frac{e^{\psi_i(y)}}{\alpha} + v_i\right) \\ &= \frac{\pi_i}{\sigma_i}\cdot\frac{e^{\psi_i(x)} - e^{\psi_i(y)}}{\alpha}.\end{aligned} \tag{S43}$$

The expectation of Eq. (S43) therefore exists only if e^{ψ_i} has a finite expectation under each of the two distributions separately. By Eq. (S32), e^{ψ_i} is the likelihood ratio L scaled by σ_i/σ. Taking the expectation with respect to the distribution of the background patches gives $\mathbb{E}_{z\sim f_{\text{bg}}}\left[e^{\psi_i(z)}\right] = \frac{\sigma_i}{\sigma}\int L(z) f_{\text{bg}}(z)\,dz = \frac{\sigma_i}{\sigma}$, which is always finite since $L f_{\text{bg}} = f_{\text{obj}}$ integrates to one. Taking it with respect to the distribution of the object patch, the two copies of $(z-\mu_i)^2$ combine, and with $a = \frac{1}{\sigma_i^2} - \frac{1}{2\sigma^2}$ and $b = \frac{\mu_i}{\sigma_i^2}$,

$$\begin{aligned}\mathbb{E}_{z\sim f_{\text{obj}}}\left[e^{\psi_i(z)}\right] &= \frac{1}{\sqrt{2\pi}\,\sigma_i}\int \exp\left(\frac{z^2}{2\sigma^2} - \frac{(z-\mu_i)^2}{\sigma_i^2}\right) dz \\ &= \frac{1}{\sqrt{2\pi}\,\sigma_i}\int \exp\left(-az^2 + 2bz - \frac{\mu_i^2}{\sigma_i^2}\right) dz && \text{(a quadratic in } z\text{)} \\ &= \frac{1}{\sqrt{2\pi}\,\sigma_i}\, e^{\frac{b^2}{a} - \frac{\mu_i^2}{\sigma_i^2}} \int \exp\left(-a\left(z - \frac{b}{a}\right)^2\right) dz && \text{(completing the square)} \\ &= \frac{1}{\sqrt{2\pi}\,\sigma_i}\sqrt{\frac{\pi}{a}}\, e^{\frac{b^2}{a} - \frac{\mu_i^2}{\sigma_i^2}} && \text{(a Gaussian integral)} \\ &= \frac{\sigma}{\sqrt{2\sigma^2 - \sigma_i^2}}\exp\left(\frac{\mu_i^2}{2\sigma^2 - \sigma_i^2}\right) && \text{(substituting } a \text{ and } b\text{).}\end{aligned} \tag{S44}$$

The Gaussian integral in the fourth line converges only for $a > 0$, that is, only when $\sigma_i < \sqrt{2}\sigma$. The last line uses $a = \frac{2\sigma^2 - \sigma_i^2}{2\sigma^2\sigma_i^2}$, which gives $\frac{1}{\sigma_i}\sqrt{\frac{1}{2a}} = \frac{\sigma}{\sqrt{2\sigma^2 - \sigma_i^2}}$ for the prefactor and $\frac{b^2}{a} - \frac{\mu_i^2}{\sigma_i^2} = \frac{\mu_i^2}{2\sigma^2 - \sigma_i^2}$ for the exponent.

By step 2 of App. E.4, the expected difference of the scores equals the expectation of Eq. (S43), so substituting the two expectations gives

$$\mathbb{E}_{\mathbf{z},n_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} \mid \mathrm{c} = i\right] = \frac{\pi_i}{\sigma_i\alpha}\left(\frac{\sigma}{\sqrt{2\sigma^2 - \sigma_i^2}}\exp\left(\frac{\mu_i^2}{2\sigma^2 - \sigma_i^2}\right) - \frac{\sigma_i}{\sigma}\right), \tag{S45}$$

for every $\alpha \neq 0$ and every class i with $\sigma_i < \sqrt{2}\sigma$. The bracket in Eq. (S45) is non-negative, and vanishes only when $\mu_i = 0$ and $\sigma_i = \sigma$, that is, only when class i has the same distribution as the

background. The sign of the expected difference is therefore that of α, which recovers items 1 and 2 of Theorem 1 for h directly from the closed form.

When $\sigma_i \geq \sqrt{2}\sigma$, that is, when class i is more than $\sqrt{2}$ times as wide as the background, the expectation under the object is infinite while the one under the background stays finite, so the expected difference diverges to $+\infty$ for $\alpha > 0$ and to $-\infty$ for $\alpha < 0$, and still carries the sign of α. Under g, the same expectations are taken after a logarithm, which compresses L, so no closed form of this kind is available and no such condition is needed.

E.7 The expected score difference as $\alpha \to 0$

The proof in App. E.4 gives the sign of the expected difference of the scores for every $\alpha > 0$, but not its size. For small $\alpha > 0$ the size has a closed form, which we derive here for completeness. Fix an object position n_{obj} and a background patch $b \neq n_{\text{obj}}$. By Eq. (S26), for $\alpha > 0$

$$\begin{aligned}
\mathbb{E}_{\mathbf{z},\mathrm{n}_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] &= \mathbb{E}_{\mathbf{z}}\left[\mathcal{S}(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - \mathcal{S}(\tilde{\boldsymbol{x}}_b)_i | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[g(\tilde{\boldsymbol{x}}_{n_{\text{obj}}})_i - g(\tilde{\boldsymbol{x}}_b)_i | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\log\left(\frac{e^{\psi_{i,n_{\text{obj}}}}}{\alpha} + v_i\right) + \beta_i - \log\left(\frac{e^{\psi_{i,b}}}{\alpha} + v_i\right) - \beta_i \middle| \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\log\left(\frac{e^{\psi_{i,n_{\text{obj}}}}}{\alpha} + v_i\right) - \log\left(\frac{e^{\psi_{i,b}}}{\alpha} + v_i\right) \middle| \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\log\left(e^{\psi_{i,n_{\text{obj}}}} + \alpha v_i\right) - \log\left(e^{\psi_{i,b}} + \alpha v_i\right) | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right]. \quad \text{(S46)}
\end{aligned}$$

Here, the expectations are over the whole image given $\mathrm{c} = i$ and $\mathrm{n}_{\text{obj}} = n_{\text{obj}}$, the fourth line cancels β_i, and the fifth cancels $-\log\alpha$, using $\log(x/\alpha + v_i) = -\log\alpha + \log(x + \alpha v_i)$.

As $\alpha \to 0$ the αv_i terms become negligible compared with the exponential terms, which do not depend on α. By dominated convergence, the limit can be taken inside the expectation, which gives

$$\lim_{\alpha\to 0} \mathbb{E}_{\mathbf{z},\mathrm{n}_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] = \mathbb{E}_{\mathbf{z}}\left[\log\left(e^{\psi_{i,n_{\text{obj}}}}\right) - \log\left(e^{\psi_{i,b}}\right) | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right]. \quad \text{(S47)}$$

Therefore, each of the two terms involves a single patch, distributed as $\mathcal{N}(\mu_i, \sigma_i^2)$ for the object and $\mathcal{N}(0, \sigma^2)$ for the background. Thus, the two expectations can be taken over those distributions separately:

$$\lim_{\alpha\to 0} \mathbb{E}_{\mathbf{z},\mathrm{n}_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] = \mathbb{E}_{\mathrm{z}_{n_{\text{obj}}}}\left[\psi_{i,n_{\text{obj}}} | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right] - \mathbb{E}_{\mathrm{z}_b}\left[\psi_{i,b} | \mathrm{c} = i, \mathrm{n}_{\text{obj}} = n_{\text{obj}}\right].$$

Given $\mathrm{c} = i$ and $\mathrm{n}_{\text{obj}} = n_{\text{obj}}$, the object entry $\mathrm{z}_{n_{\text{obj}}} \sim \mathcal{N}(\mu_i, \sigma_i^2)$ has $\mathbb{E}[\mathrm{z}_{n_{\text{obj}}}^2] = \sigma_i^2 + \mu_i^2$ and $\mathbb{E}[(\mathrm{z}_{n_{\text{obj}}} - \mu_i)^2] = \sigma_i^2$, and the background entry $\mathrm{z}_b \sim \mathcal{N}(0, \sigma^2)$ has $\mathbb{E}[\mathrm{z}_b^2] = \sigma^2$ and $\mathbb{E}[(\mathrm{z}_b - \mu_i)^2] = \sigma^2 + \mu_i^2$, which gives

$$\begin{aligned}
&\lim_{\alpha\to 0} \mathbb{E}_{\mathbf{z},\mathrm{n}_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] \\
&\quad = \mathbb{E}_{\mathrm{z}_{n_{\text{obj}}} \sim \mathcal{N}(\mu_i, \sigma_i^2)}\left[\frac{\mathrm{z}_{n_{\text{obj}}}^2}{2\sigma^2} - \frac{(\mathrm{z}_{n_{\text{obj}}} - \mu_i)^2}{2\sigma_i^2}\right] - \mathbb{E}_{\mathrm{z}_b \sim \mathcal{N}(0,\sigma^2)}\left[\frac{\mathrm{z}_b^2}{2\sigma^2} - \frac{(\mathrm{z}_b - \mu_i)^2}{2\sigma_i^2}\right] \\
&\quad = \left(\frac{\sigma_i^2 + \mu_i^2}{2\sigma^2} - \frac{\sigma_i^2}{2\sigma_i^2}\right) - \left(\frac{\sigma^2}{2\sigma^2} - \frac{\sigma^2 + \mu_i^2}{2\sigma_i^2}\right) \\
&\quad = \frac{\sigma_i^2 + \mu_i^2}{2\sigma^2} + \frac{\sigma^2 + \mu_i^2}{2\sigma_i^2} - 1. \quad \text{(S48)}
\end{aligned}$$

Grouping the terms,

$$\lim_{\alpha\to 0} \mathbb{E}_{\mathbf{z},\mathrm{n}_{\text{obj}}}\left[\mathcal{S}_{\text{obj}} - \mathcal{S}_{\text{bg}} | \mathrm{c} = i\right] = \underbrace{\left(\frac{\sigma_i^2}{2\sigma^2} + \frac{\sigma^2}{2\sigma_i^2} - 1\right)}_{\geq 0} + \underbrace{\frac{\mu_i^2}{2}\left(\frac{1}{\sigma^2} + \frac{1}{\sigma_i^2}\right)}_{\geq 0}. \quad \text{(S49)}$$

The first bracket is non-negative because, over a common denominator, it is a perfect square,

$$\frac{\sigma_i^2}{2\sigma^2} + \frac{\sigma^2}{2\sigma_i^2} - 1 = \frac{\sigma_i^4 - 2\sigma^2\sigma_i^2 + \sigma^4}{2\sigma^2\sigma_i^2} = \frac{\left(\sigma_i^2 - \sigma^2\right)^2}{2\sigma^2\sigma_i^2} \geq 0, \quad \text{(S50)}$$

and it is zero only when $\sigma_i = \sigma$. The second is non-negative and is zero only when $\mu_i = 0$. The limit is therefore positive unless $\mu_i = 0$ and $\sigma_i = \sigma$, that is, unless the class has the same distribution as the background and carries no information. This agrees with item 1 of Theorem 1, and shows that for small α the difference grows with $|\mu_i|$ and with the mismatch between σ_i^2 and σ^2.