

Unconstrained Open Vocabulary Image Classification: Zero-Shot Transfer from Text to Image via CLIP Inversion

Philipp Allgeuer

Kyra Ahrens

Stefan Wermter

University of Hamburg

{philipp.allgeuer, kyra.ahrens, stefan.wermter}@uni-hamburg.de

Abstract

We introduce NOVIC, an innovative real-time *uNconstrained Open Vocabulary Image Classifier* that uses an autoregressive transformer to generatively output classification labels as language. Leveraging the extensive knowledge of CLIP models, NOVIC harnesses the embedding space to enable zero-shot transfer from pure text to images. Traditional CLIP models, despite their ability for open vocabulary classification, require an exhaustive prompt of potential class labels, restricting their application to images of known content or context. To address this, we propose an “object decoder” model that is trained on a large-scale 92M-target dataset of templated object noun sets and LLM-generated captions to always output the object noun in question. This effectively inverts the CLIP text encoder and allows textual object labels from essentially the entire English language to be generated directly from image-derived embedding vectors, without requiring any *a priori* knowledge of the potential content of an image, and without any label biases. The trained decoders are tested on a mix of manually and web-curated datasets, as well as standard image classification benchmarks, and achieve fine-grained prompt-free prediction scores of up to 87.5%, a strong result considering the model must work for any conceivable image and without any contextual clues.¹

1. Introduction

Imagine a robot navigating through an environment, encountering a small cup, and an operator asking, “*Can you tell me what that is?*” Ideally, the robot would recognize the object with the objectives to: a) answer *cup* directly, and b) provide an answer immediately in real-time, even if there are large numbers of objects to consider at high frame rates. Contrary to the first objective, Vision-Language Models (VLM) like CLIP [53] require the robot to be prompted with a comprehensive textual list of object class candidates to perform zero-shot classification. On the other hand, equipping the

¹The authors gratefully acknowledge support from the DFG (TRR 169 – CML) and European Commission (TRAIL).

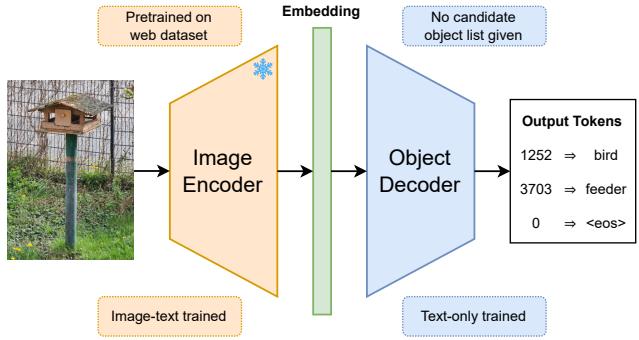


Figure 1. **Proposed open vocabulary image classifier.** The classifier processes each input image by encoding it into a CLIP embedding vector, which is then decoded into a sequence of tokens representing the object class. The object decoder is purely generative, producing free-form text without relying on a predefined list of candidate objects. This method is trained solely on textual data and is capable of zero-shot transfer to image data during inference.

robot with a high-performing multimodal Large Language Model (LLM) is computationally expensive, requires specialized remote servers for inference, and does not allow for the real-time responsiveness needed for video frame rates, thereby failing to meet the second objective. The core aim of this work is to develop models capable of unconstrained open vocabulary image classification. The models should generate *object nouns* directly as free-form text in real time, without the need for prompts or predefined label candidates.

Current approaches to open vocabulary learning [65] with VLMs for visual reasoning tasks (*e.g.* Tag2Text [29] and RAM [76] for image tagging) are in general limited by the need for candidate lists, which, with at most a few thousand entries, are not comprehensive enough to encompass all potential object classes. Although some parallel works propose strategies to utilize VLMs for generative tasks [22, 41], these are still ultimately restricted in diversity by the class candidates of labeled training datasets, leaving truly unconstrained prompt-free image classification unresolved. To address this gap in current research, we propose NOVIC, a novel *uNconstrained Open Vocabulary Image Classifier* (cf. Fig. 1). The

classifier can directly label any given image with an English object noun without requiring any category prompts. Thanks to the use of synthetic large-scale text-only datasets, very efficient and scalable training of the so-called *object decoder* transformer is achieved by training purely on CLIP text embeddings. The decoder effectively learns to invert a CLIP text encoder, mapping embeddings back to object nouns via autoregressive token sequence generation. By leveraging the shared multimodal representations of pretrained VLMs, NOVIC can perform zero-shot image classification without ever having seen images during training.

To train the object decoder, we introduce an *object noun dictionary*, and use it to build several synthetically and LLM-generated caption-object datasets. These datasets are all created through automated pipelines, allowing for a high degree of scalability, especially compared to image-based training. Inspired by prior works [22, 50], noise augmentation is used on CLIP embeddings of the captions during training to bridge the large modality gap between image and text embeddings [44]. The resulting object decoder can be evaluated on classification datasets like ImageNet-1K [54], but this does not fully capture its prompt-free open vocabulary performance. To address this, we curate three open vocabulary image datasets from original and web sources, and annotate them using human and LLM knowledge.

In summary, our main contributions² are as follows: (i) we propose a novel open vocabulary object decoder model that is trained on text only and can zero-shot classify arbitrary images in real-time without any candidates or prompts, (ii) we develop an automated pipeline to construct a comprehensive English object noun dictionary and use a multi-set prompting scheme combined with an LLM to construct large-scale synthetic caption-object datasets, (iii) we curate three new image sets intended for evaluation of open vocabulary classification and provide human and multimodal LLM annotations, and (iv) we show that our method scales in performance with the underlying CLIP model, and is able to provide accurate yet very fine-grained classifications while achieving strong in-the-wild prediction scores up to 87.5%.

2. Related Work

Learning a joint embedding space for multiple modalities such as images and text via contrastive pretraining produces powerful representations without the need for expensive annotation of datasets [19, 39]. VLMs such as CLIP [53] and ALIGN [30] paved the way for learning a broad range of visual concepts that can be transferred to downstream discriminative tasks, such as classification, via prompting [21]. Given an image, a set of candidate categories, and prompt templates like “*a photo of a [...]*”, CLIP can zero-shot classify the image by maximizing the similarity score between its

²All code [2, 3] (GitHub) and datasets [4, 5] are publicly available.

encoded representation and all encoded candidate prompts. NOVIC, by contrast, frames the image classification problem as an unconstrained generative one, eliminating the need for category candidates and inference-time prompting. Various approaches build upon and extend the idea behind CLIP towards better pretraining [42, 59, 74] or fine-tuning [63] techniques, or add additional modalities such as audio [24, 64], video [1, 28, 46, 69, 72], or point clouds [71, 75]. Other approaches use VLMs like CLIP as feature extractors for subsequent fine-tuning on more challenging downstream tasks such as visual question-answering [14, 56, 58], captioning [6, 48], motion planning [57, 60], and navigation [33].

Open vocabulary learning is a paradigm that emerged in the context of fine-tuning pretrained VLMs for vision tasks. It aims to overcome the closed-set assumption of standard classification models by allowing them to handle novel object categories outside of their training domain. Models such as ViLD [23], RegionCLIP [77], and various successor models [11, 13, 32, 51, 66, 67] enable VLMs to solve tasks such as open vocabulary object detection [73], where, given an arbitrary set of candidate object labels, the model has to detect the corresponding objects in the image. Other works extend the open vocabulary paradigm to closely related visual tasks such as semantic segmentation [25, 34, 37, 43, 45, 70] and image tagging [29, 40, 76]. Despite their merit, all the approaches mentioned above rely on candidate labels for inference, a strong assumption that we abandon in this work.

A parallel line of research aims to leverage the powerful representations provided by VLMs for generative tasks such as image captioning. As VLMs like CLIP cannot inherently produce sequences, approaches in this research area rely on additional decoders. Several methods have been proposed that either establish a mapping between a VLM and a pre-trained text decoder [17, 48, 61], or train a decoder from scratch [6, 12, 31, 56] using curated captioning datasets such as MS COCO [10], CC3M [55], and Visual Genome [35].

A recent line of research on zero-shot image captioning or retrieval with VLMs proposes text-only training to eliminate the need for curating expensive image datasets. As text and images are encoded into disjoint regions of the VLM embedding space, additional mechanisms to bridge the modality gap have been proposed, like injecting noise into the text embeddings [20, 22, 50] or projecting the image embeddings [41]. These methods face two significant disadvantages—first, captions often fail to capture key image aspects or identify core objects, and second, they are biased towards the limited set of object classes contained in their training dataset. NOVIC adopts text-only training, but focuses on predicting truly open vocabulary object labels.

3. Methodology

An overview of the proposed training and inference schemes for NOVIC is shown in Fig. 2. A text-only dataset

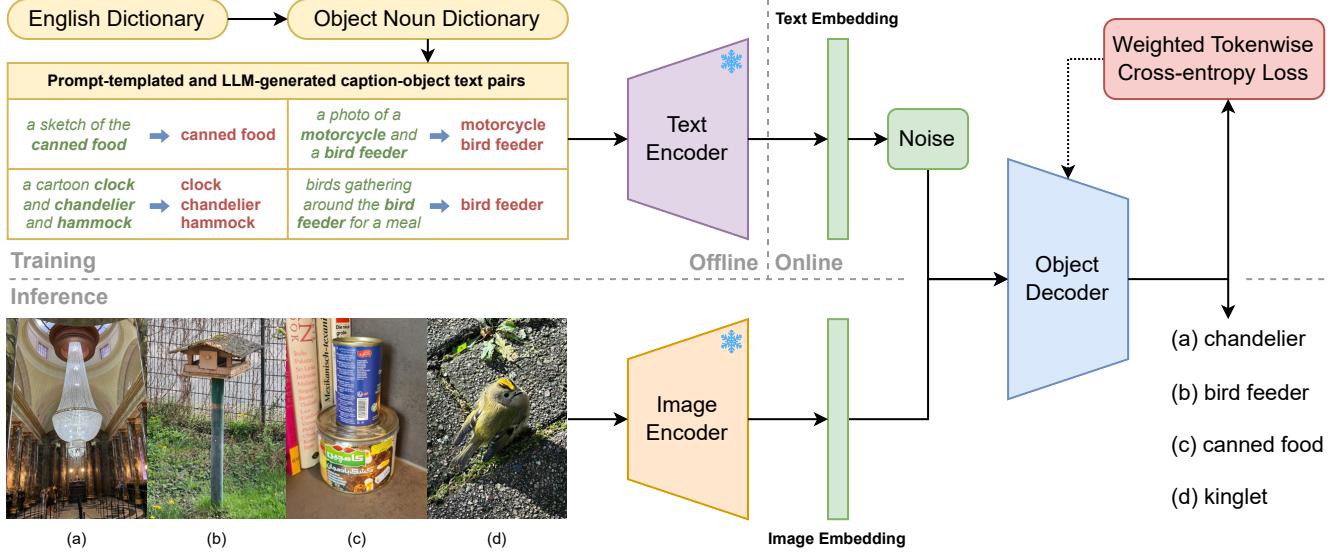


Figure 2. **Overview of the NOVIC training and inference schemes.** A dataset of caption-object text pairs is generated offline using an English dictionary, prompt templates, and LLM-based caption generation. The captions are encoded offline into text embeddings, and augmented with noise to train the object decoder. The training scheme seamlessly generalizes across the large modality gap between image and text embeddings, allowing inference of arbitrary images via the image encoder. Produced classifications can be very fine-grained.

mapping caption sentences to target object nouns is synthetically generated offline (cf. Sec. 3.2). The text embedding vectors of the captions are then precomputed using the text encoder of a frozen CLIP model. These vectors are augmented online with unit-norm-preserving noise, and used to train a decoder-only transformer, the *object decoder* (cf. Sec. 3.1), to generate the object noun corresponding to each caption. The output object nouns are generated in the form of free language using the same text tokenizer as the frozen text encoder, with no definition of candidate object nouns being provided to the decoder in any way. At inference time, the object decoder seamlessly generalizes to image embedding vectors calculated using the image encoder of the frozen CLIP model, despite CLIP models generally having very large modality gaps [44]. The performance of the decoder is measured on image classification datasets and annotated open vocabulary image datasets (cf. Sec. 3.3).

3.1. Object Decoder Model

The architecture of the object decoder model is shown in Fig. 3. The model is built around an autoregressive decoder-only language transformer [62] that has been configured to use learned token embeddings, weight tying [52], learned positional embeddings, pre-layer normalization [68], GELU activation functions [26], beam search, custom weight initialization (cf. App. A), a contractive inner feed-forward dimension, and no biases in the linear and layer normalization layers, including the final token logits linear layer. Dropout is applied after the positional embedding and within the transformer layers. The input embedding vector is sup-

plied to the transformer in the form of P prefix token vectors, *i.e.* P dynamically calculated vectors of dimension H (hidden dimension). The prefix tokens are computed from the input embedding using a learned linear projection that has PH outputs and no biases, as the immediately following positional embedding makes biases redundant. The tokenized target object nouns do not require a start token, as the first output token is predicted at the sequence location of the last prefix token. Appropriate sequence masking is also used to ensure that only sequence locations up to the first output `<eos>` contribute to the loss. The internal attention mask of the transformer is causal, with the exception that the P prefix tokens can attend to each other without restriction, allowing them to share information freely.

3.2. Training Dataset for Text-Only Supervision

The object decoder needs to be trained on a dataset of associative caption-object text pairs (cf. Fig. 2). One could attempt to generate such a dataset from a large-scale image-text web dataset ranging into the billions of samples, but it is not a simple task to accurately and sufficiently efficiently construct the required ground truth object noun annotations, especially in such a way that a fair coverage of the English language is achieved. We instead *invert* the chosen CLIP text encoder by synthetically generating captions that explicitly cover the entire English language of object nouns, and use this to systematically sample in the CLIP embedding space essentially all possible object concepts, and learn to map them to their corresponding textual representations.

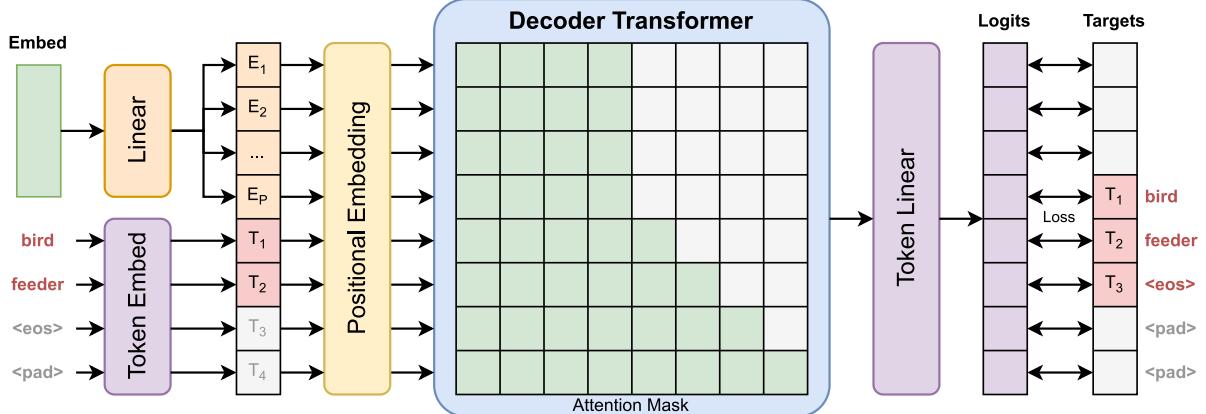


Figure 3. Architecture of the object decoder. The decoder-only transformer is given a sequence of linearly projected embedding vector tokens (full attention) followed by autoregressive object noun tokens (causal attention). The output sequence is converted to token logits using a linear layer that is weight-tied to the token embeddings. Cross-entropy loss with left-shifted target tokens is used during training.

Object noun dictionary. An exhaustive dictionary of simple and compound object nouns, referred to as the *object noun dictionary*, was created based on the WordNet dictionary [47], GNU Collaborative International Dictionary of English [9], and the categories used by various image datasets (cf. App. B). Noun entries were cleaned, collected and merged as appropriate to account for word duplication, plurality, inflection and alternate spellings, and filtered down to only include nouns that correspond to suitable object classifications (*e.g.* no abstract nouns). This resulted in 42 919 unique case-normalized object nouns corresponding to 96 526 noun variants. Each object noun was then assigned a log-scale frequency value based on its observed frequency in the English language as per the n-gram frequencies in the Google Web Trillion Word Corpus [8, 49]. Object nouns that are categories in a curated, comprehensive collection of image datasets were given appropriate frequency boosts. The log-scale frequencies were scaled and rounded so that very rare nouns saturate at a value of 1, and very frequent nouns have values ≥ 12 . Aside from being used to adjust relative sampling frequencies during training, the resulting frequencies allowed the construction of object noun dictionary subsets by filtering out nouns less or equal to a particular frequency threshold (FT), *e.g.* FT2.³

Prompt templates. A direct and extremely cheap way of generating caption-object text pairs is by using exactly the same prompt templates as is usually used for zero-shot CLIP classification. This maps captions like “*a blurry photo of a bottle*” to the object noun “*bottle*”. Adjusting the ImageNet-1K and CIFAR prompts used by [53] to account for pluralization and allow for correct indefinite article use leads to 85 unique prompt templates for singular forms of

³FT0, FT2, FT6, and FT9 have 42 919, 11 897, 5 899, and 2 919 unique object nouns, respectively. FT0 is the complete object noun dictionary.

object nouns, and 68 for plural forms. Combined with the full object noun dictionary, and including repetitions based on the log-scale frequencies of the nouns leads to 13.4M samples in the dataset. The mean observed object noun frequency is 3.5, but the true mean number of repetitions is much smaller at only 1.5, due to the use of the available alternate spellings during prompt templating.

Multiset prediction. Image embeddings can capture significantly more complicated dynamics than just the main entity present. If there is a photo of a lion with a zookeeper, for instance, the image embedding will likely contain some features corresponding to *lion*, *person*, *fence*, and potentially more abstract concepts like *zoo* as well. To help also capture these dynamics in the text embeddings seen during training, multi-target samples are introduced. The complete object noun dictionary, including explicit repetition of alternate spellings based on their individual log-scale frequencies of mean 1.5, is randomly shuffled m times into m separate lists. These lists are ‘zipped’ to form m -tuples of object nouns that are then used to generate prompt-templated captions exactly as before. The m object nouns per caption are linked by ‘and’ (cf. Fig. 2), and all m nouns use a target weight of $\frac{1}{m}$. This produces a dataset of 13.4M further samples for each value of m , referred to as the \mathcal{M}_m multiset in each case (*e.g.* \mathcal{M}_3 for $m = 3$). In total, a combined dataset of \mathcal{M}_1 ⁴ and the two multisets \mathcal{M}_2 and \mathcal{M}_3 contains 40.2M samples with 80.4M targets. We qualitatively observed that multiset training specifically helped the open vocabulary classification of images that prominently contain multiple different elements or objects. On an intuitive level, the use of multi-target samples also prevents the object decoder from being able to just ‘rote-memorize’ which local regions of the text embedding space correspond to which pure object con-

⁴ \mathcal{M}_1 is equivalent to the prompt-templated dataset as initially described.

cepts, and instead forces the decoder to learn *which* values in *which* parts of the embedding vector are actually associated with *which* object concept. Such knowledge is essential for proper generalization to image embeddings.

Caption data. In order to supply the object decoder with even more diverse synthetic training data to generalize from, captions were generated for each noun in the object noun dictionary using an LLM (cf. App. C). This approach leverages the implicit contextual knowledge of LLMs to place object nouns in *relevant* contexts—incorporating appropriate adjectives, actions, scenes, and secondary objects—in a style resembling real image captions.⁵ The minimum number of unique generated captions for each noun variant was set proportionally to the corresponding log-scale frequency (up to 100) in order to match the relative word frequencies found in the English language. OpenAI GPT-3.5 Turbo was selected as an ideal balance of cost and sufficient performance, and was used to generate a total of 1.8M captions. Ensuring frequency-appropriate use of alternate spellings, this resulted in a total of 2.9M caption samples (2.0M unique) that were then oversampled to around 30% the size of any multiset they were merged with. For the combined dataset \mathcal{M}_{1-3} this resulted in 51.7M samples with 91.9M targets.

Noise augmentation. In order to allow more robust and generalizable features of the CLIP embedding space to be learned, data augmentation in the form of noise is used during training. Due to the large modality gap between text and image embeddings [44], noise is in fact a crucial component of the training scheme. Noise allows the object decoder to learn to be invariant to even very large changes in the non-object-related parts of the text embedding vectors, which helps the generalization to image embeddings, as these are very sensitive to secondary objects, backgrounds, and even subtle visual variations. Gu et al. [22] used elementwise Gaussian noise with vector renormalization to train a captioning model on text embeddings only, but did not entirely accurately explain its effect. The CLIP embedding space is deceptively and unimaginably vast⁶, so the random noise statistically *cannot* be significantly filling the gaps between samples, or causing a meaningful overlap with image embedding vectors. The main benefit of noise is that the model can learn to focus on which parts (*i.e.* subspaces) of the embedding vector really encode which individual object concepts, instead of focusing on which region of the embedding space a vector is in as a whole. We observe that the amount of noise required for best training results is actually quite large, not small or minor [22], and its effect is to completely scatter

⁵CLIP embedding spaces are *trained* on real image captions.

⁶Even if we assumed each element of a 768-dim embedding vector only contains a *single* bit of information, this would still lead to 10^{231} different embeddings, against which even a trillion samples is essentially zero.

the cone [44] of text embeddings to a thin hyperannulus on the order of 66–80° away, with no single sample remaining significantly closer than that to any original text embedding (cf. App. D). By comparison, the very large modality gap between the image and text embedding cones can also be measured, and is around 70–85° for *matching* image-text pairs (depending on the CLIP model), and only 8–12° more than this for completely uncorrelated image-text pairs. As indicated previously though, statistically there is negligible region overlap between the scattered hyperannulus and the image embeddings cone due to the vast nature of the embedding space. The large amount of noise required can instead be seen as a trade-off between the disruptive ‘blurring’ effect of noise, and the benefit of the object decoder having seen many embedding vectors during training that are as far removed from text embeddings as image embeddings are. In this work, we use an 85% to 15% mix of elementwise Gaussian noise and uniform angle noise. The latter was developed to beneficially cover a greater range of noise dynamics and levels, by rotating input text embedding vectors uniformly in random directions by angles in the uniform 45–75° range.

3.3. Evaluation Datasets for Zero-Shot Recognition

One way to evaluate how much NOVIC has learned is by examining its zero-shot performance on standard image classification datasets like ImageNet-1K and Food-101. Context is everything for image classification datasets however. Knowing which categories are available in a dataset serves as a *strong* prior for supervised models. For instance, if a dataset contains only one bird category, such as *kinglet*, a supervised model only needs to determine whether an image is a bird to confidently classify it as a kinglet. For prompt-free open vocabulary models like NOVIC however, the task is more challenging. When presented with an image that happens to come from Food-101, it is not clear that the model should suddenly focus on the specific dish shown instead of potentially more prominent or general objects in the image. Consequently, to meaningfully evaluate NOVIC on image classification datasets, we restrict the generated beam search text tokens in every forward pass of the object decoder to those consistent with the defined category names, and score whether the correct final category is identified.

As NOVIC is trained without providing explicit categories, we wish to evaluate it also on datasets that do not specify categories. No suitable such dataset is known however, as such a dataset would need to have annotations for every possible correct object noun in the English language for every visible entity in every part of each image. To address this, three open vocabulary image datasets were curated as part of this work (cf. App. E), and individually annotated by both human and multimodal LLM annotators (cf. App. F) for the object nouns that were predicted by the trained models. The annotations specify whether each clas-

sification is *correct*, *close*, or *incorrect*, and for the human annotations, whether it relates to a *primary* or *secondary* element of the image. The three new datasets are (i) World: 272 images of which the grand majority are originally sourced (have never been on the internet) from 10 countries by 12 people, with an active focus on covering as wide and varied concepts as possible, including unusual, deceptive and/or indirect representations of objects, (ii) Wiki: 1 000 Wikipedia lead images sampled from a scraped pool of 18K, (iii) Val3K: 3 000 images from the ImageNet-1K validation set, sampled uniformly across the classes (LLM annotations only). The suffixes -H and -L are used to clarify whether human or LLM annotations are in use, *e.g.* Wiki-H. A total of 17.4K human and 112K LLM (OpenAI GPT-4o) annotations were made across the three datasets. This was at the limit of feasibility in terms of number of samples, person-hours and cost.

4. Experiments

For experiments, the SigLIP ViT B/16 model [74] is used as the base CLIP model, due to its favorable balance of speed and performance, especially when compared to the original CLIP models [53]. Training is performed for 18 epochs with the AdamW optimizer, cosine learning rate decay, weight decay on all multidimensional weights, and a gradient-accumulated batch size of $B = 8192$. Hyperparameter details can be found in App. G. The object decoder model is quite small at only 12.2M parameters, and takes less than 3 days to train on a *single* RTX A6000. For FT2, this drops to 11.4M parameters and only 1.5 days. Overall, this is *extremely* efficient, considering that up to a billion samples are being seen by a single GPU. As a scale comparison, using our method on the very same GPU to train an object decoder on all 1.28M ImageNet-1K samples for 18 epochs takes a mere 19 minutes, and results in an impressive top-1 accuracy of 88.21%. The object decoder is also very resource efficient at inference time. Requiring at most 5 GB GPU memory, the total pipeline—including CLIP model, autoregression, and beam search—takes on average 26 ms per image for single images, and 7 ms per image when using batching ($B = 256$), making it very real-time capable.

Open vocabulary scoring. Model predictions on the three introduced open vocabulary image datasets are scored by assigning $\{1, 0.8, 0.5, 0.4, 0\}$ points for $\{\text{correct primary}, \text{correct secondary}, \text{close primary}, \text{close secondary}, \text{incorrect}\}$ predictions respectively, and expressing the sum as a percent of the maximum possible score. For verification purposes, the prediction scores achieved by many models were statistically compared between human and LLM annotations on the same dataset and found to be suitably correlated (cf. App. F), thus allowing model performance insights to be gained even when manual annotation is just too prohibitive. It is expected that LLM prediction scores are lower than

Dataset	CLIP	FT9	FT6	FT2	FT0
ImageNet-1K [54]	75.91	46.76 [†]	69.50	68.11	60.56
Tiny ImageNet [38]	59.77	54.53 [†]	57.36	55.83	53.78
Imagenette [16]	99.59	99.44	99.27	99.32	99.23
Imagewoof [16]	93.28	93.07	92.07	92.20	90.15
ImageNet-A [27]	45.05	35.33 [†]	42.43	41.42	38.23
ImageNet-R [27]	90.24	74.46 [†]	88.01	86.62	83.43
Food-101 [7]	91.55	85.25	83.24	82.73	68.80
CIFAR-10 [36]	92.33	91.08	90.53	90.77	90.92
CIFAR-100 [36]	72.19	70.65	69.93	69.43	67.80

Table 1. **Performance on image classification benchmarks (%) mean of 3).** [†]FT9 has disadvantaged scores on some datasets due to the seen object nouns not covering all the classes (cf. App. J).

Annotations	FT9	FT6	FT2	FT0
ImageNet-1K	46.52	69.47	67.92	60.72
Open (correct)	70.33	75.04	72.90	66.19
Open (correct + close)	71.90	76.50	74.35	67.86

Table 2. **Performance on Val3K dataset (%) mean of 3),** showing that category-constrained scores are a significant underestimation of the true open vocabulary performance of NOVIC.

human ones due to some residual LLM mistakes, a generally narrow acceptance of classifications, and inherently less frequent use of the *close* annotation by the LLM.

4.1. Zero-Shot Classification Performance

Image classification benchmarks. Tab. 1 shows the zero-shot performance of NOVIC on various image classification benchmarks. The performance of the underlying CLIP model is also shown as a measure of how completely the vast knowledge of the CLIP model has been captured by the object decoder and converted into generative textual understanding. The decoder relies on the organization of the embedding space provided by the CLIP model, and so naturally is not expected to outperform it in this category-prompted setting. Overall, NOVIC shows competitive top-1 accuracies, with some ‘distillation’ losses being observed for tendentially finer-grained classification datasets and lower frequency thresholds. Classification benchmarks, however, do not tell the whole story. The act of restricting the object decoder to a fixed vocabulary *forces* it to choose object nouns it does not even consider to be best. In Tab. 2, we evaluate the top-1 accuracy on Val3K using the original ImageNet-1K annotations, and compare this to the percent of open vocabulary classifications on the very same images that are *correct* as per Val3K-L. Prediction scores that include points for *close* predictions are also shown. The results demonstrate that the object decoder’s knowledge is more accurately assessed when used for unconstrained open vocabulary classification.

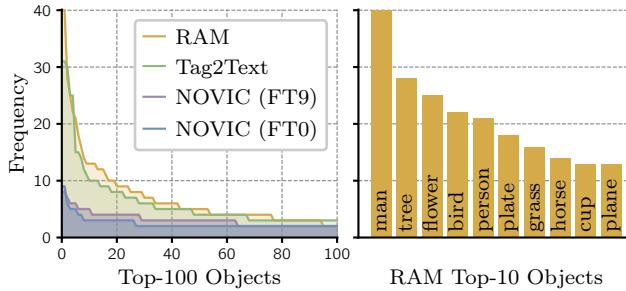


Figure 4. Diversity comparison of predicted object nouns. Plots showing the sorted top frequency counts of the object nouns predicted for Wiki-H. Even when trained on FT9, NOVIC shows much greater diversity and has a less peaked frequency distribution. **Right:** The top-10 nouns for RAM are generic and overused.

Comparison with related work. As discussed in Sec. 2, no directly equivalent state-of-the-art model is known that performs real-time prompt-free open vocabulary image classification. The most suitable models for comparison are image tagging models like Tag2Text [29] and the Recognize Anything Model (RAM) [76], as they are designed to be able to provide text labels for images of arbitrary content. We evaluate the largest model variants by taking the highest scoring prediction for each image, and compare this to NOVIC in Tab. 3 and App. K. While the tagging models at first glance numerically seem to obtain a reasonable *Prediction Score*, the comparatively low *Primary Scores* indicate that the models, unlike NOVIC, make a large portion of their score from *secondary* or *close* predictions, as opposed to *correct primary* ones (a deficiency clearly illustrated in Tab. 7). Also, when accounting for prediction specificity, the resulting *Overall Score* clearly demonstrates the superiority of NOVIC by up to 19% absolute. A closer look reveals that the low scores of the tagging models are due to a severe lack of diversity and specificity (cf. Fig. 4 and qualitative results in App. H). The produced tags are tendentially coarse and repetitive, with around a third of all images receiving one of only 20 generic top labels, often missing the main point of images. Even when provided with the full object noun dictionary, the total diversity of labels stays below 400, with over 80% of predictions still being made from the small core RAM vocabulary. This is in contrast to our model, which has a much flatter prediction distribution, diversities up to 934 (cf. Tab. 3), and a remarkably high prediction score despite generating *fine-grained* classifications. The tags produced by Tag2Text and RAM also tend to be uninformative—when applied to the images of the Imagewoof [16] validation set for example, both models simply output *dog* 88% of the time, and never a single dog breed. Our FT0 model, on the other hand, predicts over 90 different dog breeds, with much of the weight covering variations of the actual ground truth classes. When RAM is evaluated on ImageNet-1K by only providing it the 1000 class names to choose from, it only scores 38.3%

Model	Seen Objects	Used Objects	RAM Vocab	Primary Score	Prediction Score	Overall Score
Tag2Text [29]	3 429	397	89.0	69.15	81.35	62.61
RAM [76]	4 585	348	100.0	65.05	80.73	59.98
RAM (FT2)	11 897	391	83.4	62.30	77.62	60.58
RAM (FT0)	42 919	399	81.7	61.70	76.94	60.25
NOVIC (FT9)	2 919	693	64.0	68.55	72.35	69.08
NOVIC (FT6)	5 899	794	49.3	69.90	72.70	71.07
NOVIC (FT2)	11 897	846	39.7	68.65	72.53	71.88
NOVIC (FT0)	42 919	852	37.5	68.70	72.34	71.31
NOVIC [†] (FT2)	11 897	893	36.3	77.05	80.13	79.02
NOVIC [†] (FT0)	42 919	934	28.7	77.10	79.18	78.21

Table 3. Open vocabulary Wiki-H results in comparison to baselines (best of 3). Tag2Text and RAM achieve artificially high scores by frequently outputting uninformative generic tags like *man*, *tree*, *grass*, *plate*, and not for instance what is *on* the grass or *plate*, or what *type* of tree it is, or what the man is *using*. Scores are in %. *RAM Vocab* is the percent of samples whose predictions are in the core RAM vocabulary. *Primary Score* is the *Prediction Score* when rewarding only primary predictions. *Overall Score* is the *Prediction Score* with $\times 0.5$ applied to the scores of coarse predictions. See App. K. [†]Using DFN-5B H/14-378 CLIP model.

(25.4% for non-core vocabulary), which is *far* less than the 75.54% of the ViT-L CLIP model it was trained on, and *far* less than the 60–70% nominally scored by NOVIC.

4.2. Ablation Studies

Tab. 4 shows the effect on decoder performance of varying the dataset configuration, training noise, and underlying CLIP model. FT2 was chosen as the ablation baseline, as many ablation runs were required, and FT2 strikes a balance between overall feasibility of training times, and ensuring that a diverse and expressive range of object nouns are seen during training. Further ablations are presented in App. I. Overall, as expected, the human annotations exhibit generally higher scores and are slightly more consistent, due to human checking and more prevalent use of the *close* category. The decoder ImageNet-1K scores are also observed to be well-correlated with general open prediction performance.

Training dataset. The effect of the object noun dictionary frequency threshold (cf. Sec. 3.2) on decoder classification performance is shown in Tab. 4. The prediction scores grow as the training dataset size is increased, until a limit is reached between FT2 and FT0 where the exploding diversity of seen object nouns saturates the amount of order the CLIP model can provide in the embedding space. In Tab. 4, we systematically analyze the contribution of each component of the training set. Starting with M_1 and incrementally building the nominal training set (*i.e.* $M_1 + M_2 + M_3 + \text{LLM}$ captions), we observe consistent gains. Immediate losses are seen if the LLM captions are removed, or if the lower-order M_1 and M_2 multisets are removed. Further tests with

Configuration	World-H	Wiki-H	Wiki-L	IN1K
FT9	76.13	71.20	63.57	46.76
FT6	78.42	72.33	62.62	69.50
FT2*	78.92	72.03	63.97	68.11
FT0	73.98	71.48	61.27	60.56
\mathcal{M}_1 dataset	70.11	-	50.05	63.88
+ LLM captions	72.16	-	55.37	66.06
+ \mathcal{M}_2 dataset	78.37	-	61.60	67.97
+ \mathcal{M}_3 dataset*	78.92	72.03	63.97	68.11
$\mathcal{M}_1 + \mathcal{M}_2 + \mathcal{M}_3$	77.56	71.60	62.08	65.36
\mathcal{M}_3 + LLM captions	77.88	71.28	62.93	61.12
Gauss 1.00	19.44	-	15.62	13.36
Gauss 1.75	70.37	-	52.07	49.23
Gauss 2.50	76.26	-	62.10	67.94
Gauss 3.25	77.57	70.95	62.38	66.46
+ 15% Uniform 45–75*	78.92	72.03	63.97	68.11
Uniform 45–75	76.46	-	63.27	64.75
DataComp-1B B/16 [18]	76.85	-	63.55	64.17
SigLIP B/16* [74]	78.92	72.03	63.97	68.11
DataComp-1B L/14 [18]	84.27	-	70.35	70.05
SigLIP SO/14 [74]	87.49	-	70.28	74.85
DFN-5B H/14-378 [15]	87.08	-	74.88	75.82
→ FT0	86.41	-	74.48	75.68

Table 4. **Dataset, training noise, and CLIP model ablations (% mean of 3).** IN1K is the ImageNet-1K classification performance of the trained decoders, and the asterisks correspond to the nominal ablation baseline. See App. I for more ablation results. Some Wiki-H scores are unavailable due to the high cost of manual annotation.

variations in training noise configuration also validate that very large amounts of Gaussian noise are *required* in order to produce models that can generalize across the modality gap,⁷ and that instead using uniform angle noise on 15% of the samples produces the best seen configuration.

CLIP model. The proposed NOVIC training scheme scales well with the performance of the underlying CLIP model. Tab. 4 demonstrates that progressively stronger CLIP models yield progressively stronger object decoders. Remarkably however, the training times remain almost unchanged, as text encoders only scale slowly with CLIP model strength and the datasets of text embedding vectors are generated offline and only once anyway (cf. Fig. 2). It can be observed from Tab. 4 that the performance of the FT0 object decoder that was trained on the strong DFN-5B CLIP model is only marginally below the corresponding FT2 performance. This is much closer than for the nominal CLIP model, and can be attributed to the improved order and separation of concepts that the CLIP model provides in the embedding space. Based on the World-H results, it is actu-

⁷Gauss 3.25 in the table refers to elementwise Gaussian noise with a standard deviation so high that the expected norm of the noise vector that is added to the unit-norm embedding is 3.25, *i.e.* many times its magnitude.

ally expected that the DFN-5B FT0 model can provide very fine-grained *correct primary* classifications for up to 84.2% of pictures it is given. Having trained on the full unfiltered object noun dictionary, the model is frequently able to make correct classifications as detailed as *red velvet cake*, *brazil nut tree*, *electron microscope*, and *egretta caerulea*.

Language generalization. During training, the object decoder develops an implicit understanding of language and its relationship to object concepts, demonstrating qualitative evidence that it can occasionally use this knowledge to generalize beyond the dataset. For instance, the unseen noun *New Zealand willow* was predicted for a scenic photo of a willow tree that was indeed taken in New Zealand. The object decoder also inherits CLIP’s slight tendency to be able to ‘read’, as evidenced for example by a generic blue plushie being correctly classified by multiple decoders as *genus staphylococcus*, but not anymore if the small visible text *Staphylokokke* is removed, falling back to *stuffed toy*. In another case, a sign reading *Aral* is classified as *ara* (a macaw), indicating that the decoder can associate certain features in the embedding space with certain spellings. As another form of language generalization, the object decoder can easily be trained to output high-performing predictions in any chosen language—even low resource languages for which insufficient image-text data exists to train a CLIP model. By simply applying dictionary or machine translation to the target object nouns in the \mathcal{M}_m multisets, a predominantly English CLIP model can be used to directly train a strong object decoder in any translatable language.

5. Conclusion

In this paper, we address the problem of unconstrained real-time open vocabulary image classification, and highlight the limitations of current state-of-the-art methods in accomplishing this task. To overcome these limitations, we propose an autoregressive object decoder trained on a large-scale text-only dataset. The decoder generates object nouns directly as language on the basis of CLIP embeddings, and crucially, does not require any class candidates to be provided. To create the text-only training dataset, we introduce a comprehensive object noun dictionary and employ multi-set prompt-template in combination with LLM-generated captions and a pivotal noise augmentation strategy to effectively bridge the modality gap from text to images. Our approach demonstrates promising results on both established and newly proposed in-the-wild classification datasets. We lift the common limitation of zero-shot image recognition models regarding prompts and class candidates, ultimately allowing agents to immediately recognize and label *any* object in dynamic environments. Future work can include providing an input to the decoder that controls the level of semantic granularity used in the predicted output nouns.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, et al. Flamingo: a Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022. [2](#)
- [2] Philipp Allgeuer. Object Noun Dictionary generation source code. https://github.com/pallgeuer/object_noun_dictionary. [2](#)
- [3] Philipp Allgeuer and Kyra Ahrens. NOVIC: Unconstrained open vocabulary image classification. <https://github.com/pallgeuer/novic> and <https://paperswithcode.com/paper/unconstrained-open-vocabulary-image>. [2](#)
- [4] Philipp Allgeuer and Kyra Ahrens. NOVIC: Unconstrained open vocabulary image classification. <https://www.inf.uni-hamburg.de/en/inst/ab/wtm/research/corpora.html#novic> and <https://paperswithcode.com/dataset/novic-caption-object-data>. [2](#)
- [5] Philipp Allgeuer and Kyra Ahrens. Open vocabulary image classification datasets. <https://www.inf.uni-hamburg.de/en/inst/ab/wtm/research/corpora.html#ovic-datasets> and <https://paperswithcode.com/dataset/ovic-datasets>. [2](#)
- [6] Manuele Barraco, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. The Unreasonable Effectiveness of CLIP Features for Image Captioning: An Experimental Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4662–4670, 2022. [2](#)
- [7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014. Springer International Publishing. [6](#)
- [8] Thorsten Brants and Alex Franz. Google web trillion word corpus: Web 1T 5-gram version 1. <https://catalog.ldc.upenn.edu/LDC2006T13>. [4, 15](#)
- [9] Patrick Cassidy et al. GNU Collaborative International Dictionary of English. <https://gcide.gnu.org.ua>. [4, 15](#)
- [10] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server, 2015. arXiv:1504.00325. [2](#)
- [11] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. YOLO-World: Real-Time Open-Vocabulary Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16901–16911, 2024. [2](#)
- [12] Marcella Cornia, Lorenzo Baraldi, Giuseppe Fiameni, and Rita Cucchiara. Generating More Pertinent Captions by Leveraging Semantics and Style on Multi-Source Datasets. *International Journal of Computer Vision*, 2023. [2](#)
- [13] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning To Prompt for Open-Vocabulary Object Detection With Vision-Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14084–14093, 2022. [2](#)
- [14] Sedigheh Eslami, Christoph Meinel, and Gerard de Melo. PubMedCLIP: How Much Does CLIP Benefit Visual Question Answering in the Medical Domain? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1181–1193, Dubrovnik, Croatia, 2023. Association for Computational Linguistics. [2](#)
- [15] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data filtering networks. In *12th International Conference on Learning Representations*, 2024. [8](#)
- [16] fast.ai. Imagenette GitHub. <https://github.com/fastai/imagenette>. [6, 7](#)
- [17] Junjie Fei, Teng Wang, Jinrui Zhang, Zhenyu He, Chengjie Wang, and Feng Zheng. Transferable Decoding with Visual Entities for Zero-Shot Image Captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3136–3146, 2023. [2](#)
- [18] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, et al. DataComp: In search of the next generation of multimodal datasets. In *37th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. [8](#)
- [19] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One Embedding Space To Bind Them All. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2023. [2](#)
- [20] Geonmo Gu, Sanghyuk Chun, Wonjae Kim, Yoohoon Kang, and Sangdoo Yun. Language-only Training of Zero-shot Composed Image Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13225–13234, 2024. [2](#)
- [21] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A Systematic Survey of Prompt Engineering on Vision-Language Foundation Models, 2023. arXiv:2307.12980. [2](#)
- [22] Sophia Gu, Christopher Clark, and Aniruddha Kembhavi. I Can’t Believe There’s No Images! Learning Visual Tasks Using only Language Supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2672–2683, 2023. [1, 2, 5](#)
- [23] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *International Conference on Learning Representations*, 2022. [2](#)
- [24] Andrey Guzhov, Federico Raue, Jorn Hees, and Andreas Dengel. Audioclip: Extending Clip to Image, Text and Audio. In *ICASSP 2022 - 2022 IEEE International Conference on*

- Acoustics, Speech and Signal Processing (ICASSP)*, pages 976–980, Singapore, Singapore, 2022. IEEE. 2
- [25] Wenbin He, Suphanut Jamonnak, Liang Gou, and Liu Ren. CLIP-S4: Language-Guided Self-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11207–11216, 2023. 2
- [26] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3
- [27] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *Computer Vision and Pattern Recognition (CVPR)*, 2021. 6
- [28] Jingjia Huang, Yinan Li, Jiashi Feng, Xinglong Wu, Xiaoshuai Sun, and Rongrong Ji. Clover: Towards a Unified Video-Language Alignment and Fusion Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14856–14866, 2023. 2
- [29] Xinyu Huang, Youcai Zhang, Jinyu Ma, Weiwei Tian, Rui Feng, Yuejie Zhang, Yaqian Li, Yandong Guo, and Lei Zhang. Tag2Text: Guiding Vision-Language Model via Image Tagging. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 7, 23
- [30] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 2021. 2
- [31] Wooyoung Kang, Jonghwan Mun, Sungjun Lee, and Byungseok Roh. Noise-Aware Learning from Web-Crawled Image-Text Data for Image Captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2942–2952, 2023. 2
- [32] Prannay Kaul, Weidi Xie, and Andrew Zisserman. Label, Verify, Correct: A Simple Few Shot Object Detection Method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14237–14247, 2022. 2
- [33] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but Effective: CLIP Embeddings for Embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14829–14838, 2022. 2
- [34] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2
- [35] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 2
- [36] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. PhD thesis, University of Toronto, 2009. 6
- [37] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning Segmentation via Large Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9579–9589, 2024. 2
- [38] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS 231N, Stanford University*, 2015. 6
- [39] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal Foundation Models: From Specialists to General-Purpose Assistants, 2023. arXiv:2309.10020. 2
- [40] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 2022. 2
- [41] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. DeCap: Decoding CLIP Latents for Zero-Shot Captioning via Text-Only Training. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2
- [42] Xianhang Li, Zeyu Wang, and Cihang Xie. CLIPAv2: Scaling CLIP Training with 81.1% Zero-shot ImageNet Accuracy within a \$10,000 Budget; An Extra \$4,000 Unlocks 81.8% Accuracy, 2023. arXiv:2306.15658. 2
- [43] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-Vocabulary Semantic Segmentation With Mask-Adapted CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7061–7070, 2023. 2
- [44] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 17612–17625. Curran Associates, Inc., 2022. 2, 3, 5
- [45] Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang. Open-World Semantic Segmentation via Contrasting and Clustering Vision-Language Embedding. In *Computer Vision – ECCV 2022*, pages 275–292, Cham, 2022. Springer Nature Switzerland. 2
- [46] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 2
- [47] George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995. 4, 15
- [48] Ron Mokady, Amir Hertz, and Amit H. Bermano. ClipCap: CLIP Prefix for Image Captioning, 2021. arXiv:2111.09734. 2
- [49] Peter Norvig. Natural language corpus data: Beautiful data. <http://norvig.com/ngrams>. 4, 15
- [50] David Nukrai, Ron Mokady, and Amir Globerson. Text-Only Training for Image Captioning using Noise-Injected CLIP. In

- Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4055–4063, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. 2
- [51] Chau Pham, Truong Vu, and Khoi Nguyen. LP-OVOD: Open-Vocabulary Object Detection by Linear Probing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 779–788, 2024. 2
- [52] Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2016. 3
- [53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 1, 2, 4, 6
- [54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 6
- [55] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, 2018. Association for Computational Linguistics. 2
- [56] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How Much Can CLIP Benefit Vision-and-Language Tasks? In *International Conference on Learning Representations*, 2022. 2
- [57] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and Where Pathways for Robotic Manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 894–906. PMLR, 2022. 2
- [58] Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. CLIP Models are Few-Shot Learners: Empirical Studies on VQA and Visual Entailment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6088–6100, Dublin, Ireland, 2022. Association for Computational Linguistics. 2
- [59] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved Training Techniques for CLIP at Scale, 2023. arXiv:2303.15389. 2
- [60] Guy Tevet, Brian Gordon, Amir Hertz, Amit H. Bermano, and Daniel Cohen-Or. MotionCLIP: Exposing Human Motion Generation to CLIP Space. In *Computer Vision – ECCV 2022*, volume 13682, pages 358–374. Springer Nature Switzerland, Cham, 2022. Series Title: Lecture Notes in Computer Science. 2
- [61] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. ZeroCap: Zero-Shot Image-to-Text Generation for Visual-Semantic Arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17918–17928, 2022. 2
- [62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3
- [63] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust Fine-Tuning of Zero-Shot Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971, 2022. 2
- [64] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2CLIP: Learning Robust Audio Representations from Clip. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567, Singapore, Singapore, 2022. IEEE. 2
- [65] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards Open Vocabulary Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2024. 1
- [66] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Wentao Liu, and Chen Change Loy. CLIM: Contrastive Language-Image Mosaic for Region Representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6117–6125, 2024. 2
- [67] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. CORA: Adapting CLIP for Open-Vocabulary Detection With Region Prompting and Anchor Pre-Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7031–7040, 2023. 2
- [68] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML’20. JMLR.org, 2020. 3, 13
- [69] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 2
- [70] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-Vocabulary Panoptic Segmentation With Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2955–2966, 2023. 2

- [71] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP: Learning a Unified Representation of Language, Images, and Point Clouds for 3D Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1189, 2023. 2
- [72] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, et al. Florence: A New Foundation Model for Computer Vision, 2021. arXiv:2111.11432. 2
- [73] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-Vocabulary Object Detection Using Captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14393–14402, 2021. 2
- [74] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023. 2, 6, 8
- [75] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. PointCLIP: Point Cloud Understanding by CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8552–8562, 2022. 2
- [76] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, Yandong Guo, and Lei Zhang. Recognize Anything: A Strong Image Tagging Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1724–1732, 2024. 1, 2, 7, 23
- [77] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. RegionCLIP: Region-Based Language-Image Pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16793–16803, 2022. 2

Appendix

A. Object Decoder Weight Initialization

Various state-of-the-art transformer models like BERT, OpenCLIP and GPT use heuristics for weight initialization, for example by initialising weights to a zero-mean normal distribution of standard deviation 0.02 or $\frac{1}{\sqrt{H}}$, for H the hidden dimension. Guided by the principles of mean field theory, we rigorously calculate weight initialization standard deviations such that the expected forward pass norms are statistically preserved throughout the object decoder, and such that the overall contributions of the main and residual pathways are balanced, avoiding excessive input signal dilution. This comes with numerical and gradient stability benefits, leading to fast and robust initial convergence.

A.1. Strategies for Strong Initial Convergence

In general, the rule-of-thumb aims of the presented *variance-balanced weight initialization strategy* for the forward pass of a network are:

- **Avoid input signal dilution.** For example, if the residual pathway of the input through the network involves repeatedly combining it with values calculated from random weights and renormalizing, then the ‘proportion’ of the final output signal that actually corresponds statistically to the input signal can be exponentially microscopic, making initial convergence more difficult.
- **Avoid combining unbalanced scales of data.** If two signals are added or concatenated that have significantly different scales (multiplication is less critical), especially if done repeatedly, then this has potentially negative numerical effects on backpropagation. It can also quickly lead to signal dilution.
- **Avoid significant scale decay/growth.** For example, if each residual block in a network initially causes a rescaling of signals by only 50% on average, then a moderate 12-layer transformer with cross-attention will initially produce outputs that are only one 70 billionth of the inputs, which is undesirable for learning.
- **Aim for around unit scale throughout.** Making an optimizer deal with vastly different scales of signals and gradients throughout the network is undesirable, even if the optimizer has features to help deal with it. Floating-point representations have their limits, especially if attempting to train in reduced or mixed precision. Certain layers, like for example certain activation functions (*e.g.* GELU), are also ‘designed’ for approximately unit standard deviation scales of input data.

Based on these rules of thumb, it can immediately be seen why transformers with post-layer normalization [68]

in general struggle to train for larger numbers of layers. In each residual block, the initially random weights in the residual path cause an essentially random residual signal to be added to the input signal, before the combined signal is then layer-normalized. This results in the input signal to the next residual block only being a certain fraction the input signal, and the rest just random noise. As this combined signal is now the new input signal for the next block, it further down-scales the original input signal by a certain factor and adds even more random noise. This process continues throughout the entire transformer, leading to prototypical signal dilution, with the problem getting *exponentially* worse for larger numbers of layers. This is why pre-layer normalization was chosen for NOVIC.

A.2. Data Variance Propagation

We must first understand how variances propagate through the individual layers of a transformer network before we can integrate this knowledge into a complete strategy.

Addition. If a vector containing elements of variance σ_1^2 is added to an (uncorrelated) vector of variance σ_2^2 , then the variance of the elements in the sum of the two vectors is

$$\sigma_{1+2}^2 = \sigma_1^2 + \sigma_2^2. \quad (1)$$

Linear layer. If a vector $\mathbf{v} = (v_1, \dots, v_D)$ of dimension D containing elements of variance σ_D^2 is passed to a (zero-bias or biasless) linear layer of output dimension T , then given weights w_{ij} of variance σ_w^2 , the expected variance of the output elements $\mathbf{u} = (u_1, \dots, u_T)$ is

$$\begin{aligned} \sigma_T^2 &= \text{Var}(u_i) = \text{Var}\left(\sum_{j=1}^D w_{ij} v_j\right) = \sum_{i=1}^D \text{Var}(w_{ij} v_j) \\ &= \sum_{i=1}^D \text{Var}(w_{ij}) \cdot \text{Var}(v_j) = D \cdot \sigma_w^2 \cdot \sigma_D^2. \end{aligned} \quad (2)$$

A.3. Variance-Balanced Transformer Initialization

In terms of the layers that are relevant for the statistical analysis of the data passing through the model during training, the object decoder fundamentally consists of (cf. Fig. 3):

- **Input stage:** A linear projection of the input embedding vector in parallel to the token embeddings of the ground truth object noun tokens, followed by an additive positional embedding,
- **Transformer:** L consecutive transformer decoder layers, each sequentially consisting of two residual blocks, first for multi-head attention and then for feedforward layers, and,

- **Output stage:** A layer normalization, followed by a linear projection to obtain the token logits, where the weights of the projection are tied to the token embeddings matrix.

Note that no biases are used throughout the architecture, including in all linear, layer norm, and multi-head attention layers, and thus can be mathematically neglected in the analysis. If they were present, they would be initialized to zero anyway however, to allow them to be used in the long run by the optimizer, but initially not affect the optimization too much. We proceed with details of the variance-balanced weight initialization strategy for each part of the transformer model.

Input stage. The input to NOVIC is a unit embedding vector $\mathbf{e} = (e_1, \dots, e_F)$ of dimension F (*e.g.* $F = 768$ for the nominal CLIP embedder), so the elements of this input signal vector have a zero-mean variance of

$$\sigma_e^2 = \frac{1}{F} \sum_{i=1}^F e_i^2 = \frac{1}{F} \|\mathbf{e}\|^2 = \frac{1}{F}. \quad (3)$$

To ensure approximate unit scale throughout the transformer post-initialization, we wish for each signal element throughout the network to have near-unit variance. As such, we initialize the weights W_p of the linear projection layer from a zero-mean normal distribution of variance $\frac{1}{2}$, *i.e.* $\mathcal{N}(0, \frac{1}{2})$, and do the same for all weights in the token embeddings W_{te} and learned position embeddings W_{pe} . If W_{te}^* is the learnable token embeddings for the specific ground truth tokens being passed to the object decoder, then from Eqs. (1) to (3), we can see that the expected variance σ_p^2 post-positional embedding, *i.e.* of the actual input sequence vectors to the transformer layers, is thus

$$\begin{aligned} \sigma_p^2 &= \text{Var}(\text{Concat}(\mathbf{e}W_p, W_{te}^*) + W_{pe}) \\ &= \text{Concat}(\text{Var}(\mathbf{e}W_p), \text{Var}(W_{te}^*)) + \text{Var}(W_{pe}) \\ &= \text{Concat}\left(F \cdot \text{Var}(W_p) \cdot \sigma_e^2, \frac{1}{2}\right) + \frac{1}{2} \\ &= \text{Concat}\left(F \cdot \frac{1}{2} \cdot \frac{1}{F}, \frac{1}{2}\right) + \frac{1}{2} \\ &= 1, \end{aligned} \quad (4)$$

as desired. Note that some simplifying leniency is being used in the notation here by referring to elementwise variances with $\text{Var}(\cdot)$ of a matrix, and to work around the detail that the PH outputs of the linear projection layer need to be split into P (number of prefix tokens) individual sequence vectors of dimension H for the dimensions to work out.

Transformer. The decoder-only transformer used in NOVIC consists of L layers, each containing two residual

blocks, for a total of $2L$ consecutive residual blocks that strictly add calculated residuals to the input signal but otherwise leave it untouched. Due to the random weights used in the calculation of the residual contributions, the residual blocks can be modeled as adding statistically independent noise to the main path that we want to ensure does not excessively dilute the input signal. We initialize the weights of each residual pathway so that it contributes an expected variance of $\frac{1}{2L}$, leading to a final expected output variance of 2, of which half statistically comes directly from the input signal. The immediately following layer normalization of the output stage then renormalizes the scale of the data.

We achieve a residual pathway variance of $\frac{1}{2L}$ in each self-attention block by using weights of all 1 for the leading layer normalization, using weights of variance $\frac{1}{H}$ in order to make the query-key-value projections variance-preserving, and using weights of variance $\frac{1}{2LHS}$ in order to generate the desired output variance while accounting for (using S) the variance reduction that occurs in the scaled dot-product attention (SDPA). It can be shown that for P prefix tokens and an expected input variance σ^2 to the SDPA, a good approximation of the variance reduction is given by the factor

$$S = \frac{1}{P} \left(1 + \sigma^4 \frac{(P-1)}{P} \right). \quad (5)$$

As the layer normalization layer outputs unit variance signals and the query-key-value projections are variance-preserving, the expected input variance to the SDPA is $\sigma^2 = 1$, so this reduces to

$$S = \frac{2P-1}{P^2}. \quad (6)$$

For the residual pathway of each feedforward block we use a similar strategy, and use weights of all 1 for the leading layer normalization, use weights of variance $\frac{1}{H}$ in order to make the first feedforward linear layer variance-preserving (refer to Eq. (2)), and use weights of variance $\frac{1}{2LKA}$ for the second feedforward linear layer, where K is the intermediate feedforward dimension, and the factor A compensates for the variance reduction of the activation function. A is estimated by transforming the expected unit normal probability distribution by the activation function and calculating the variance around zero of the resulting distribution. For ReLU this yields $A = \frac{1}{2}$, and for GELU this yields $A \approx 0.4252$.

Output stage. As the weights of the token logits linear layer is tied to the already-initialized learnable token embeddings (variance $\frac{1}{2}$), it only remains to initialize the weight of the final layer normalization that precedes it (required due to the use of pre-layer normalization). As the final token logits need to be a suitable scale for the log-softmax operation that follows as part of the loss, we effectively ‘normalize’ the output sequence vectors by initializing the output layer normalization weights to $\sigma_o = \frac{1}{\sqrt{H}}$. Using Eq. (2), this results

in token logits of standard deviation

$$\sigma_l = \sqrt{H \cdot \frac{1}{2} \cdot \left(\frac{1}{\sqrt{H}}\right)^2} = \frac{1}{\sqrt{2}} \approx 0.707, \quad (7)$$

which is a reasonable near-unit standard deviation that provides a good slightly conservative initialization of computed forward pass logits—crucially, without excessive input signal dilution—for fast stable convergence. The complete variance-balanced weight initialization strategy has been empirically confirmed at every layer of the model to work exactly like the statistical modeling suggests (also as a verification of the underlying independence assumptions), up to the expected approximations already discussed.

B. Object Noun Dictionary

The object noun dictionary was created based on a multitude of data sources, but most prominently the WordNet dictionary [47], GNU Collaborative International Dictionary of English [9], and the categories used by 35 different image classification and object detection/segmentation benchmarks. The benchmarks were divided into two groups, *object* and *general* benchmarks, depending on whether the class names were considered to be all valid object nouns or alternatively only used to provide frequency boosts, respectively. The steps used to generate the object noun dictionary were as follows:

1. All noun synsets in WordNet were collected that are part of one of the following 11 lexical classes (considered to be the only ones out of the 26 available that could possibly contain object nouns): animal, artifact, body, event, feeling, food, object, person, phenomenon, plant, or possession.
2. The entire WordNet synset hierarchy was formatted as a tree and used to manually select nodes and sub-hierarchies considered to fit the definition of an object noun. 1598 manual synset specifications were recorded and used to select 23 942 synsets. This is the only step that required manual human intervention.
3. The chosen synsets were expanded to a working list of lemmas, and canonicalization was used to collect and deal with alternate spellings. A total of 42 981 canonical lemmas resulted.
4. 4105 unique object names were loaded from the *object* benchmarks and used to increase the number of canonical lemmas to 43 588.
5. A dataset of 1818 spelling equivalence mappings between American and British English were used to add additional spellings to the lemmas.

6. The GNU Collaborative International Dictionary of English was parsed and used to provide lemma noun alternates, and matches between singular and plural forms, especially for irregular nouns.
7. The WordNet plural noun exceptions list was used to deal with multiple singulars that share a plural form and multiple plurals that share a singular form.
8. A word inflection strategy was used to generate singular forms for nouns that had only appeared as plurals thus far, and the plural forms for all seen singulars.
9. The main spelling was resolved for each lemma with the use of language heuristics, and multiple lemmas that shared common spellings across singulars and plurals were merged. At this point there were 118 045 noun alternates in total across the now 42 919 lemmas.
10. The n-gram frequencies of the Google Web Trillion Word Corpus [8, 49] were used to assign frequencies to each individual alternate of each lemma.
11. 68 077 (non-unique) class names were loaded from the *object* and *general* benchmarks, and used to boost the frequencies of matching nouns in the dataset.
12. All noun frequencies were converted to an affine log-scale to better represent how often the corresponding nouns should be sampled during dataset generation.

The final object noun dataset has 65 210 singulars and 31 316 plurals for a total of 96 526 noun alternates that are associated with 42 919 unique canonicalized object nouns. The average log-scale frequency of each noun alternate is 1.5, resulting in an average log-scale frequency of 3.5 per object noun.

C. LLM-based Caption Generation

Given the complete object noun dictionary, an LLM was used to generate rich dynamic captions involving each of the object nouns in the dictionary. Based on the log-frequencies obtained from the object noun dictionary, ten times as many captions are generated independently for singular and plural forms of the noun, with at most 100 being generated for either category. OpenAI GPT-3.5 Turbo was used, specifically *gpt-3.5-turbo-1106* with a temperature of 0.8, maximum 1000 output tokens, and example prompts/responses as shown in [Prompt 1](#). Ten captions are generated in every API call, and parsed using all available spelling variations of the object noun into appropriate caption templates. For instance, "*Two children playing with a bicycle in the yard*" is parsed and stored as "*Two children playing with a {singular} in the yard*". In cases of words where there are multiple spellings, like for example *orangutan*, this ensures that a caption template can be used in future to generate data for any desired

Prompt 1: Singular/plural caption generation examples

SYSTEM: You are a helpful AI assistant.

PROMPT: Generate a list of 10 image captions each containing the target object “bicycle” (singular). Use the style of the image captions for large language model pre-training. Don’t use poetic language. Use a different sentence structure for each caption. Output the list in the following form: <nr>. <caption>

RESPONSE:

1. A red bicycle parked against a brick wall.
2. The bicycle resting under a tree in the park.
3. A woman riding a bicycle along the beach.
4. A vintage bicycle leaning against a lamppost.
5. Two children playing with a bicycle in the yard.
6. The bicycle locked to a metal railing outside the store.
7. A man carrying a folded bicycle up the stairs.
8. The bicycle lying on the grass with its wheels spinning.
9. A bicycle with a basket filled with colorful flowers.
10. The bicycle propped up against a wooden fence.

SYSTEM: You are a helpful AI assistant.

PROMPT: Generate a list of 10 image captions each containing the target object “bicycles” (plural). Use the style of the image captions for large language model pre-training. Don’t use poetic language. Use a different sentence structure for each caption. Output the list in the following form: <nr>. <caption>

RESPONSE:

1. A row of bicycles parked by the street, waiting for their riders.
2. Bicycles lined up neatly in a bike-sharing station, ready for use.
3. Many bicycles secured to a rack outside a busy train station.
4. Several bicycles leaning against a wall, casting shadows in the afternoon sun.
5. Dozens of bicycles scattered across the open field during a cycling event.
6. A group of bicycles locked together in the urban bike parking area.
7. Bicycles parked in a row, their colorful frames catching the eye of passersby.
8. Multiple bicycles lined up near a city park, offering a convenient mode of transportation.
9. A cluster of bicycles parked haphazardly outside a bustling cafe.
10. Bicycles stacked on top of each other in a crowded bike storage area.

spelling if required by the sampling strategy. A total of 1.8M captions were generated in this way.

D. Effect of Noise Augmentation

Based on experiments, the ‘magnitude’ of Gaussian noise that performed best for noise augmentation during training is $G = 3.25$. The precise definition of how this noise was implemented was by adding elementwise Gaussian noise with a standard deviation of

$$\sigma_G = \frac{G}{\sqrt{F}} \quad (8)$$

to the input text embedding vectors and then renormalizing the result back to a unit vector. For the SigLIP B/16 model, which has an embedding dimension of $F = 768$, this corresponds to an elementwise standard deviation of $\sigma_G \approx 0.117$. If $\mathbf{g} = (g_1, \dots, g_F)$ is the random noise vector with $g_i \sim \mathcal{N}(0, \sigma_G^2)$, then we can see that

$$\mathbb{E}[\|\mathbf{g}\|^2] = \mathbb{E}\left[\sum_{i=1}^F g_i^2\right] = \sum_{i=1}^F \mathbb{E}[g_i^2] = F\sigma_G^2 = G^2, \quad (9)$$

$$\begin{aligned} \text{Var}(\|\mathbf{g}\|^2) &= \text{Var}\left(\sum_{i=1}^F g_i^2\right) = \sum_{i=1}^F \text{Var}(g_i^2) = \sum_{i=1}^F 2\sigma_G^4 \\ &= 2F\left(\frac{G}{\sqrt{F}}\right)^4 = \frac{2G^4}{F}. \end{aligned} \quad (10)$$

For our case, this means the random noise vector has an average square-norm $\|\mathbf{g}\|^2$ of $G^2 \approx 10.56$ with a standard deviation of $G^2 \cdot \sqrt{\frac{2}{F}} \approx 0.539$. This clearly shows that statistically, the added noise vector will *always* have a norm significantly larger (on the order of $\|\mathbf{g}\| \approx G = 3.25$) than the unit embedding vector $\mathbf{e} = (e_1, \dots, e_F)$, which has $\|\mathbf{e}\| = 1$. On top of being such a large vector, due to the high dimensionality of the embedding space, the random noise vector \mathbf{g} is actually also essentially *guaranteed* to be near-perpendicular to the embedding vector \mathbf{e} , causing a large directional change of the embedding vector no matter what—no embedding vector can ever remain anywhere close to where it started. To demonstrate this, we consider the cosine of the angle θ between \mathbf{e} and \mathbf{g} , given by

$$\cos \theta = \frac{\mathbf{e} \cdot \mathbf{g}}{\|\mathbf{e}\| \|\mathbf{g}\|} = \sum_{i=1}^F e_i \frac{g_i}{\|\mathbf{g}\|}. \quad (11)$$

Due to the statistical independence of each element,

$$\begin{aligned} \mathbb{E}[\cos \theta] &= \mathbb{E}\left[\sum_{i=1}^F e_i \frac{g_i}{\|\mathbf{g}\|}\right] \\ &= \sum_{i=1}^F e_i \mathbb{E}\left[\frac{g_i}{\|\mathbf{g}\|}\right] \xrightarrow[0]{\text{symmetry}} \\ &= 0, \end{aligned} \quad (12)$$

due to the symmetrical isotropic Gaussian nature of \mathbf{g} . The corresponding variance is then

$$\begin{aligned} \text{Var}(\cos \theta) &= \mathbb{E}[(\cos \theta)^2] - \mathbb{E}[\cos \theta]^2 \xrightarrow{0} \\ &= \mathbb{E}\left[\left(\sum_{i=1}^F e_i \frac{g_i}{\|\mathbf{g}\|}\right)^2\right] \\ &= \sum_{i=1}^F e_i^2 \mathbb{E}\left[\frac{g_i^2}{\|\mathbf{g}\|^2}\right] + \sum_{i \neq j} e_i e_j \mathbb{E}\left[\frac{g_i g_j}{\|\mathbf{g}\|^2}\right] \xrightarrow[0]{\text{orthogonality}} \\ &= \sum_{i=1}^F e_i^2 \cdot \frac{1}{F} \\ &= \frac{1}{F}, \end{aligned} \quad (13)$$

as \mathbf{e} is a unit vector. In other words, the cosine similarity between an embedding vector and the noise vector that is added to it is on average 0 (perpendicular) with a very

narrow standard deviation of $\frac{1}{\sqrt{F}} \approx 0.036$, implying essentially guaranteed near-perpendicularity. The average angle β between the initial embedding vector e and the noise augmented vector $\frac{e+g}{\|e+g\|}$ is thus approximately

$$\beta \approx \text{atan}\left(\frac{\|g\|}{\|e\|}\right) \approx \tan^{-1} G \approx 72.9^\circ. \quad (14)$$

The standard deviation of the angle β can empirically be measured to be 1.94° . This narrow distribution of β , *i.e.* the distribution of angle separations that occur when using the nominal Gaussian noise augmentation strategy, is shown in Fig. 5. Also shown in the plot is the measured distribution of angle separations for *matching* and *non-matching* image-label pairs of the ImageNet-1K validation set, *i.e.* demonstrating the modality gap as well as the very thin separation that CLIP models provide between matching and non-matching pairs. The positive distribution-widening effect of the nominal strategy of adding 15% uniform angle noise from $45\text{--}75^\circ$ is also shown (labeled “Gauss 3.25 + Uniform” in Fig. 5). With this nominal strategy, for 85% of all samples Gaussian noise of magnitude $G = 3.25$ is added, and for the remaining 15% of samples (randomly chosen), uniform angle noise of random magnitude $45\text{--}75^\circ$ is applied.

E. Open Vocabulary Image Datasets

Three open vocabulary image datasets were collected as part of this work, World, Wiki and Val3K, as described in the following sections. All three datasets were annotated both manually by humans and by a multimodal LLM (cf. App. F),⁸ using one of five labels for each predicted object noun—*correct primary* (1.0 points), *correct secondary* (0.8 points), *close primary* (0.5 points), *close secondary* (0.4 points), and *incorrect* (0 points). The sum of all points is divided by the number of images to get the final percent accuracy *prediction score*. The suffixes -H and -L are used for the datasets to signify which annotations are being used, leading to five open vocabulary evaluation datasets, World-H, World-L, Wiki-H, Wiki-L, and Val3K-L.

E.1. World Dataset

The World dataset was collected with the express intention of trying to curate a dense collection of both regular and tricky object concepts. This included a specific focus on trying to include images from around the world that have never been on the internet (and thus cannot ever have been trained by a CLIP model before). Tricky images include ones that show: unusual objects, common object concepts but in peculiar variations, deceptive objects that are one thing but styled to look somewhat like another, and indirect representations of objects, *e.g.* paintings, cartoons, posters,

⁸All except human annotations for Val3K.

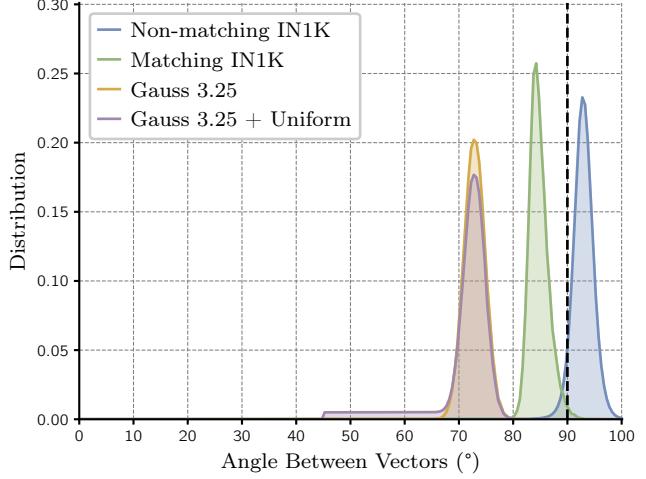


Figure 5. Comparison of angle separation distributions. A plot of the distribution of embedding vector angle separations in 768-dimensional embedding space for matching and non-matching image-label pairs from the ImageNet-1K validation set. Also shown is the distribution of angle separations between original and noise-augmented text embeddings for both the pure Gaussian and Gaussian with uniform strategies. The dashed vertical line shows exactly perpendicular vectors that are thus mathematically uncorrelated.

sculptures, stuffed toys, drone show patterns, and such. A total of 12 people collected 85.3% of the images from original photographs in 10 countries of the world (Australia, Austria, England, France, Germany, Netherlands, New Zealand, Philippines, UAE, USA), with an active focus on covering as wide and varied concepts as possible. The remaining 14.7% of images were complemented from web sources to help cover concepts (*e.g.* animals) that could not so readily be manually photographed. The 272 images consist of 15.4% animals, 10.0% plants, 16.2% wider scenes, 6.6% indirect representations of objects, and 51.8% focused images of objects. Both human (World-H) and LLM (World-L) annotations are available for a wide variety of evaluated models.

E.2. Wiki Dataset

The Wiki dataset was collected by scraping 18 660 lead images, *i.e.* ‘title images’, from Wikipedia articles. Lead images often show a clear object concept that can be predicted and annotated, reducing the number of ambiguities and grey zones that need to be dealt with consistently during annotation. The scraped images had a noticeable bias in their topic of focus towards plants, animals, and people, so the sampling of the scraped images to a dataset size of 1 000 was guided by a probability-adjusted sampling method to conservatively correct that bias. Both human (Wiki-H) and LLM (Wiki-L) annotations were made for a variety of evaluated models.

E.3. Val3K Dataset

The Val3K dataset aims to help quantify how well a model is *really* doing on the ImageNet-1K validation set, if it is not forced to output lower probability classifications in order to align with the class names of the dataset. The dataset was collected by sampling 3 random validation images (out of the 50 available) for each of the 1 000 ImageNet-1K classes. This led to a dataset of 3 000 images of mixed content. LLM annotations were made on all 3 000 images to obtain Val3K-L. As LLM annotations, especially when passing images to a multimodal LLM, are attributed with cost, it was not feasible to perform annotations on the full 50 000 ImageNet-1K validation set.

F. Multimodal LLM Image Annotation

Due to the recent availability of some multimodal LLMs with a very high level of image understanding, it is conceivable to approximately quantify the performance of open vocabulary classification models by auto-annotating their predictions on image sets using these LLMs. This is particularly useful when it is just not feasible to perform the required amount of annotations as a human. The general process is as follows:

1. The chosen to-be-evaluated model(s) are inferred on the required dataset of images, and the predicted object nouns are recorded (saving the top-k predictions for each image is also possible).
2. For each individual image, the deduplicated set of all predicted object nouns across all models is collected.
3. Groups of up to 15 of these predicted object nouns are evaluated per call to the LLM, until every object noun has up to 5 opinions from the LLM whether it is correct or incorrect (present in the image or not).
4. A ground truth annotation is established per-image per-predicted object noun based on the opinions that were given by the LLM.

We use OpenAI GPT-4o, specifically *gpt-4o-2024-05-13* with a temperature of 0.2, maximum 512 output tokens, and a cumulative probability threshold of 0.6 for nucleus sampling. The chosen parameters reflect the stability and consistency that is desired from the LLM. The prompt used for auto-annotation is shown in [Prompt 2](#) for an example image. Asking the LLM to first provide a description of the image allows it to adjust its summarized knowledge of the image according to the provided object noun candidates, and enhances its classification precision. For best results, 8–15 object noun candidates are always given, as providing too few or too many has a small potential of causing inconsistent annotations for fine-grained object nouns. With intentional

Prompt 2: Multimodal LLM image classification

SYSTEM: You are an AI assistant that has one and only one narrow task that you should strictly adhere to at all times. Given an image and an enumerated list of nouns by the user, you should first describe everything you see in the image in complete detail, and then provide an exactly matching enumerated list that for each noun provided by the user (explicitly repeat the noun) strictly classifies the noun into one of exactly two categories using a single word - Correct or Incorrect. Correct means that at least one instance of the noun is visible in the image. Incorrect means nothing very visually similar to the noun is visible in the image.

PROMPT: <image base64> (max 512×512)

- 1) african hunting dog
- 2) african wild dog
- 3) canis niger
- 4) cape hunting dog
- 5) hyena
- 6) hyena dog
- 7) lycaon pictus
- 8) mammal
- 9) mammalian
- 10) mexican hairless
- 11) mexican hairless dog
- 12) monkey dog
- 13) okapi



RESPONSE:

The image shows an African wild dog, also known as Lycaon pictus. It has a distinctive coat with patches of black, brown, white, and yellow, large rounded ears, and a slender body.

- 1) african hunting dog - Correct
- 2) african wild dog - Correct
- 3) canis niger - Incorrect
- 4) cape hunting dog - Correct
- 5) hyena - Incorrect
- 6) hyena dog - Incorrect
- 7) lycaon pictus - Correct
- 8) mammal - Correct
- 9) mammalian - Correct
- 10) mexican hairless - Incorrect
- 11) mexican hairless dog - Incorrect
- 12) monkey dog - Incorrect
- 13) okapi - Incorrect

randomization of candidate ordering and which candidate is picked for which API request, each object noun is annotated 3–5 times, stopping before 5 if there is at least 80% agreement in the LLM’s opinions. If the final agreement is at least 80%, then the corresponding opinion (*correct* or *incorrect*) is taken as the final annotation, otherwise the annotation is *close*. No *primary* vs. *secondary* classification is made because it was experimentally deemed too inconsistent. Due to the relatively unlikely condition required for the *close* category (LLMs tend to be wrong more often than they are inconsistent at a low temperature), it is systematically not seen as often as for human annotations.

In order to verify that the LLM annotations are sufficiently accurate to make inferences about the performance of evaluated NOVIC models, we compared the prediction scores computed on human and LLM annotations for 397 models on the World dataset (cf. [Fig. 6](#)). As expected, a clear correlation and trend can be identified, whether *close* classifications are considered in the scores or not, with residual root mean square errors of 1.1% in both cases. This suggests that even though the absolute values of the scores do noticeably differ, the coarse ordering it applies to models is similar.

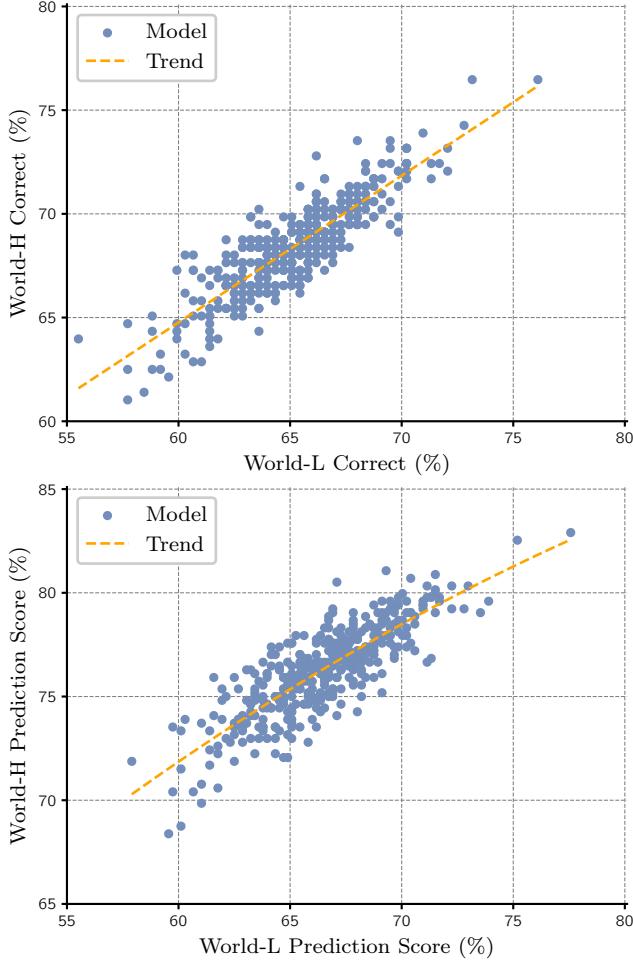


Figure 6. Comparison of human and LLM image annotations. World-H scores are plotted against World-L scores for 397 models. **Top:** Scores considering *correct* only. **Bottom:** Standard prediction scores considering *close* classifications as well. For human annotations, *secondary* classifications are scored as *primary* as the LLM does not make that distinction.

G. Model Hyperparameters

The nominal values of various NOVIC hyperparameters are shown in Tab. 5. The corresponding hyperparameter value ranges that were covered in some configurations during testing are also shown.

H. Qualitative Comparison

A qualitative comparison between NOVIC and image tagging baselines on the World-H and Wiki-H datasets is presented in Fig. 7 and Fig. 8, respectively. RAM and Tag2Text in general provide relatively coarse-grained classifications and often miss central concepts in the images, focusing instead on known nouns. For example, they output *man* for an image where a man is presenting a *banknote* to the camera.

Hyperparameter	Value	Tested Range
CLIP model	SigLIP B/16	See Tab. 4
CLIP mixed precision	float16	float16
Noun frequency threshold	0	0–10
Multiset order m	3	1–4
LLM captions	✓	✗/✓
Noise augmentation	Mix	Gauss, Uniform, Mix
Gauss noise magnitude	3.25	0.0–4.0
Uniform angle noise	45–75	Up to 0–85
Noise mix factor	15%	0–20%
Prefix tokens P	4	2–6
Hidden dim H	512	384, 512
Feedforward dim K	128	64–512
Num layers L	6	4–12
Num heads	8	8
Dropout	0.1	0.05–0.2
Bias parameters	✗	✗/✓
Label smoothing	✗	✗/✓
Beam width	10	1–10
Temperature	1	0.33–3
Epochs E	18	9–24
Batch size B	8192	512–16384
Optimizer	AdamW	AdamW, AdamP
Adam β_1	0.9	0.9
Adam β_2	0.95	0.95, 0.99
LR scheduler	cosine	cosine
LR warmup epochs	0	0–2
Initial LR	1.5e-3	5e-4–3e-3
Weight decay	0.1	0–0.3
Weight decay for 1D	✗	✗/✓

Table 5. NOVIC hyperparameter values and tested ranges. Batch sizes use gradient accumulation up to factor 16.



Tag2Text	stage (I)	water (S)	ladder (I)	bird (P)
RAM	stage (I)	water (S)	pavement (S)	bird (P)
RAM (FT0)	stage (I)	water (S)	pavement (S)	bird (P)
SigLIP NOVIC (FT2)	trophy cup (P)	hippopotamus amphibius (P)	bicycle rack (P)	parrot (CP)
SigLIP NOVIC (FT0)	trophy cup (P)	hippopotamus amphibius (P)	bicycle rack (P)	mourning dove (I)
DFN-5B NOVIC (FT0)	firework (S)	hippopotamus (P)	bicycle rack (P)	kea (P)



Tag2Text	man (P)	palm tree (S)	walk (I)	car (P)
RAM	man (P)	palm tree (S)	walk (I)	car (P)
RAM (FT0)	man (P)	palm tree (S)	female person (P)	car (P)
SigLIP NOVIC (FT2)	money (P)	wreath (P)	pavement (P)	roof rack (P)
SigLIP NOVIC (FT0)	money (P)	wreath (P)	pedestrian crossing (P)	roof rack (P)
DFN-5B NOVIC (FT0)	banknote (P)	florida key (P)	pedestrian (P)	roof rack (P)



Tag2Text	cake (P)	flower (P)	animal (I)	panda (P)
RAM	cake (P)	yellow (I)	blue (I)	red (I)
RAM (FT0)	cake (P)	flower (P)	blue (I)	red cole (I)
SigLIP NOVIC (FT2)	red velvet cake (P)	jonquil (P)	stuffed toy (P)	red panda (P)
SigLIP NOVIC (FT0)	cake cake (I)	jonquil (P)	stuffed toy (P)	red panda (P)
DFN-5B NOVIC (FT0)	red velvet cake (P)	daffodil (P)	genus staphylococcus (P)	ailurus (P)

Figure 7. Examples of open vocabulary classification on the World-H dataset. While RAM and Tag2Text select the most likely noun from a limited vocabulary, NOVIC can freely output any object noun. Possible classification scores are *correct primary* (P), *correct secondary* (S), *close primary* (CP), *close secondary* (CS), and *incorrect* (I). *Correct primary* and *incorrect* labels are highlighted in color. Output labels that are not nouns are uninformative for the image and thus considered incorrect. Similarly, invalid nouns are also deemed incorrect.

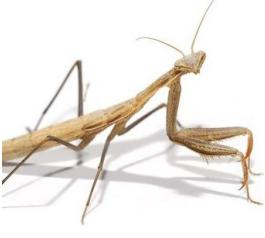
	Tag2Text RAM RAM (FT0) SigLIP NOVIC (FT2) SigLIP NOVIC (FT0) DFN-5B NOVIC (FT0)	elk (P) moose (CP) stand (I) buckskin (I) moose wood (I) elk (P)	man (S) hand (S) finger (S) hearing aid (P) hearing aid (P) hearing aid (P)	tree (P) tree (P) tree (P) brewers spruce (CP) western larch (CP) subalpine larch (P)	flower (P) flower (P) flower (P) babies breath (P) genus saponaria (CP) genus gysophila (P)
	Tag2Text RAM RAM (FT0) SigLIP NOVIC (FT2) SigLIP NOVIC (FT0) DFN-5B NOVIC (FT0)	mantid (P) photo (I) mantis (P) praying mantis (P) praying mantis (P) mantis (P)	grass (CS) armadillo (P) armadillo (P) nine banded armadillo (P) armadillo (P) dasypus (P)	church (P) church (P) church (P) church (P) church (P) aurora (P)	old (I) tool (P) tool (P) spur (P) spur (P) spur (P)
	Tag2Text RAM RAM (FT0) SigLIP NOVIC (FT2) SigLIP NOVIC (FT0) DFN-5B NOVIC (FT0)	dog (P) dog (P) dog (P) rhodesian ridgeback (P) rhodesian ridgeback (P) rhodesian ridgeback (P)	telescope (CP) camera (CP) camera (CP) viewfinder (I) viewfinder (I) magic lantern (P)	monkey (P) sit (I) monkey (P) gibbon (P) hylobates syndactylus (CP) gibbon (P)	roll (P) roll on (I) roll (P) plaster bandage (P) compression bandage (P) compression bandage (P)

Figure 8. **Examples for open vocabulary classification on the Wiki-H dataset.** The images are an assortment of Wikipedia lead images, covering a broad range of visual concepts. Possible classification scores are *correct primary* (P), *correct secondary* (S), *close primary* (CP), *close secondary* (CS), and *incorrect* (I). *Correct primary* and *incorrect* labels are highlighted in color.

Configuration	World-H	Wiki-L	IN1K
Prefix tokens $P = 2$	77.91	62.75	65.51
Prefix tokens $P = 4^*$	78.92	63.97	68.11
Prefix tokens $P = 6$	78.03	64.15	68.08
Layers $L = 4$	77.23	62.38	63.85
Layers $L = 6^*$	78.92	63.97	68.11
Layers $L = 8$	77.94	62.78	69.05
Layers $L = 12$	78.15	64.00	71.19
Hidden dim $H = 384$	76.97	61.65	65.77
Hidden dim $H = 512^*$	78.92	63.97	68.11
Batch size $B = 4096$	78.14	63.73	66.76
Batch size $B = 8192^*$	78.92	63.97	68.11
Batch size $B = 16384$	79.56	63.17	67.85
Epochs $E = 12$	77.33	63.30	66.33
Epochs $E = 18^*$	78.92	63.97	68.11
Epochs $E = 24$	77.77	63.00	67.71

Table 6. Object decoder architecture and training hyperparameter ablations (% mean of 3). IN1K is the ImageNet-1K classification performance of the trained decoders, and the asterisks correspond to the nominal ablation baseline (FT2). World-H and Wiki-L are prediction scores (described in Sec. 4) on the corresponding datasets. See Sec. 4.2 for more ablation results.

Additionally, they sometimes output colors (*e.g. blue, yellow*), verbs (*e.g. walk, stand*), attributes (*e.g. old*), or merely image style information (*e.g. photo*), thereby failing to convey the actual content of the image. Even when given access to the complete object noun dictionary (*i.e.* FT0), which is an order of magnitude more comprehensive than the RAM vocabulary, RAM predominantly predicts the familiar words it was trained on rather than utilizing a broad range of object nouns as labels.

On the other hand, NOVIC demonstrates strong capabilities in precisely detecting and outputting the central object of an image. Particularly when trained on the DFN-5B CLIP model, NOVIC provides very fine-grained classifications and can accurately identify Latin names of plants and animals (*e.g. genus gypsophila*), names of very specific uncommon objects (*e.g. magic lantern*), and even geolocations of pictures (*e.g. florida key*) with high precision. Since NOVIC generates free-form text during inference, it occasionally fails to “stop in time” when outputting nouns however, resulting in incorrect labels (*e.g. cake cake, moose wood*). This issue is more prevalent for smaller CLIP models and lower word frequency thresholds of the training dataset.

I. Decoder Architecture Ablations

Ablations of the core NOVIC model hyperparameters are shown in Tab. 6. The results support the nominal choice of hyperparameters, and suggest that slight gains could poten-

tially be made with a greater number of transformer layers (although model generalization becomes slightly less reliable) and/or batch size. Refer to Sec. 4.2 for ablation results pertaining to the training dataset, noise augmentation, and CLIP model.

J. Image Classification Benchmark Trends

Tab. 1 shows the performance of NOVIC on various image classification benchmarks, many of which are commonly used to benchmark the zero-shot image classification performance of CLIP models. In general, the trend can be seen that the classification performance of NOVIC models slowly drops off as the complexity and extensiveness of the training data is increased towards the full size of the object noun dictionary. This can be expected, as the object decoder has increasingly many object concepts to learn as the frequency threshold (FT) decreases, and these concepts are also increasingly densely packed in the embedding space, making them harder to accurately keep apart. An exception to this general trend for some of the classification benchmarks is the FT9 score, as the FT9 training data does not actually always cover all of the target nouns required for each dataset. This concretely means that 412/1000 ImageNet-1K classes, 17/200 Tiny ImageNet classes, 50/200 ImageNet-A classes, and 37/200 ImageNet-R classes, cannot contribute to the final classification score, resulting in the disadvantaged lower scores marked with a dagger (\dagger) in Tab. 1. Another exception to this general trend are the benchmarks that have few and coarse-grained classes, such as Imagenette and CIFAR-10. For these datasets, the performance depends less on the diversity of object nouns seen during training, and more on just the raw ability of the CLIP image encoder to produce an accurate and representative embedding, resulting in fairly consistent classification performances irrespective of the frequency threshold.

Training on large-scale datasets is also inherently not deterministic—in particular also due to the randomized data (*e.g. noise*) augmentation—and some natural variations in the results will occur even if taking the mean of three training runs. The object decoder has thousands to tens of thousands of concepts to learn, and evaluating on a classification dataset with *e.g.* 10–101 classes is probing a small fraction of what the model has learned and is rewarded for by its training loss. This means that two models with identical loss will always have some concepts that they have learned better, and some they have not learned quite as robustly. If that happens to align with the classes required for a classification dataset, the models then either do slightly better or slightly worse on them. As previously mentioned, CIFAR-10 is an example of all models doing about the same, up to natural variation, as the dataset classes are very few and coarse, and the dataset is relatively ‘easy’.

The classification results for the FT6 and FT2 NOVIC

Model	Prompt-free Classification	Text-only Training	Seen Objects	Used Objects	RAM Vocab	Primary Score	Prediction Score	Overall Score
Tag2Text [29]	✗	✗	3 429	176	91.9	62.32	← ^{-15.29}	77.61 → ^{-13.49} 64.12
RAM [76]	✗	✗	4 585	160	100.0	57.35	← ^{-17.65}	75.00 → ^{-15.86} 59.14
RAM (FT2)	✗	✗	11 897	188	76.8	64.71	← ^{-11.61}	76.32 → ^{-13.84} 62.48
RAM (FT0)	✗	✗	42 919	188	74.3	63.97	← ^{-11.03}	75.00 → ^{-13.84} 61.16
NOVIC (FT9)	✓	✓	2 919	235	66.9	74.82	← ^{-2.79}	77.61 → ^{-3.22} 74.39
NOVIC (FT6)	✓	✓	5 899	252	58.5	76.47	← ^{-2.21}	78.68 → ^{-1.01} 77.67
NOVIC (FT2)	✓	✓	11 897	253	52.6	77.94	← ^{-1.62}	79.56 → ^{-0.64} 78.92
NOVIC (FT0)	✓	✓	42 919	251	49.6	71.14	← ^{-3.53}	74.67 → ^{-1.10} 73.57
NOVIC [†] (FT2)	✓	✓	11 897	263	55.5	86.77	← ^{-1.17}	87.94 → ^{-0.81} 87.13
NOVIC [†] (FT0)	✓	✓	42 919	260	48.5	86.95	← ^{-1.32}	88.27 → ^{-0.37} 87.90

Table 7. **Open vocabulary World-H results in comparison to baselines (best of 3).** Tag2Text and RAM achieve artificially high scores by frequently outputting uninformative generic tags like *man*, *tree*, *grass*, *plate*, and often miss the actual intent of an image. Scores are in %. *RAM Vocab* is the percent of samples whose predictions are in the core RAM vocabulary. *Primary Score* is the *Prediction Score* when rewarding only primary predictions. *Overall Score* is the *Prediction Score* with $\times 0.5$ applied to the score contributions of any coarse predictions, effectively penalizing vague generic predictions. See Tab. 3 for Wiki-H results. [†]Using DFN-5B H/14-378 CLIP model.

models are similar for many of the benchmarks, as both training sets have a reasonable size and comfortably cover all required class names as target nouns. The dip in performance for FT0 is more pronounced in general, due to the near-quadrupling relative to FT2 of the number of object concepts to learn, with the magnitude of the dip being tentatively correlated to larger numbers of classes and more fine-grained classification concepts overall. Both of these factors result in smaller regions of the embedding space, with less cosine distance between them, needing to be kept apart. This, in combination with the more finely ‘concept-subdivided’ embedding space of an FT0 model, leads to the performance dip.

K. Additional Baseline Comparisons

In Tabs. 3 and 7, we compare (as described in Sec. 4.1) NOVIC to the image tagging baselines Tag2Text [29] and RAM [76], on the Wiki-H and World-H datasets respectively. While the image tagging baselines require a list of noun candidates to be supplied at inference time and are trained on image-text pairs, NOVIC performs completely prompt-free image classification and is trained much more efficiently on text-only data. Also, while NOVIC is trained primarily on a synthetic text dataset constructed in a controllable and unbiased manner from an entire English dictionary of object nouns, the baselines train on ‘only’ 14M image-text pairs, which have a significantly lower diversity of seen objects, and do not offer any controllability of the distributions of the nouns and image embeddings seen during training. Furthermore, Tag2Text has a fixed set of labels, while RAM attempts to provide some level of open vocabulary recognition beyond its nominal set of labels by distilling an image encoder from the base CLIP image encoder. Due to the lim-

ited size of the used image-text dataset however, and noting that the core set of labels was already *constructed* by parsing these exact text captions, one can understand that the ‘open’ abilities of RAM are somewhat limited, *i.e.* RAM is not truly open vocabulary in any way close to how one considers CLIP to exhibit open vocabulary abilities.

In Tabs. 3 and 7, we can see that NOVIC has seen significantly more diverse object nouns during training (42 919 as opposed to less than a few thousand for the baselines), and this is clearly reflected by its ability to use a wide array of object nouns in its predictions—generally close to as many unique nouns as there are images in the corresponding dataset (*e.g.* 263 unique predictions for 272 images). The tendency of the baselines not to provide such a diverse set of predictions can be observed from their significantly lower number of used objects. The restricted ability of RAM to perform open vocabulary classifications can also be identified by observing the high proportion of its generated predictions that belong to the core RAM vocabulary, even when provided with a large variety of object nouns to choose from (*e.g.* FT0). RAM’s open vocabulary predictions are also observed to have a significantly higher error rate than core vocabulary predictions.

Three prediction scores are calculated for each model—*Prediction Score*, the standard prediction score as defined in Sec. 4, *Primary Score*, which does not reward secondary predictions and thereby emphasizes the important ability of the model to focus on the actual intended object of an image,⁹ and *Overall Score*, which adjusts the standard prediction

⁹Note that even the primary score does not adequately penalize the baselines’ scores in practice however, for example in instances where a plate of food is the focus of the image and the baselines invariably predict *plate* whereas NOVIC tends to predict the specific name of the dish.

score to reduce the reward for overly coarse predictions and thereby emphasizes the important ability of the model to provide meaningful and informative fine-grained classifications. It should be noted that both the primary and overall scores can by definition only at most be equal to the standard prediction score, so an ideal model that emits only *fine-grained primary* predictions would have all three scores be equal. We can observe that for NOVIC this is indeed almost the case, whereas the baselines exhibit very significant drops in score when penalizing secondary and coarse predictions as such. Overall, this demonstrates that NOVIC produces high quality predictions that are simultaneously diverse and fine-grained yet at the same time accurate, as well as correctly focusing on the main object of an image, all while being real-time, prompt-free, truly open vocabulary, and very efficiently trained with text only. This makes NOVIC a strong improvement over the state-of-the-art baselines.

L. Ethical AI

Our open vocabulary image classifier may inherit biases from the CLIP model it is trained on, and be similarly susceptible to adversarial attacks during inference, such as manipulated images misguiding label generation. Although the object decoder is generative, it is trained in a constrained setting, making it unlikely to produce sensitive outputs. Training the object decoder on the proposed text datasets is significantly more cost-effective than training alternative methods such as RAM on large-scale image datasets. However, due to the large amount of noise used, identifying the optimal model configuration often necessitates many training runs, each with an environmental impact.