Journal of Artificial Intelligence Research 74 (2022) 1201-1223 Submitted 02/2022; published 07/2022

CoLLIE: Continual Learning of Language Grounding from Language-Image Embeddings

Gabriel Skantze SKANTZE@KTH.SE
Bram Willemsen BRAMW@KTH.SE
KTH Royal Institute of Technology,
Stockholm, Sweden

Abstract

This paper presents CoLLIE: a simple, yet effective model for continual learning of how language is grounded in vision. Given a pre-trained multimodal embedding model, where language and images are projected in the same semantic space (in this case CLIP by OpenAI), CoLLIE learns a transformation function that adjusts the language embeddings when needed to accommodate new language use. This is done by predicting the difference vector that needs to be applied, as well as a scaling factor for this vector, so that the adjustment is only applied when needed. Unlike traditional few-shot learning, the model does not just learn new classes and labels, but can also generalize to similar language use and leverage semantic compositionality. We verify the model's performance on two different tasks of identifying the targets of referring expressions, where it has to learn new language use. The results show that the model can efficiently learn and generalize from only a few examples, with little interference with the model's original zero-shot performance.

1. Introduction

Any artificial agent interacting with an environment, using vision, and communicating with other agents (such as humans), using language, needs to be able to ground the meaning of language with the visual properties of the environment. One approach to this problem is to project vision and language into a joint semantic embedding space (Frome et al., 2013; Bruni et al., 2014). In such a model, a visual stimulus and a language construct that have similar representations are supposed to have similar meanings. In order to name a given object with certain visual features, the agent should try to generate a referring expression that has a similar embedding as the visual features of the object, and in order to understand what a referring expression is denoting, it should look for objects that have a similar visual feature embedding as that of the referring expression.

Recent developments in multimodal representation learning using large amounts of data have given impressive results. An example of a model integrating language and vision is CLIP by OpenAI (Radford et al., 2021), which was trained using constrastive learning on 400 million pairs of images and their captions. Images and texts are embedded (separately) using state-of-the-art computer vision and language processing pipelines into a 512-dimensional vector. By calculating the dot product of the two embeddings, it is possible to determine how similar an image is to a text (or an image to an image, or a text to a text), as illustrated in Figure 1a. The model was shown to be very effective at so-called *zero-shot learning*, which for CLIP means that the model can do image retrieval by ranking the similarity of images to a given label (such as "a black cat"). This can be contrasted with traditional image

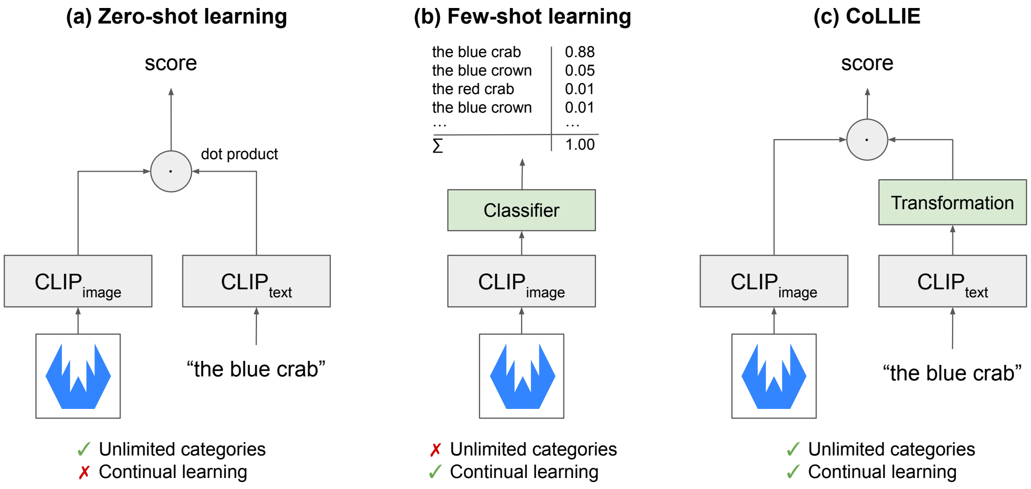
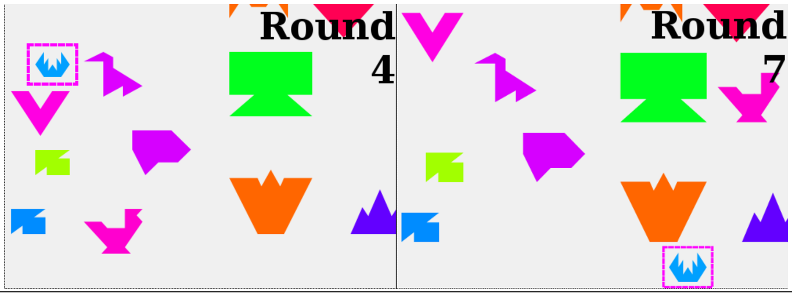


Figure 1: Comparison of CoLLIE to Zero-shot and Few-shot learning. Green boxes show where continual learning is taking place.

classification, where the model is specifically trained to classify images into a predefined set of categories (e.g., Deng et al., 2009). In addition to being more flexible (as it can use a virtually infinite set of categories), CLIP was also shown to be more robust against noise and variations in the images, compared to supervised image classification (Radford et al., 2021).

While such a model is potentially very useful for agents that need to ground language in vision, it is limited in that it is trained once, without any mechanism for updating its representations in light of new data, unless the entire model is retrained and the number of new examples is sufficient. This is clearly limiting the model's usefulness in real-life application scenarios for agents interacting in a dynamic environment. Not only will new objects with new properties emerge, but the way humans talk about objects changes over time. As has been shown repeatedly in experiments on human-human interaction, this is not only a long-term issue, but the exact meaning of language may often be negotiated and evolve during the course of a single interaction, and then develop into partner-specific language use (Brennan & Clark, 1996; Barr & Keysar, 2002; Brennan & Hanna, 2009; Ibarra & Tanenhaus, 2016; Shore & Skantze, 2018). This phenomenon has been referred to as *conceptual pacts* (Brennan & Clark, 1996), or more generally as *alignment* in communication (Pickering & Garrod, 2006). For example, if a hard-to-describe object is being referred to, the partners might establish a new name for it and then continue using that name for similar objects. An example of this is shown in Figure 2, where two human subjects were asked to play a game where they take turns referring to tangram figures on a shared game board (Shore & Skantze, 2018). In round 4, speaker B uses the referring expression "a blue crab sticking up his claws", and in round 7, speaker A adopts the term "crab" when referring to it again. Other pairs of subjects formed other conceptual pacts for the same shape, such as "bat" or "crown". Furthermore, humans can make use of *semantic compositionality*, which means that if one person refers to an object with "the blue crab", it is likely that the expression "the pink crab" would be understood as a reference to a similarly shaped object in a different color. Thus, the novel language use "crab" can be combined with the already established language use for colors.

It is unreasonable to expect that a model like CLIP should be able to resolve such innovative language use, and it is not feasible to retrain the entire model frequently enough. Thus, if an artificial agent should be able to engage in such a task, it would clearly need to be able to adjust its language-to-vision grounding model, based on a single example. Even if we are not considering completely novel language use, there might be small misalignments between the agent's and the human's language use, or in their perceptual representations (Chai et al., 2016), which could lead to miscommunication if the model is not adapted. Often, the exact meaning of words depends largely on the context and the task at hand.



Speaker	Round	Utterance
B	4	*eh it looks like a blue crab sticking up his claws*
		. . .
A	7	*it's the same the little crab again*

Figure 2: Example of repeated reference across rounds; the referent the participants have to collaboratively resolve is indicated here with a magenta outline (figure from Shore and Skantze (2018)).

The ability to continually learn over time has been a long-standing challenge for machine learning and artificial intelligence, and this area of research has been referred to as *continual* or *lifelong learning* (Parisi et al., 2019). There are several problems involved in this. First, whereas humans can learn new concepts using only a few examples, machine learning models typically need several orders of magnitude more examples. Second, computational models have been shown to be prone to so-called *catastrophic forgetting* (Parisi et al., 2019). Unless the model is re-trained entirely from scratch (which is infeasible for large models like CLIP), the updates to the model's parameters might interfere with previously learned knowledge, resulting in abrupt performance drops. This is also referred to as the *stability-plasticity dilemma* (Parisi et al., 2019).

Learning from few examples greatly depends on having powerful enough representations. Thus, the first problem has been addressed using *transfer learning*, where a fixed *foundation model* (Bommasani et al., 2021) learns rich general representations from other (but related) tasks. This foundation model is then used as input to a simple classifier with only a few parameters (such as logistic regression), requiring only a few training examples. This is often referred to as *few-shot learning* (Wang et al., 2020). Since the CLIP model is trained to learn powerful general representations, it was also shown to be fairly good as a foundation model for few-shot learning (Radford et al., 2021). In principle, an agent that sees a new object and hears it being referred to as "a blue crab" by a human could train a few-shot classifier to be able to identify such objects in the future, as illustrated in Figure 1b. However, a problem with this form of few-shot learning is that it is based on the same principles as the conventional supervised image classification discussed above, where labels do not have any inherent meaning, but are instead treated as atomic symbols. If an agent using a language-image embedding model (such as CLIP) would learn a new label using this approach, it is not clear when it should use its foundation model to resolve language-image relationships (as in Figure 1a), and when it should apply the newly learned classifier for the specific label (as in Figure 1b). Moreover, those new categories would have no relationship to other (previously known or newly acquired) categories. For example, if the agent would learn how the term "the red monitor" is used in a specific situation, it would not be able to infer that "the red display" might be used in a similar way. In addition, it is unclear how it should be able to make use of semantic compositionality (as described above), i.e., to combine (in a principled way) the newly acquired language with the language it already knows in a compositional manner, to understand expressions such as "the blue monitor".

In this paper, we propose **CoLLIE**, a simple, yet effective, model for **Co**ntinual learning of **L**anguage grounding from **L**anguage-**I**mage **E**mbeddings. The general principle of CoLLIE is illustrated in Figure 1c. Instead of learning a new model for each new concept (as in few-shot learning), the model relies on a foundation model of language-image embeddings in a joint embedding space (CLIP in our case), with zero-shot capabilities. We then use and update a separate *transformation model* that makes adjustments to the language embedding to better fit the new concepts that are being learned, when needed. This transformation is done by predicting the *difference vector* between the image and the language embeddings, and then adding this difference vector to the language embedding. This way, only the misaligned dimensions of the language embedding are corrected, while the others are unaffected. Since the continual learning is only taking place in this transformation model, and the foundation model is fixed, this is a very lightweight process. Our aim is to achieve the following characteristics:

- **Sample efficient**: We want to be able to learn new language-image mappings quickly with only a few examples.

- **Computationally efficient**: The transformation model is very lightweight and relatively cheap to retrain.

- **Generalizable**: We want to be able to use the newly learned concepts to understand new related concepts, and make use of semantic compositionality.

- **Robust**: As the model learns new concepts, it should continue to perform equally well on tasks it could do before, and newly learned concepts should not interfere with each other.

2. Related Work

Language grounding is a core problem of AI, and is related to the more general problem of symbol grounding. i.e., how the symbols used by an AI system get their meaning in terms of how they are anchored to the external world (Harnad, 1990). The more specific problem of how language is grounded in vision has been addressed in different fields, including computer vision, computational linguistics, and artificial intelligence. In computer vision, the task of image classification has been studied extensively for a long time (e.g., Deng et al., 2009). This is analogous to the problem of image retrieval, where images are ranked based on how well they match a certain class. In this formulation of the problem, each image is classified as belonging to a fixed number of classes. However, in real language use, language exhibits semantic compositionality and can express an almost infinite number of "classes" by combining different concepts, such as "the black cat on the mat". The problem of identifying the target (or referent) of such expressions has been referred to as *referring expression comprehension* (Qiao et al., 2021). As discussed in Section 1, this is typically done by encoding the image and text into a joint semantic embedding space, and rank the target referents according to how close they are in this space (ibid.). In (computational) linguistics, the problem of identifying referents of referring expressions in the external world is called *exophoric* reference resolution, which is different from *anaphoric* reference (or coreference) resolution, where references to entities in the past discourse are being resolved.

The inverse problem, how to generate referring expressions based on the visual properties of the target referent, has also received considerable attention, earlier using rule-based approaches (Krahmer & van Deemter, 2012), and more recently using neural language generation conditioned on the visual encoding of the image (e.g., Panagiaris et al., 2021). A challenge when generating referring expressions is that the model should ideally take the potential distractors into account (in order to uniquely identify the target), while at the same time being as efficient (brief) as possible (the Gricean Maxim of Quantity (Grice, 1975)) (Krahmer & van Deemter, 2012). A related problem is that of image captioning, where the task is to describe an entire image or scene, rather than a specific object in an image (e.g., You et al., 2016).

When it comes to the visual grounding of language in interaction, there has also been a lot of research done in the areas of Visual Question-Answering (VQA) and multi-turn VQA, where visual language understanding and generation are combined and treated in an end-to-end fashion (Kafle & Kanan, 2017; Das et al., 2017). Another field where this problem has been studied is human-robot interaction (e.g., Chai et al., 2016). However, these studies typically assume that a fixed model of language grounding can be trained, and that the language use does not change after that. As discussed in Section 1, the grounding of language is often (explicitly or implicitly) negotiated in dialog to handle new or specific situations; new words might be invented, or the exact meaning of words might change.

One approach to accommodate partner-specific language use that evolves in dialog is to feed the dialog history as input to the model. An example of this is Takmaz et al. (2020), who trained a model to generate subsequent referring expressions, conditioned on past coreference chains in the dialog. However, this is only feasible for short-term effects, and not for language use that evolves over longer periods of time. For that, the parameters of the model need to be updated (i.e., some form of continual learning). Shore and Skantze (2018) explored the accommodation of partner-specific language use in exophoric reference resolution for the game depicted in Figure 2. They showed that if the reference resolution model was retrained after each round (in light of new data), the performance increased significantly. While this was feasible to do given the small size of the model and the limited domain, it is clearly not feasible for real use case scenarios, where models like CLIP are trained on 400 million data points. For such cases, some form of continual learning is needed.

Previous research on continual learning has mainly been done in the context of image classification, where there is a limited set of classes, but where new classes are gradually added to the model, so-called "incremental class learning" (e.g., Rebuffi et al., 2017; Kemker & Kanan, 2018; Kemker et al., 2018). Early studies of continual learning in neural networks showed that the newly learned information interfered with previous knowledge in the shared representational resources, resulting in catastrophic forgetting (McCloskey & Cohen, 1989). Parisi et al. (2019) outline three basic approaches to alleviate catastrophic forgetting for continual learning in neural networks: First, various *regularization approaches* may be used to impose constraints on the update of the model's parameters. Second, it is possible to allow the architecture of the network to change, for example, by adding neurons or layers. Third, *complementary learning systems* are inspired by the human brain, in that they rely on an interplay between episodic memory (specific experience) and a semantic memory (general structured knowledge), where learning first happens in the former, and is eventually consolidated with the latter (during "sleep"). One approach to avoid catastrophic forgetting is to store some of the older data points and mix these in when training with newer data points, a technique called "rehearsal" (Parisi et al., 2019). A variant of this is "pseudo-rehearsal", where a generative model is used to generate older data points (Kemker & Kanan, 2018).

CoLLIE does not fit squarely into any of these three approaches, but comes closest to complementary learning in its use of a foundation model (where parameters are fixed) and a dynamic model (where learning happens). It should be noted that our work is a bit different from how continual learning is typically addressed and how the problem is typically formulated. In our case, we do not have a limited set of classes which is expanded during training. Instead, we assume a language-image embedding model (such as CLIP) that can be used for zero-shot reference resolution, and where the language can form a virtually endless number of "classes" (i.e., the same way the problem is formulated in referring expression comprehension). Our task is then to adjust the model to learn a domain-specific *language use*, while retaining the zero-shot performance of the model on the language use it was trained for. It should be stressed that this assumes that the model already has good representations of the visual domain, and the aim of the learning is not to improve those representations, but rather to learn how to better map those to new language use. Thus, the performance of CoLLIE is inherently constrained by the performance of the foundation model.

While continual learning should ideally happen without any retraining/rehearsal and without keeping training data in memory, we accept keeping the newly learned examples in memory (and a small fixed set of negative examples, as we will see), since we do not have to keep the training data for the foundation model in memory. This way, only the transformation model needs to be re-trained using those new examples. It should also be noted that the parameters of the transformation model are relatively few and the footprint of the training samples is quite small (since we only store their embeddings). If we can achieve the sample efficiency objective, they should also be limited in quantity.

3. Data, Task and Metric

For our evaluations, the task is that of image retrieval or referring expression comprehension (Qiao et al., 2021), i.e., to rank a set of candidate referents based on how well they match a referring expression. While referring expression comprehension is typically done on objects within images (ibid.), we use separate images for the object here, in order to more easily control the set of distractors and evaluate the performance based on that. As our metric, we use the Mean Reciprocal Rank (MRR), which is equal to 1 divided by the assigned rank of the correct candidate, yielding a score between 0 and 1. Thus, an MRR of 1 corresponds to ranking the correct candidate first and 0.5 corresponds to ranking it second (which can still be considered quite good if the number of candidates is large). The reason we choose MRR instead of accuracy is that it does not only take the top-ranked candidate into account, and therefore can be considered to be a more nuanced metric. Another metric that is sometimes used for similar tasks is Recall@K. However, we think that metric is better suited when there are several potential targets, which is not the case in our experiments.

In this paper, we use two datasets. First, we use the **LAD dataset** (Large-scale Attribute Dataset) by Zhao et al. (2018), from which we selected a set of 200 categories belonging to the super-categories animals, fruits, electronics, and vehicles, with a total of 68,247 images. To verify CLIP's zero-shot performance on this dataset[1], we did 20 iterations where we randomly selected one image per category (i.e., 200 images) and performed the ranking task using the LAD labels of the categories as referring expressions, yielding an MRR of 0.773. We think this confirms CLIP's impressive zero-shot performance on these types of images.

To study a more challenging set of images (for CLIP), we also use the images from the **KTH Tangrams dataset** (Shore et al., 2018), which were used for the task depicted in Figure 2. To assess CLIP's zero-shot performance on these tangram figures, we took the 17 shapes used in the study and picked a subset of five colors (red, green, blue, yellow, and purple), constituting a set of 85 candidate referents. The referring expressions were constructed by combining the color with the name of the shape used by the authors of the paper (e.g., "the blue giraffe"). As expected, CLIP's zero-shot performance on these referents is not as good, only yielding an overall MRR of 0.310. (The MRR for the individual shapes are shown in Figure 10). While some shapes are identified correctly ("mountain", "barn"), most of them are not. This is of course understandable, given that these images are not representative of CLIP's training data.

In fact, it was not that easy for the human participants in the experiment to do this task either, at least not for their initial attempts. However, as discussed in Section 1 and shown in Figure 2, they soon started to invent names for the different shapes, forming conceptual

[1] For the experiments in this paper, we use the publicly released pre-trained CLIP model with the ViT-B/32 Vision Transformer architecture, unless otherwise stated (`https://github.com/openai/CLIP`).

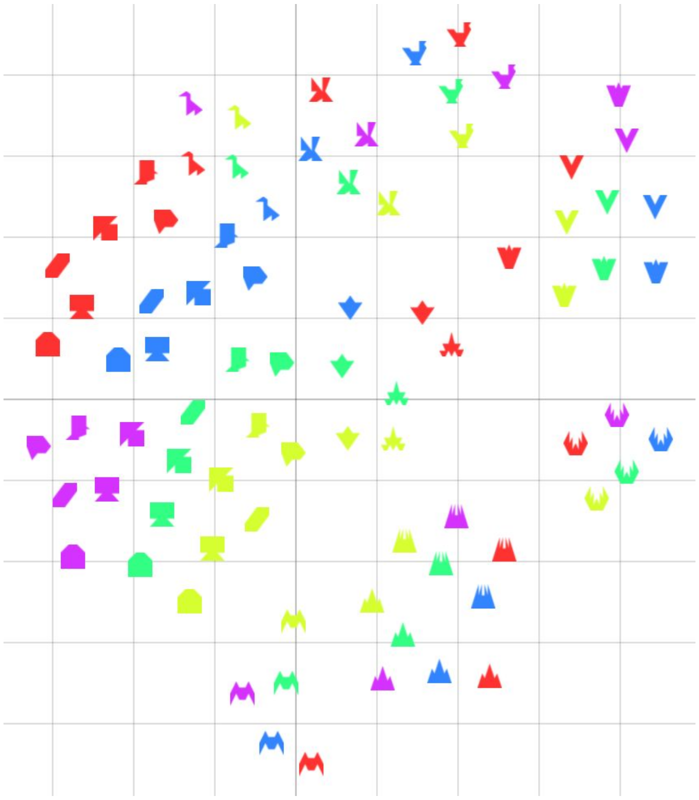


Figure 3: t-SNE dimensionality reduction of the colored tangram CLIP embeddings.

pacts after repeated interactions and making the interactions more efficient over time (Shore & Skantze, 2018). If an artificial agent should be able to engage in such a task, it would clearly have to be able to apply some form of continual learning in the way outlined in the introduction. For this to work, CLIP still needs to have a good representation of the images. To investigate whether this is the case, we performed a t-SNE analysis (Van Der Maaten & Hinton, 2008) on the CLIP embeddings of the colored tangram images to reduce the 512 dimensions to 2 dimensions, as illustrated in Figure 3. As can be seen, the shapes and colors seem to form clusters and to be handled in a somewhat consistent fashion, which suggests that it might be possible to associate them with names. We will discuss this task in more detail in Section 5.

4. The CoLLIE Transformation

The idea behind CoLLIE is to learn a **transformation function**, $T' = f(T) : \mathbb{R}^{512} \rightarrow \mathbb{R}^{512}$, which takes the CLIP embedding of the text, T, and returns another transformed embedding, T', that better represents the new language use, and is closer to the CLIP image embedding I, as illustrated in Figure 4. It is important to note that in order to retain the zero-shot performance, f should in most cases return a similar output as input, unless the text has a domain-specific meaning that the model should correct for.

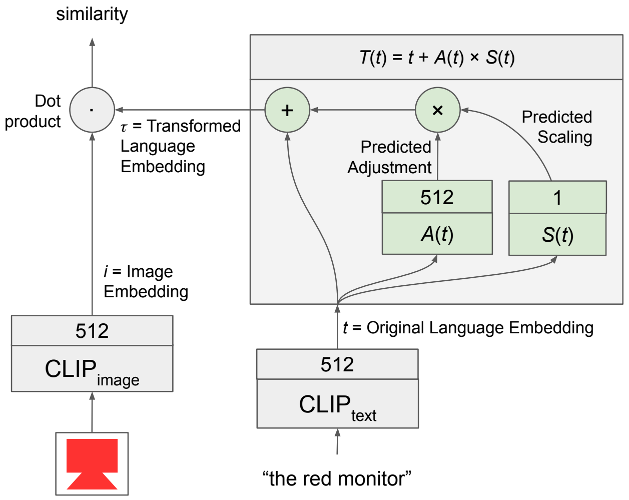


Figure 4: The CoLLIE transformation.

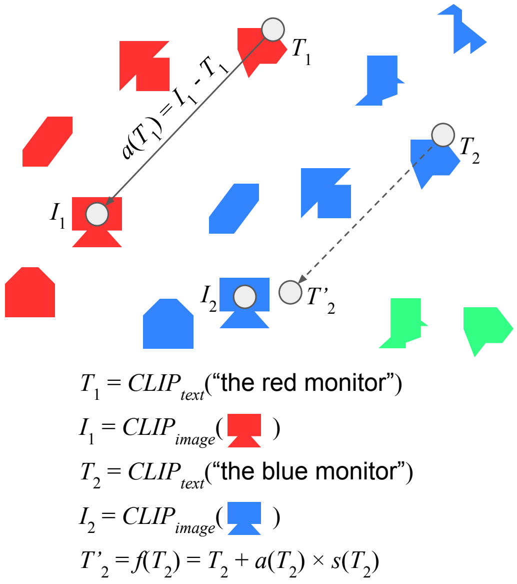

$a(T_1) = I_1 - T_1$
T_1
T_2
I_1
I_2
T'_2
$T_1 = CLIP_{text}$("the red monitor")
$I_1 = CLIP_{image}($ $)$
$T_2 = CLIP_{text}$("the blue monitor")
$I_2 = CLIP_{image}($ $)$
$T'_2 = f(T_2) = T_2 + a(T_2) \times s(T_2)$

Figure 5: A principled illustration of the intuition behind CoLLIE.

The transformation function is modelled as $f(T) = T + a(T) \times s(T)$, where $a(T) : \mathbb{R}^{512} \rightarrow \mathbb{R}^{512}$ is an **adjustment function**, and $s(T) : \mathbb{R}^{512} \rightarrow [0, 1]$ is a **scaling function**. As we will see, this scaling function helps to retain the zero-shot performance of the model. It can be noted that this principle is similar to that of residual connections in neural networks (He et al., 2016) and Gated Linear Units (Dauphin et al., 2017).

Training examples are stored as pairs of text and image embeddings $\langle T, I \rangle$, and thus have a limited footprint (512+512 floats). $a(T)$ is then trained to predict the *difference vector*, $I - T$, using the accumulated training examples. When doing the adjustment, the predicted difference vector is then added to T. The reason for predicting the difference vector (rather than I directly) is that we want to learn which dimensions need to be corrected, while not affecting the dimensions that are already correct. We learn A using linear regression: $a(T) = \beta T + m, \beta \in \mathbb{R}^{512 \times 512}, m \in \mathbb{R}^{512}$. To avoid overfitting (given the limited number of training examples) we use ridge regression (L2 regularization with $\lambda = 0.001$). We will return to this and evaluate alternatives in Section 5.

The objective of the scaling function, s, is to return a value close to 1 when the input is a text that should be transformed (i.e., close to any example in the training set), and close to 0 otherwise. We learn s using a regression model, where the accumulated training examples are used as positive examples (with training target 1). As negative examples (with training target 0), we simply use a list of the 1,000 most common nouns in English (representing expressions that should not be transformed). The rationale for using nouns is that all referring expressions are expected to have at least one head noun (since they are noun phrases), and that these common nouns should cover the embedding space fairly well. For example, the embedding for the word "shoe" should be fairly similar to "the large shoe" or "the shoe with laces". As we add positive training examples, we create exceptions for nouns (and noun phrases) that should be adjusted. For our initial tests, we learn s using support vector regression (SVR) with a linear kernel (coerced in the range $[0, 1]$), but we will evaluate alternative models in Section 5.

Figure 5 illustrates the intuition behind the model: Given that we have a reference to an image ("the red monitor"), we encode it using CLIP and get an embedding T_1. As can be seen, in this case, the text embedding is not very close to the embedding of the corresponding image I_1, and will thus retrieve the wrong referent. To teach the model to make better predictions in the future, we add the pair $\langle T_1, I_1 \rangle$ as a training example for a and $\langle T_1, 1 \rangle$ as a training example for s. Using the accumulated training examples, we train a to approximate the difference vector between the embedding of the image and the text $(I_1 - T_1)$, and s to predict a value close to 1 for the input T_1. Recall that we retrain only the added transformation and scaling functions, while the CLIP weights remain frozen. Now, when a new referring expression is to be resolved, "the blue monitor", the expression is encoded by CLIP into T_2. Again, directly using this embedding would result in a poor match for this domain. If we now apply the learned adjustment function $a(T_2)$, it is likely to return a similar vector as $I_1 - T_1$ (given that T_2 is relatively close to T_1 and that we do not have any other, more similar, training examples). Similarly, $s(T_2)$ is likely to return a value fairly close to 1 (given that T_2 is fairly close to T_1). We now get a new vector $T'_2 = T_2 + a(T_2) \times s(T_2)$, which is indeed closer to the true referent I_2. Thus, while the color dimensions seem to align well between T_2 and I_2, the predicted adjustment (the difference vector) corrects for the misaligned shape dimensions.

5. Experiments

To evaluate CoLLIE, we devised two experiments. In the first experiment, we test whether the model can learn novel pseudo-words for already known objects (the LAD dataset). The focus of this experiment is to ensure that the model can generalize to other images of the same object and that the model can retain its zero-shot performance. In the second experiment, we test whether the model can learn to ground known words to novel objects (the KTH Tangrams dataset). Here, we want to ensure that the model can utilize semantic compositionality and generalize to similar language use.

5.1 Experiment I: Learning Pseudo-words for Realistic Objects

We first devised an evaluation scheme to see whether the model can learn new words for photographic images from the LAD dataset, while we monitor the retained zero-shot performance during training (i.e., to make sure that the model still understands the existing words). We randomly select a set of N categories, C_{train}, (out of the 200 categories) for which we want to teach the model new names. We then assign a new name for each of these N categories, using randomly selected pseudo-words from the Novel Object and Unusual Name (NOUN) Database ("boskot", "derd", "tust", etc.) (Horst & Hout, 2016). Training is then performed over five and testing is performed over six *rounds* (the initial round of testing is performed without training, which reflects the model's zero-shot performance). At the beginning of each round, we randomly select one image for each of the 200 categories, without ever reusing images between rounds. We then let the model rank the 200 images as potential referents for each pseudo-word, and the MRR is computed as described

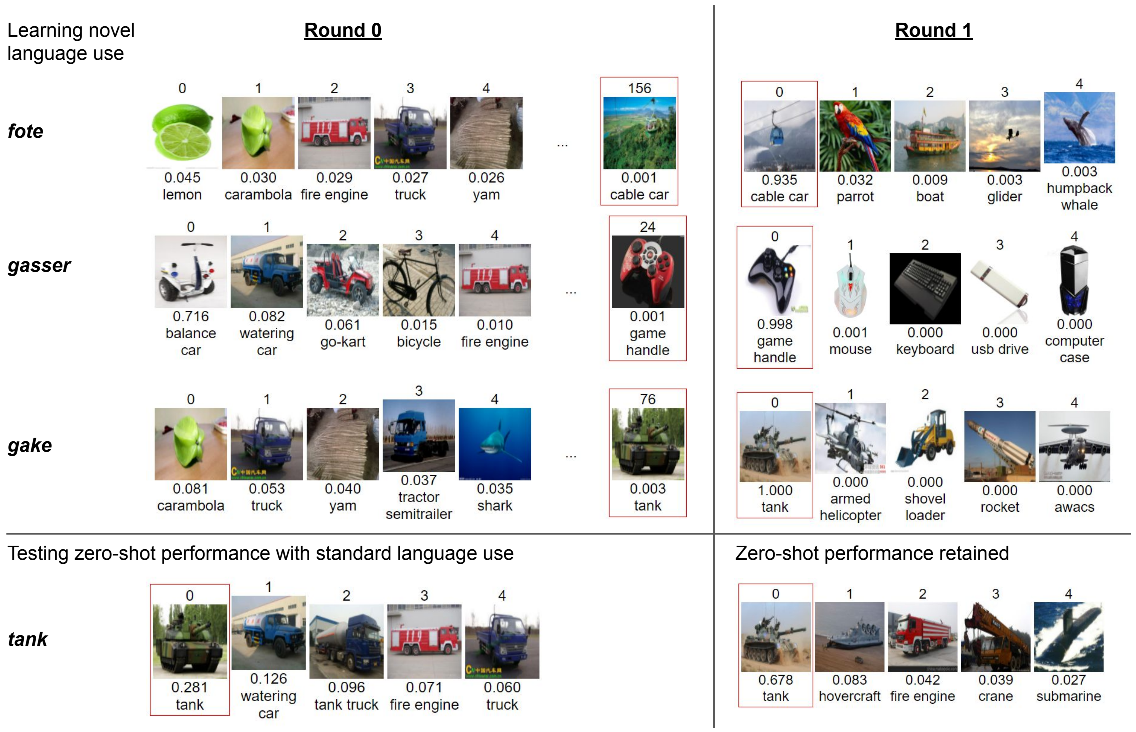


Figure 6: An example from the first two rounds of Experiment I. The referring expression is shown to the left and the correct target is marked with a red square. At round 0 (before any training), CoLLIE obviously doesn't recognize the new words, and they are ranked very low. At round 1 (after having seen just one example of each new word), it correctly ranks them first. The zero-shot performance of the model (identifying the referents of the original names) is unaffected.

in Section 3. At the end of each round, we add the images from C_{train} and their associated pseudo-words as training examples (i.e., one example per category) to the model, and retrain it. An example of the first two rounds is shown in Figure 6. This whole procedure is repeated over 50 *iterations* (with new pseudo-words and categories randomly selected and assigned), in order to get a smooth average performance per round.

We evaluate and compare the performance using (1) the CoLLIE model, (2) the fixed CLIP model, and (3) a few-shot classifier based on logistic regression (implemented in the same way as in Radford et al., 2021). For the few-shot classifier, each pseudo-word is treated as a class. To study the effect of the scaling function, we also add (4) the CoLLIE model without the scaling function.

The results are shown in Figure 7(A), where $N = 50$. As can be seen, CoLLIE quickly learns the new pseudo-words, and reaches a fairly good performance (0.618) already after one round (i.e., when it has only been provided with one example per category), increasing to 0.750 at the final round (where five examples have been provided), which is quite close

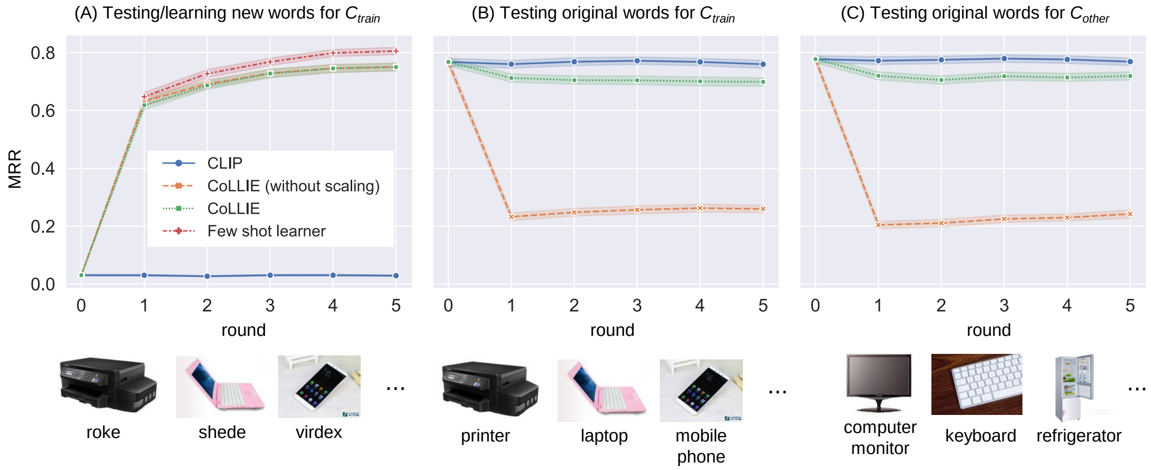


Figure 7: Performance of CoLLIE (with SVR scaling function) over five rounds of training on the LAD dataset (averaged over 50 iterations, 95% CI), where $|C_{train}| = 50$. One training example per category is added per round. (A) shows continual learning performance. (B) and (C) show retention performance. Few-shot learner is only applicable for pane (A).

to CLIP's zero-shot performance of 0.773 for the original words on this dataset. Here, the scaling function has very little effect. Since the CLIP model is not doing any learning, it obviously has a very poor zero-shot performance on these new words. However, the few-shot classifier has a slightly better performance than CoLLIE, especially after five examples are added (0.805). This is perhaps not very surprising, given that it is optimizing this classification task, rather than transforming the embedding space. Also, in this experiment, CoLLIE does not benefit from generalization of the learned words, as they are arbitrarily assigned and there is no semantic compositionality effect (which we will get back to in Experiment II).

To study the retained zero-shot performance of the model during training, we also plot the performance of the models when using the original words for the 50 categories in C_{train}, in Figure 7(B). As can be seen, the original names for those categories can still be resolved by CoLLIE with a slight (but not catastrophic) drop in performance compared to the static CLIP model (0.699 vs. 0.760 at the final round), even though the model has also learned new words for them. In this case, the scaling function is important and without it, the performance drops considerably (to 0.260). Similarly, for each round, we also study the models' retained zero-shot performance on 50 randomly selected categories, C_{other}, which were not part of C_{train}, using their original names. This is shown in Figure 7(C). Again, when using the scaling function, CoLLIE does not seem to interfere much with CLIP's original zero-shot performance (0.719 vs. 0.769 at the final round), while there is a drastic drop in performance (to 0.242) when the scaling function is not used. Since the few-shot learner cannot be applied to these problems, its performance is not plotted in pane B-C.

Table 1: Performance of the model (MRR) on the LAD dataset (averaged over 50 iterations), with different implementations of the scaling function, where $|C_{train}| = 50$.

	Learning new words for C_{train}		Testing original words for C_{train}		Testing original words for C_{other}	
Round	1	5	1	5	1	5
SVR (RBF)	0.639	0.743	0.731	0.739	0.711	0.722
SVR (sigmoid)	0.582	0.672	**0.764**	**0.765**	**0.749**	**0.752**
SVR (linear)	0.627	0.723	0.743	0.749	0.726	0.732
Logistic regression	0.633	**0.756**	0.656	0.679	0.649	0.657
Linear regression	0.648	0.755	0.659	0.674	0.636	0.656
KNN (K=10)	**0.650**	0.755	0.762	0.763	0.744	0.748
No scaling	**0.650**	0.755	0.240	0.270	0.210	0.243
No scaling, negative examples in adjustment function	0.541	0.728	0.563	0.416	0.552	0.391

5.1.1 Effects of Scaling and Adjustment Functions

As these results show, the scaling function is important for the retention of the model's zero-shot performance. As mentioned earlier, we initially used SVR (with a linear kernel) for the scaling function, as it was showing promising performance. In Table 1, we also investigate different implementations of the scaling function[2]: using different SVR kernels (RBF, sigmoid and linear), linear regression, logistic regression, and a KNN regressor (with a weighted distance function and $K = 10$). As can be seen, when taking both continual learning and retention performance into account, SVR (with either RBF or linear kernel) compares favorably compared to logistic and linear regression. The KNN regressor also shows a good performance. One explanation why KNN and SVR classifiers might perform better is that they both give more weight to the nearest neighbors, and handle class imbalance better (between the newly learned words and the negative examples). For a good scaling function, we want it to output a score close to 1 when the expression is part of the newly learned words, and close to 0 otherwise. For this, the closest words in the training set should be the most informative ones.

A potential alternative to a separate scaling function is to instead use the negative examples (common nouns) as training examples for the adjustment function, with zero-length vectors as targets. This could potentially allow the adjustment function to directly learn that no adjustment should be applied for those examples. This is shown in the last row of Table 1. As can be seen, the performance is much lower (especially when it comes to retaining zero-shot performance), which motivates the need for a separate scaling function.

[2] We use the implementations in scikit-learn (Pedregosa et al., 2011) with standard parameters. The standard error of the mean is consistently around 0.007, and is therefore not reported in the table.

Table 2: Performance of the model (MRR) on the LAD dataset (averaged over 50 iterations), with different implementations of the adjustment function, where $|C_{train}| = 50$.

	Learning new words for C_{train}		Testing original words for C_{train}		Testing original words for C_{other}	
Round	1	5	1	5	1	5
Linear	**0.640**	**0.744**	0.729	0.148	0.708	0.0825
Ridge ($\lambda = 0.0001$)	**0.640**	**0.744**	0.730	0.738	0.708	0.722
Ridge ($\lambda = 0.001$)	**0.640**	0.743	0.731	0.739	0.711	0.722
Ridge ($\lambda = 0.01$)	0.625	0.740	0.741	0.740	0.724	0.726
Ridge ($\lambda = 0.1$)	0.474	0.702	**0.758**	**0.751**	**0.743**	**0.740**

In Table 2, we show a similar comparison, but with different variants of the adjustment function (using SVR with RBF kernel for the scaling function). For the continual learning of the new words, both linear and ridge regression have similar performance, as long as the regularization in the ridge regression is not too strong (i.e., λ is set too high). For the retention performance (testing existing words), the simple linear regression results in catastrophic forgetting as more training examples are added (see rightmost column), while the ridge regression keeps the performance at an acceptable level. Thus, we conclude that ridge regression with $\lambda = 0.001$ gives a fairly good trade-off.

5.1.2 Effects of Training Size

To further investigate the performance, we also run experiments with different numbers of classes/pseudo-words to learn ($N = |C_{train}|$), and different numbers of negative examples in the scaling function. The results are shown in Table 3. As can be seen in (A), the continual learning performance is relatively stable for different values of N, and the choice of scaling function or number of negative examples has no large impact. However, as seen in (B), the zero-shot retention performance is clearly affected as N increases. This is especially true when only 100 negative examples are used for the scaling function. Thus, it is possible that the drop in retention performance could be mitigated by adding even more negative examples, as N increases. As the number of pseudo-words to learn increases, the KNN regressor also shows a clearly better performance than SVR, in terms of retention. Investigating suitable scaling functions for large values of N is an important topic for future work.

5.2 Experiment II: Learning Language for Tangram Figures

The biggest expected benefit of CoLLIE comes from its ability to generalize from the language it is learning, for example through semantic compositionality, which was not addressed in Experiment I. We thus devised an evaluation scheme to see how quickly the model can learn to identify the colored tangram shapes (introduced in Section 3), for which it clearly

Table 3: Performance of the model (MRR) on the LAD dataset (averaged over 50 iterations), with different implementations of the scaling function, number of negative examples (n/e), and number of classes/pseudo-words to learn (N). (A) shows continual learning performance and (B) shows retention performance.

$N = \|C_{train}\|$	10	50	100	150	10	50	100	150
(A) New words	C_{train}: After 1 example				C_{train}: After 5 examples			
SVR (1,000 n/e)	0.616	0.618	0.636	0.634	0.770	0.750	0.746	0.751
KNN (1,000 n/e)	0.646	**0.648**	0.637	0.635	0.738	**0.758**	**0.749**	0.747
SVR (100 n/e)	0.639	0.629	0.639	0.636	0.770	0.750	0.746	0.752
No scaling	**0.654**	0.632	**0.641**	**0.637**	**0.774**	0.751	0.746	**0.753**
CLIP baseline	0.033	0.030	0.030	0.028	0.028	0.029	0.030	0.031
(B) Orig. words	C_{train}: After 5 examples				C_{other}: After 5 examples			
SVR (1,000 n/e)	0.780	0.753	0.633	0.529	0.774	0.725	0.645	0.522
KNN (1,000 n/e)	0.778	**0.767**	0.707	0.641	**0.779**	0.743	0.707	0.631
SVR (100 n/e)	0.769	0.621	0.450	0.333	0.732	0.638	0.443	0.314
No scaling	0.515	0.260	0.127	0.078	0.361	0.242	0.112	0.054
CLIP baseline	**0.798**	0.760	**0.768**	**0.773**	**0.779**	**0.769**	**0.783**	**0.774**

had a very poor zero-shot performance. Here, we expect the model to benefit from the compositionality of the referring expressions. As discussed earlier, given that it has learned what visual properties to associate with the phrase "the blue rock", it should be able to generalize this understanding to "the red rock".

Again, the task is to rank the 85 potential referents, given a referring expression. Similar to Experiment I, the model starts out with no training examples (round 0). We then train the model over 30 *rounds*. In each round, one random referent is picked, the model's performance (in terms of MRR) on this referent is assessed, the image-text pair of the referent is added to the training set, the model is retrained, and a new round begins. This whole procedure is then repeated over 3,000 *iterations*, resetting the model after each iteration. An example of the first few rounds in one iteration is shown in Figure 8.

The MRR per round (over all iterations) is illustrated in Figure 9. We do a similar comparison with other models as in Experiment I. Here, we let the few-shot learner (again, a logistic regression classifier) fall back on CLIP when it is presented with a referring expression it has not seen before. We then add the new referring expression as a new class for the few-shot learner, and retrain it.

As can be seen, CoLLIE (with SVR scaling function) quickly learns the names for the tangrams, reaching an MRR of 0.860 after 30 rounds. Note that the 85 images are all unique in terms of shape-color combinations, which means that the model must be able to generalize in order to achieve this performance. In contrast, the few-shot learner has much worse performance, as new referring expressions (classes) are introduced in most rounds in the beginning (unless the exact same object happened to be picked twice), and it has no way of generalizing from already learned classes.

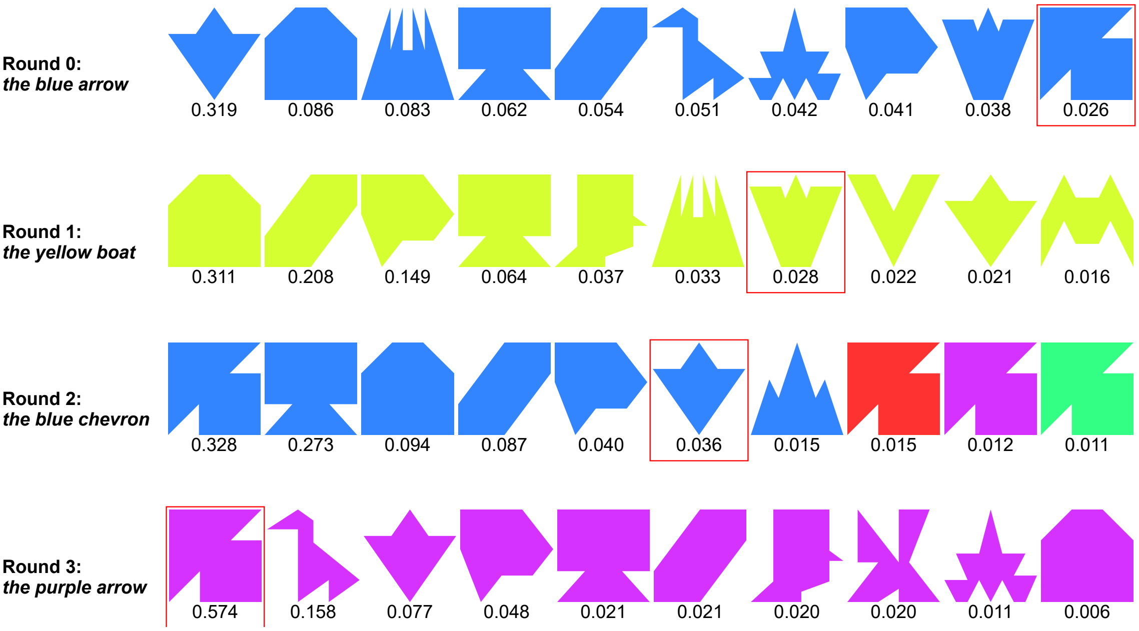


Figure 8: An example of the first four rounds of one iteration of Experiment II. Each round shows the randomly selected referring expression, the top 10 candidates as ranked by the model left-to-right (with their softmax scores), as well as the correct referent (marked with a red square). In round 0–2, new names of shapes are introduced, but after having seen one example of a “blue arrow” in round 0, the model correctly identifies the “purple arrow” in round 3.

This supports the hypothesis that CoLLIE should be able to benefit from the compositionality of language: After being taught what a “red giraffe” looks like, CoLLIE is now better at identifying a “blue giraffe”, combining the base representation of “blue” with the learned meaning of “giraffe”. To further confirm this ability, we also performed an experiment where we first train the model on all 17 shapes of one random color, and then evaluate it on the same shapes with different random colors. This was iterated 100 times. Whereas the CLIP baseline model (and the few-shot learner, which has to fall back on the CLIP model) only had an average MRR of 0.317 on these unseen combinations, CoLLIE achieved an MRR of 0.857.

The intuition behind why this works was illustrated in Figure 5: CoLLIE learns to predict the *difference vector* that needs to be applied. Thus, if the color dimensions in the CLIP embedding were already aligned between the language and the image, there will not be any need to adjust those dimensions – it is only the dimensions related to the shape that need to be adjusted. The fact that this works despite CLIP’s representation being entirely distributed is interesting. The steady improvement also indicates that the learning of each concept does not interfere with the learning of other concepts. However, as can be seen in Figure 9, without the scaling function, the model has a drastic drop in performance

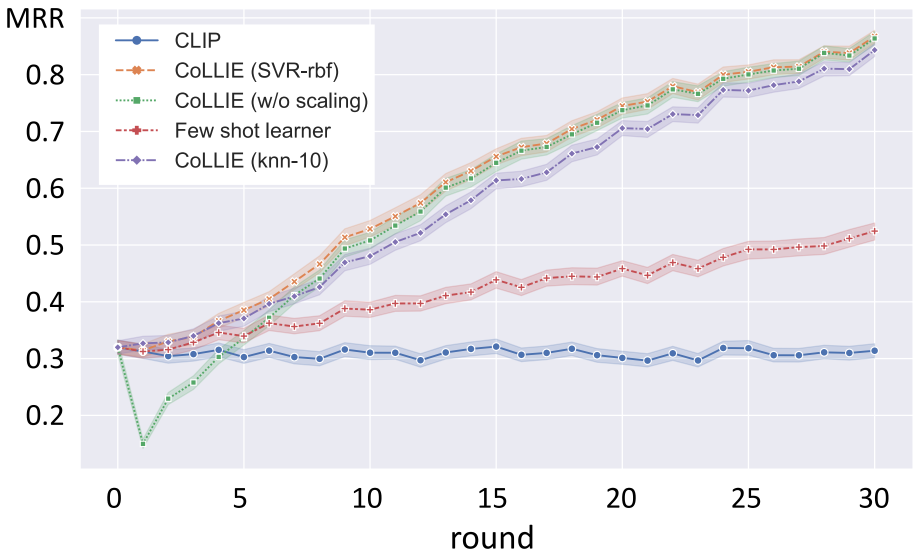


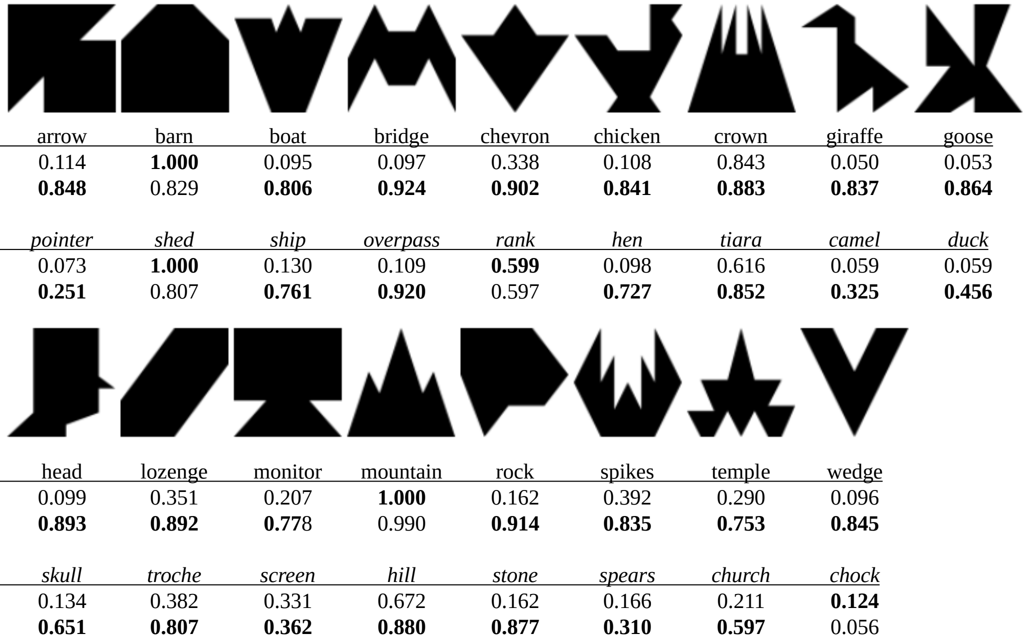
Figure 9: Performance of the model over 30 rounds of training (averaged over 3,000 iterations) when training on the colored tangrams (95% CI). One new training example is added per round.

for the first rounds, which is likely because the newly learned adjustments are added too generously to unrelated referring expressions.

Figure 9 also shows a comparison between the SVR and the KNN scaling functions. Unlike in Experiment I, SVR here gives a slightly better performance than KNN. This can perhaps be explained by the fact that, in this experiment, the scaling function has to give a high score to similar, but not identical, referring expressions, in order to generalize the adjustments it has learned.

5.2.1 Generalizing with Synonyms

As a further (limited) test to verify the model's ability to generalize, we substituted the names of the tangrams with synonyms[3] ("barn"→"shed", "chicken"→"hen", etc.). This way, we formed referring expressions such as "the blue hen". Using the CoLLIE model trained for 30 rounds as described above, we then evaluated these expressions (over all 3,000 iterations). The MRR for these was 0.602, which is clearly better than the baseline of 0.290 (using CLIP directly), providing further evidence for the model's ability to generalize. Given that many of the names had no obvious synonyms, their individual performance varied greatly (MRR 0.056-0.920). A breakdown of these results can be found in Figure 10.



Shape	arrow	barn	boat	bridge	chevron	chicken	crown	giraffe	goose
CLIP	0.114	**1.000**	0.095	0.097	0.338	0.108	0.843	0.050	0.053
CoLLIE	**0.848**	0.829	**0.806**	**0.924**	**0.902**	**0.841**	**0.883**	**0.837**	**0.864**

Synonym	*pointer*	*shed*	*ship*	*overpass*	*rank*	*hen*	*tiara*	*camel*	*duck*
CLIP	0.073	**1.000**	0.130	0.109	**0.599**	0.098	0.616	0.059	0.059
CoLLIE	**0.251**	0.807	**0.761**	**0.920**	0.597	**0.727**	**0.852**	**0.325**	**0.456**

Shape	head	lozenge	monitor	mountain	rock	spikes	temple	wedge
CLIP	0.099	0.351	0.207	**1.000**	0.162	0.392	0.290	0.096
CoLLIE	**0.893**	**0.892**	**0.778**	0.990	**0.914**	**0.835**	**0.753**	**0.845**

Synonym	*skull*	*troche*	*screen*	*hill*	*stone*	*spears*	*church*	*chock*
CLIP	0.134	0.382	0.331	0.672	0.162	0.166	0.211	**0.124**
CoLLIE	**0.651**	**0.807**	**0.362**	**0.880**	**0.877**	**0.310**	**0.597**	0.056

Figure 10: Performance (MRR) on individual tangram shapes and their synonyms. Both the original zero-shot performance of CLIP (over the 85 candidates), and the performance of CoLLIE (after training on 30 examples with the original names), are shown.

Table 4: Performance of CoLLIE (with SVR scaling function) on Experiment II (tangram figures), with different vision backbones in the CLIP model. Average results (MRR) over 3,000 iterations are shown. The rightmost column shows the number of dimensions in the CLIP embeddings.

Round	0 (CLIP)	1	5	20	30	CLIP Dim.
ResNet-50	0.300	0.269	0.355	0.630	0.848	1024
ResNet-101	0.293	0.305	0.407	0.680	0.860	512
ResNet-50x4	0.237	0.269	0.367	0.658	0.849	640
ViT-B/32	0.321	0.307	0.386	0.661	0.857	512

5.2.2 Effects of Different Vision Backbones

So far, we have used the CLIP model with the ViT-B/32 vision transformer architecture to encode images. To test whether CoLLIE can also work with different variants of the foundation model, we did an experiment where we used different image encoders in the CLIP

[3] Taken from `thesaurus.com`

model (all part of the public release). Not only do these different models have different vision architectures, they also produce text and image embeddings with different dimensionalities. Apart from the vision transformer, there are also three variants of ResNet (He et al., 2016): ResNet-50, ResNet-101, and an EfficientNet-style model scaling using approximately 4x the compute of a ResNet-50 (ResNet-50x4). The performance on the tangram task with these different models is shown in Table 4. As can be seen, although the CLIP zero-shot performance with the ResNet-50x4 model is lower, the performance of CoLLIE seems to be relatively unaffected by the choice of vision backbone.

6. Discussion

Whereas most previous studies on continual learning have focused on incremental class learning in image classification (e.g., Rebuffi et al., 2017; Kemker & Kanan, 2018; Kemker et al., 2018; Benavides-Prado et al., 2020), we have addressed a somewhat different problem. In our case, we rely on a foundation model capable of zero-shot image retrieval, where there is not a finite set of classes to learn. Instead, such a model encodes the image and language into a joint embedding space, in order to assess how well they match semantically. The advantage of such an approach is that we can resolve a virtually endless number of referring expressions, exhibiting semantic compositionality, such as “the red barn”. This is also the approach that is typically taken to address the problem of referring expression comprehension (Qiao et al., 2021). However, whereas previous studies have assumed that language use is fixed, we address the problem of how to accommodate new language use through continual learning.

We have proposed to learn a transformation function that adjusts the language embedding, when needed, to be closer to the image embedding of the intended referent. To the best of our knowledge, this problem of continual adjustment of language-image embeddings to learn new language grounding has not been addressed before. Thus, we do not have any results from prior work to compare our performance with. However, we hope that this work can serve as a benchmark for future studies and alternative models.

Returning to the four characteristics we aimed to achieve, we think that the model has shown to be **sample efficient**, as it seems to reach fairly high performance with only one training example per new category. Second, Experiment II showed that the model was able to **generalize** from the newly learned language use, thanks to the semantic compositionality of the referring expressions, but we also saw indications that it could understand synonymous expressions to an extent. Of course, these experiments were limited to shorter expressions and more basic forms of semantic compositionality; future work should investigate to what extent the model can handle more complex constructions. Third, Experiment I showed that the model was fairly **robust**, as despite a slight drop in the model’s original zero-shot performance, it did not exhibit catastrophic forgetting. Finally, the model is fairly **computationally efficient**; the transformation function uses very simple models with few parameters and the stored training examples have a very small footprint. It can thus be updated quickly on a fairly basic CPU[4]. Nevertheless, since the transformation model needs to be retrained when new examples are added, there are limits to its scalability. Whether this

[4] On an Intel Core i7-1065G7 CPU, one iteration of Experiment II (i.e., 30 model updates) takes about 1 second for the standard CoLLIE model.

is a problem, however, depends entirely on the use case scenario. Regardless, the continual learning of the transformation function without storing examples is also an interesting topic for future work.

As we have seen, the scaling function plays a very important role in retaining the model's zero-shot performance, making sure that only the newly learned terms are adjusted. However, given how the scaling function was trained here (simply using common nouns as negative examples), this will not always work, and we therefore still saw a slight drop in zero-shot performance, especially as the number of new concepts to be learned increases. The scaling function could of course be more or less restrictive. For example, it could require an exact match with a training example to set the scale to 1, and 0 otherwise. This would retain all of the zero-shot performance, at the expense of being able to generalize the learning to similar language use. Exploring more sophisticated scaling functions that provide a good balance between retention and generalization is an interesting topic for future work. For example, Shin et al. (2017) explore the use of a generative model to generate samples for rehearsal, which alleviates the need for storing training examples.

Of course, CoLLIE's performance also relies on the foundation model (CLIP in our case) already having good representations, and thus it is limited by the performance of this model to accurately represent the landscape of visual properties of objects. As pointed out by Radford et al. (2021), CLIP's representations are limited in certain aspects, including counting objects in an image or representing detailed attributes.

In its current form, CoLLIE learns a transformation on top of the language embedding. It should in principle be possible to instead apply this transformation to the image embedding, to better match the language embedding. This choice is dependent on the use case: If it is applied to the language embedding, both the newly learned and already existing referring expressions will be correctly resolved (as we saw in Experiment I), but this would not be the case if the image embedding was transformed to match the new language use. Recent work on tuning models for image and text embeddings has also shown that it is better to freeze the image model, while fine-tuning the text model (Zhai et al., 2022).

An interesting topic for future work is how to consolidate the learned transformation function into the foundation model, and then learn a new transformation function on top of this, or to use different transformation functions in different contexts, as language use is highly context-dependent. Another line of future work is to incorporate the model into a system that learns through interaction. Given the small number of examples needed to learn new language use, the model should be interesting for studies on continual language grounding in the context of human-robot interaction (e.g., Chai et al., 2016), where the robot should be able to both comprehend and generate referring expressions based on partner-specific language use.

7. Conclusion

We have presented CoLLIE: a simple, yet effective model for continual learning of how language is grounded in vision. Given a pre-trained language-image embedding model capable of zero-shot image classification or referring expression comprehension, such as CLIP, CoLLIE learns a transformation function that adjusts the language embeddings when needed to accommodate new language use. The transformation function learns the difference vector that needs to be applied to the embedding, and uses a scaling function to retain embeddings that should not be adjusted. We establish new benchmarks to capture the trade-off between continual learning, retention (avoiding catastrophic forgetting), and generalization. While our evaluation is limited in several regards, it indicates that the model can learn new language use with very few examples. Unlike traditional few-shot learning, the model does not just learn new labels, but can also generalize to similar language use, and benefit from the semantic compositionality of language.

Reproducability Statement

The models were implemented using scikit-learn (`https://scikit-learn.org/`) with standard parameters unless stated otherwise. The code used for running the experiments and reproducing the results in this paper is provided on GitHub (`https://github.com/gabriel-skantze/CoLLIE`), including necessary data or pointers to data.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors would like to thank Johan Boye, Ulme Wennberg, Dmytro Kalpakchi, and the anonymous reviewers for their helpful comments.

References

Barr, D. J., & Keysar, B. (2002). Anchoring Comprehension in Linguistic Precedents. *Journal of Memory and Language*, *46*(2), 391–418.

Benavides-Prado, D., Koh, Y. S., & Riddle, P. (2020). Towards Knowledgeable Supervised Lifelong Learning Systems. *Journal of Artificial Intelligence Research*, *68*, 159–224.

Bommasani, R., et al. (2021). On the Opportunities and Risks of Foundation Models. *arXiv, 2108.07258*.

Brennan, S., & Clark, H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology*, *22*(6), 1482–1493.

Brennan, S. E., & Hanna, J. E. (2009). Partner-specific adaptation in dialog. *Topics in Cognitive Science*, *1*(2), 274–291.

Bruni, E., Tran, N. K., & Baroni, M. (2014). Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, *49*, 1–47.

Chai, J. Y., Fang, R., Liu, C., & She, L. (2016). Collaborative language grounding toward situated human-robot dialogue. *AI Magazine*, *37*(4), 32–45.

Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M., Parikh, D., & Batra, D. (2017). Visual dialog. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 1080–1089.

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, p. 933–941.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., & Mikolov, T. (2013). DeViSE: A Deep Visual-Semantic Embedding Model. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 2121–2129.

Grice, H. P. (1975). Logic and Conversation. *Syntax and Semantics*, *3*, 41–58.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, *42*(1-3), 335–346.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Horst, J. S., & Hout, M. C. (2016). The Novel Object and Unusual Name (NOUN) Database: A collection of novel images for use in experimental research. *Behavior Research Methods*, *48*(4), 1393–1409.

Ibarra, A., & Tanenhaus, M. K. (2016). The Flexibility of Conceptual Pacts: Referring Expressions Dynamically Shift to Accommodate New Conceptualizations. *Frontiers in Psychology*, *7*.

Kafle, K., & Kanan, C. (2017). Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, *163*, 3–20.

Kemker, R., & Kanan, C. (2018). FearNet: Brain-Inspired Model for Incremental Learning. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*.

Kemker, R., McClure, M., Abitino, A., Hayes, T., & Kanan, C. (2018). Measuring Catastrophic Forgetting in Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

Krahmer, E., & van Deemter, K. (2012). Computational Generation of Referring Expressions: A Survey. *Computational Linguistics*, *38*(1), 173–218.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation*, *24*, 109–165.

Panagiaris, N., Hart, E., & Gkatzia, D. (2021). Generating unambiguous and diverse referring expressions. *Computer Speech and Language*, *68*, 101184.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, *113*, 54–71.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pickering, M., & Garrod, S. (2006). Alignment as the Basis for Successful Communication. *Research on Language and Computation*, *4*, 203–228.

Qiao, Y., Deng, C., & Wu, Q. (2021). Referring Expression Comprehension: A Survey of Methods and Datasets. *IEEE Transactions on Multimedia*, *23*, 4426–4440.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. *arXiv*, *2103.00020*.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010.

Shin, H., Lee, J. K., Kim, J., & Kim, J. (2017). Continual Learning with Deep Generative Replay. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2994–3003.

Shore, T., Androulakaki, T., & Skantze, G. (2018). KTH Tangrams: A Dataset for Research on Alignment and Conceptual Pacts in Task-Oriented Dialogue. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pp. 768–775.

Shore, T., & Skantze, G. (2018). Using Lexical Alignment and Referring Ability to Address Data Sparsity in Situated Dialog Reference Resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2288–2297.

Takmaz, E., Giulianelli, M., Pezzelle, S., Sinclair, A., & Fernández, R. (2020). Refer, Reuse, Reduce: Generating Subsequent References in Visual and Conversational Contexts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4350–4368.

Van Der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605.

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Computing Surveys*, *53*(3), 1–34.

You, Q., Jin, H., Wang, Z., Fang, C., & Luo, J. (2016). Image Captioning With Semantic Attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4651–4659.

Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., & Beyer, L. (2022). LiT: Zero-Shot Transfer with Locked-image text Tuning. *arXiv*, *2111.07991*.

Zhao, B., Fu, Y., Liang, R., Wu, J., Wang, Y., & Wang, Y. (2018). A Large-scale Attribute Dataset for Zero-shot Learning. *arXiv*, *1804.04314*.