Practical Cross-modal Manifold Alignment for Robotic Grounded Language Learning

Andre T. Nguyen, Luke E. Richards
Booz Allen Hamilton
University of Maryland, Baltimore County
{Nguyen_Andre,Richards_Luke}@bah.com

Gaoussou Youssouf Kebe
University of Maryland, Baltimore County
mb88814@umbc.edu

Edward Raff
Booz Allen Hamilton
University of Maryland, Baltimore County
Raff_Edward@bah.com

Kasra Darvish, Frank Ferraro, Cynthia Matuszek
University of Maryland, Baltimore County
{kasradarvish,ferraro,cmat}@umbc.edu

Abstract

We propose a cross-modality manifold alignment procedure that leverages triplet loss to jointly learn consistent, multi-modal embeddings of language-based concepts of real-world items. Our approach learns these embeddings by sampling triples of anchor, positive, and negative data points from RGB-depth images and their natural language descriptions. We show that our approach can benefit from, but does not require, post-processing steps such as Procrustes analysis, in contrast to some of our baselines which require it for reasonable performance. We demonstrate the effectiveness of our approach on two datasets commonly used to develop robotic-based grounded language learning systems, where our approach outperforms four baselines, including a state-of-the-art approach, across five evaluation metrics.

1. Grounded Language Acquisition Through the Lens of Manifold Alignment

As robots become advanced and affordable enough to have in daily life, more needs to be done to make these machines as intuitive as possible. Language offers an approachable interface. However, understanding how natural language can best be grounded to the physical world is still very much an open problem. Combining language and robotics creates unique challenges that much of the current work on grounded language learning has not yet addressed.

Acquiring grounded language—learning associations between symbols in language and their referents in the physical world—takes many forms [14]. With some exceptions [37, 39], the majority [21, 34] of current work focuses on grounding language to RGB images. Due to the availability of large datasets consisting of up to millions of parallel RGB images and language [21, 25, 31], these tasks typically operate with a large pool of labeled data. Large annotated datasets are rare in the field of grounded language for robotics, especially datasets containing depth information in the form of RGB-D.

This is a complex problem space, and learning has been demonstrated successfully in domains as varied as soliciting human assistance with tasks [20], interactive learning [36], and understanding complex spatial expressions [28]. Previous work has made many simplifying assumptions such as using a bag-of-words language model [29] and focusing on using domain-specific visual features for training classifier models [33]. Our approach relaxes these assumptions: we assume neither a particular language model nor specific visual features.

We approach the grounding problem as a manifold alignment problem where we want to find a mapping from heterogeneous representations to a shared manifold in a latent space. In particular, we demonstrate how to recast existing but disparate language and vision domain representations into a joint space. We do so by learning a transform of both language and RGB-D sensor data embeddings into a joint space using manifold alignment. This enables the learning of grounded language in a cross-domain manner and provides a bridge between the noisy, multi-domain perceived world of the robotic agent and unconstrained natural language. In particular, we use triplet loss in combination with Procrustes analysis to achieve the alignment of language and vision.

Our approach to alignment attains state-of-the-art performance on the language enhanced University of Washington

RGB-D Object Dataset [22, 33] and on the dataset of Pillai and Matuszek [29]. Importantly, as our approach leverages existing feature extractors, it should be able to integrate with existing robot language and vision models with little additional overhead.

We make four main contributions. First, we introduce an easy to implement manifold alignment approach to the grounded language problem for systems where sensor data representations do not live in the same space. Second, we demonstrate that our method is generalizable to the unsupervised setting. Third, we show that our approach can benefit from, but does not require, post-processing steps such as Procrustes analysis—in contrast to some of our baselines which do not perform well without it. Finally, we demonstrate that our method can be effective in lower-resource and lower-data settings compared to traditional uses of manifold alignment in grounded language learning.

2. Related Work

We treat the language grounding problem as one of manifold alignment—finding a mapping from heterogeneous representations (commonly the case with language and sensor datasets) to a shared structure in latent space [40]. This makes the assumption that there is an underlying, latent manifold that datasets share, obtained by leveraging correspondences between paired data elements. Jointly learning embeddings for different data domains to a shared latent space can yield a consistent representation of concepts across domains.

Figure 1 illustrates the goal of aligning language and vision. Given n different domains, the manifold alignment task is to find n functions, $f_1, ..., f_n$ such that each function maps each m_i-dimensional space to a shared latent M-dimensional space, $f_i \colon \mathbb{R}^{m_i} \rightarrow \mathbb{R}^M \ , i = 1, ..., n$. In our case, $n = 2$ where the domains correspond to RGB-D and natural language.

Applying manifold alignment to learning groundings between language and physical context is a relatively novel approach. Most prior work in this area focus on the cooking domain using the much larger Recipe1M dataset containing around one million cooking recipes and eight hundred thousand food images [5, 12, 34]. Our work differs from these previous works as we demonstrate the effectiveness of a manifold alignment approach using a much smaller amount of labeled data (our datasets have less than one percent of the number of data points in the Recipe1M dataset). Lazaridou et al. [23] learn a projection of image-extracted features to an existing and fixed language embedding space.

In the robotics domain, Cohen et al. [7] combine Bayesian Eigenobjects with a language grounding model that maps natural language phrases and segmented depth images to a shared space. This Bayesian Eigenobjects approach is however evaluated on only three classes of objects. Moreover, Choi et al. [6] employ nonparametric regression and deep latent variable modeling to transfer human motion data to humanoid robots. Lu et al. [24] introduce ViLBERT, task-agnostic and transferable joint representations of image content and natural language. Su et al. [38] similarly introduce VL-BERT. Our work differs from ViLBERT and VL-BERT as we are not tackling the problem of learning joint embeddings but rather the problem of recasting different existing embeddings into a joint space. Also, the computational requirements for our work are lower than those needed for training ViLBERT and VL-BERT.

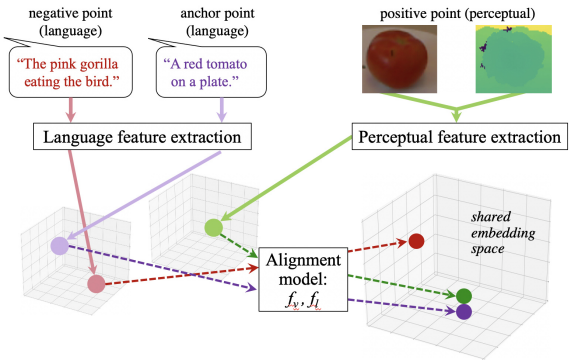


Figure 1: A language + vision manifold alignment approach to language grounding. As an example here, natural language (top left) and perceptual data (top right) are vectorized by a feature extractor and embedded (bottom left). Triplet loss-based alignment functions (f_v and f_l) are then applied to learn a mapping in which similar concepts in different domains are "close" in the new shared embedding, while dissimilar concepts are distant (bottom right).

3. Heterogeneous Domain Alignment

Deep metric learning [18] uses deep neural networks to learn a projection of data to an embedding space where intra-class distances are smaller than inter-class distances. Our intention is that the learned metric and embedding capture the semantics of the paired data. The triplet loss directly encodes the desire that data from a common class be 'closer together' than data from other classes [3, 35]. In particular, triplet loss seeks to minimize the distance between an anchor point and a positive point belonging to the same class as the anchor, while maximizing the distance between the anchor point and a negative point belonging to a different class. Given an anchor x_a, positive x_p, and negative x_n triplet each in $\mathbb{R}^m$, we seek to minimize the following triplet loss where d is a distance metric, f is the embedding function we want to learn, and α is a margin enforced between positive and negative data pairs:

$$L = \max\left(d\left(f(x_a), f(x_p)\right) - d\left(f(x_a), f(x_n)\right) + \alpha, 0\right) \tag{1}$$

Previous work has used triplet loss for learning metric embeddings, for example Hermans et al. [16] maps similar data from homogeneous domains closer to each other in a shared lower-dimensional latent space. Our approach, in contrast, is to use data from heterogeneous domains to learn the metric embeddings based on triplet loss.

More specifically, we wish to learn two embedding alignment functions f_v and f_l that map RGB-D (i.e., "vision" f_v) and language (f_l) data respectively to a shared representation space. We do not fine-tune the original embeddings: our empirical results demonstrate that certain types of grounded language learning can be accomplished without it. In order to jointly learn f_v and f_l, we use the triplet loss but select triplets to be cross-domain. In particular, we uniformly at random select triplets such that the anchor, the positive, and the negative can independently belong to either domain. For example, in the case where the anchor and negative come from the vision domain and the positive comes from the language domain, the loss for that triplet is:

$$L = \max\left(d(f_v(x_a), f_l(x_p)) - d(f_v(x_a), f_v(x_n)) + \alpha, 0\right) \tag{2}$$

In the above example, x_a could be a cat RGB-D image, x_p a textual description of a cat, and x_n a toaster image.

Once the embedding alignment transformations f_v and f_l are learned, an optional fine-tuning step can be included in the form of a Procrustes analysis [13] which finds the optimal translation, scaling, and rotation of two shapes to minimize the Procrustes distance between the shapes. The Procrustes distance is the Euclidean distance between the shapes after the learned optimal translation, scaling, and rotation of shapes. An optimal rotation matrix R is found such that the Euclidean distance between the shapes after translation and scaling is minimized

$$R^* = \arg\min \left|\left| \frac{f_v(X_v) - m_v}{\|f_v(X_v) - m_v\|_F} - \frac{f_l(X_l) - m_l}{\|f_l(X_l) - m_l\|_F} R^T \right|\right|_2 \tag{3}$$

where X_v and X_l are the vision and language data respectively (where rows from each domain form pairs), where m_v and m_l are the means of $f_v(X_v)$ and $f_l(X_l)$, and $\|\cdot\|_F$ is the Frobenius matrix norm. All the Procrustes analysis parameters are fit using the training set. As we will show, our method can benefit from, but does not require, Procrustes analysis, in contrast to some of our baselines which require it for reasonable performance. Our primary method, called "Triplet Method" throughout this paper, uses cosine distance as the distance metric d and includes Procrustes analysis. The full training procedure for the triplet method is given in algorithm 1.

Algorithm 1: Training Procedure for Triplet Method

Input: Dataset X of paired RGB-D and language feature vectors (x_v, x_l).

Output: Embedding alignment functions f_v and f_l that map RGB-D and language to a shared space and a trained Procrustes transform.

$f_v, f_l \leftarrow$ randomly initialized neural networks with parameters θ_v and θ_l respectively

while *not converged* **do**

  $x_a \leftarrow$ randomly selected vision or language feature vector from X

  $x_p \leftarrow$ randomly selected vision or language feature vector from X belonging to the same class as x_a

  $x_n \leftarrow$ randomly select any other vision or language feature vector from X belonging to a different class than x_a and x_p

  Incur loss L using Equation 2, and backpropogate to update parameters θ_v and θ_l

end

$m_v \leftarrow \frac{1}{|X|} \sum_{\forall (x_v, x_l) \in X} f_v(x_v)$

$m_l \leftarrow \frac{1}{|X|} \sum_{\forall (x_v, x_l) \in X} f_l(x_l)$

$s_v \leftarrow \|f_v(X_v) - m_v\|_F$

$s_l \leftarrow \|f_l(X_l) - m_l\|_F$

$R \leftarrow$ solution to Equation 3

return $f_v, f_l, m_v, m_l, s_v, s_l, R$

4. Experiments

4.1. Grounded Language Data and Evaluation

We use the same RGB-D object dataset used by Richards and Matuszek [33], which extends the classic and well-known University of Washington object dataset [22] with natural language text descriptions. The dataset consists of 7,455 RGB-D image and text description pairs where the pairs each belong to one of 51 classes and where the number of data points per class range from 33 to 366. We split the data so that objects tested on do not appear in the training set. So for example, all data derived from a specific "water bottle" will appear in only one of the training and testing sets. In other words, data in the testing set does not come from the same objects as data in the training set.

Figure 2 shows example data from the *tomato*, *pear*, and *food bag* classes. The three examples shown in Figure 2 illustrate how ambiguity can occur in natural language, as all three classes can be described using the word "fruit." During evaluation, we desire that a good approach map the

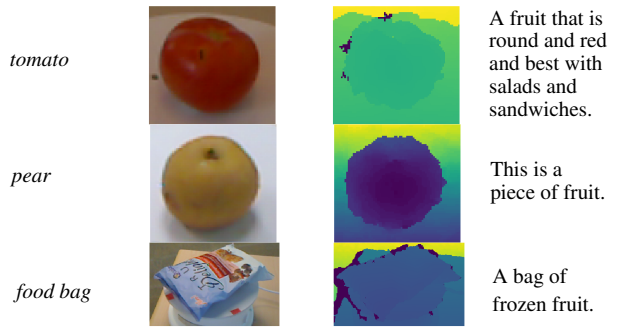


Figure 2: Example data. Columns correspond to class label, RGB, depth, and text descriptions.

RGB-D and language representations of each object class near each other but far from the representations of objects from other classes.

4.2. Models

For the language feature extraction model b_l, we use a 12-layer BERT model pre-trained on lowercase English text [9]. We used the concatenated output of the last four BERT-base layers, resulting in a 3,072 dimensional embedding.

For the vision feature extraction b_v, we use a ResNet152 pre-trained on ImageNet [15] with its last fully connected layer removed. The depth component is dealt with via colorization (which we shall call D2RGB) in a similar manner to the procedure from Eitel et al. [11] which encodes a depth image as an RGB image where the information contained in the depth data is spread across all three RGB channels. This allows us to use the same pre-trained ResNet to process both the RGB image and the transformed depth information. The vectors resulting from the RGB images and the D2RGB depth-to-RGB colorization are concatenated to create a final 4,096 dimensional RGB-D vision embedding. This gives us $b_v(x_{RGB-D}) = [\text{ResNet}(x_{RGB}); \text{ResNet}(\text{D2RGB}(x_D))]$.

Lu et al. [24] introduce ViLBERT, joint representations of images and natural language. We note that while a pre-trained ViLBERT embedding could be used for the vision and language feature extraction, we do not use ViLBERT as our feature extractor in our experiments. This is because ViLBERT learns vision and language embeddings jointly, and so the representations are already designed to work together. Our interest is in adapting embeddings that have no prior relation.

In our experiments, the network architectures for our alignment models consist of an input layer, two hidden layers of size equal to the input layer size, and an output layer that has the size of the desired embedding dimensionality, set to 1,024 in our experiments. Rectified linear units were used as hidden layer activation functions, and Adam was used as the optimizer [19]. The triplet loss uses cosine distance as the distance metric with a margin of $\alpha = 0.4$, where we did not tune the margin. PyTorch 1.4.0 was used on a Ubuntu 18.04 server with a GeForce RTX 2080 Ti GPU. The embedding space is chosen to be 1,024-dimensional and we fix the pre-trained feature extraction models b_l and b_v during training, only optimizing the alignment models.

The fixing of the feature extraction models directly connects to the robotics use-case where feature extraction model outputs may be used for multiple tasks and where there may be memory and latency constraints. By not having to store and process data through multiple feature extraction models, our approach is advantageous in how it can fit on top of existing state-of-the-art algorithms used by the robot for separate tasks. To illustrate, the feature extraction models together have $167{,}626{,}048$ parameters, and the alignment models together have $59{,}785{,}216$ parameters. In the case of an existing system with language and vision models currently being used for other tasks, the integration of manifold alignment would result in a 36% increase in the number of parameters if the feature extraction models are reused whereas an 136% increase in the number of parameters would occur if the feature extraction models are retrained.

4.3. Baselines

We compare our manifold alignment method with the following baselines. We also augment each of these baselines with a Procrustes analysis for additional, stronger baselines.

4.3.1 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) finds the linear combinations of variables within each of two datasets that maximizes the linear correlation between the combinations from each of the datasets [17].

4.3.2 Deep CCA

Deep Canonical Correlation Analysis (Deep CCA) is an extension of CCA where a nonlinear transformation of two datasets is learned to maximize the post-transformation linear correlation [2]. Deep CCA suffers from known numerical stability issues due to the need to backpropagate through eigen-decompositions. Additionally, mini-batched stochastic gradient descent cannot be directly used for optimization as correlation is a function of the training data in its entirety and does not decompose into a sum over data points. As a result, care needs to be taken when training Deep CCA [41, 42]. In particular, we found it necessary to select a smaller embedding space dimensionality of 64 (instead of 1,024) for Deep CCA in order to avoid numerical instability

during the training process. Testing with larger dimensions resulted in a failure to converge.

4.4. Manifold Metrics

To evaluate the quality of the manifolds learned, we will use the three metrics specified below: Mean Reciprocal Rank (a measure of global order preservation), K-Nearest Neighbors (a measure of local order preservation), and Distance Correlation (a measure of global absolute distance preservation). A successful manifold alignment approach should perform well in all three of these tasks. We do not argue that these are sufficient for determining all aspects about a manifold's quality, but posit that they are useful to the tasks we are concerned with. Similar metrics were used for example in Diaz and Metzler [10] and Aalto and Verma [1].

4.4.1 Mean Reciprocal Rank

Given an image and text pair, we can compute the distance in the joint embedding space between the text element and all data points in the vision domain. These distances can then be ranked with 1 being the closest, 2 being the second closest, and so forth. Common in information retrieval, Mean Reciprocal Rank (MRR) is the average across the data of the multiplicative inverse of the rank in embedding space of the nearest item from the same class that comes from the other domain [8].

4.4.2 Distance Correlation

Intuitively, if two embedding manifolds are aligned, distances in one embedding should be correlated to distances in the other embedding. Specifically, if we select two image and text pairs, the distance between them in the vision embedding should be correlated with the distance between them in the language embedding space. To capture this property, we randomly select $10{,}000$ pairs of image and text pairs and compute the distance between them. The Pearson correlation is then computed between the vision space distances and the language space distances, resulting in a metric between -1 and 1 where closer to 1 means better alignment. We call this metric Distance Correlation (DC) in this paper. The sampling is done due to the prohibitive cost to compute the pairwise correlation for the entire dataset.

4.4.3 K-Nearest Neighbors

As a final metric, we use K-Nearest Neighbors (KNN) classification accuracy with $K = 5$ in our experiments. This metric captures what performance would look like in an applied setting where a robot may need to associate natural language with a visual concept.

5. Supervised Alignment Evaluation

5.1. Grounded Language Learning

Our ultimate goal for manifold alignment is to enable the grounding of language to referents in the physical world. To directly assess the effectiveness of cross-modal manifold alignment for grounded language, we evaluate the aligned embeddings on the task of determining which objects in RGB-D space correspond to a given language description. In particular, every text description datum can be considered a separate classification task where the goal is the binary classification of all RGB-D images as relevant or not relevant given the text description.

For each of the classification tasks, an Area Under the Receiver Operating Characteristic Curve (AUC) score is obtained. Figure 3 shows cumulative counts over AUCs. We note that for any particular AUC score, our triplet method has more better scoring tasks than Deep CCA. In other words, Deep CCA has more and worse failure cases. We also compare our triplet method which uses cosine distance with a version of our triplet method that uses Euclidean distance instead. This ablation finds our cosine method best.

Table 1 summarizes the mean micro and macro averaged F1 scores across methods. The triplet method outperforms all of the other methods on the grounded language task. For the computation of F1 scores, the distance between the text description element and RGB-D image element in the shared space was computed for each datum pair in the training set. The relevance distance threshold was set to the mean of these distances plus a standard deviation.

A comparison of the achieved mean macro averaged F1 score of 0.725 for the triplet method in the known class scenario with the 0.689 macro averaged F1 score reported in Richards and Matuszek [33], the current state-of-the art on this dataset, shows a 5.2% improvement and suggests that a manifold alignment approach to grounded language is promising, attaining at least similar or likely better performance than traditional word-as-classifier models. The triplet method without Procrustes achieves a higher 9.9% improvement in macro averaged F1 score, but we will later discuss our preference for the triplet method with Procrustes.

5.2. Effective Deep Metric Learning using Triplet Loss

Table 2 shows the MRR, KNN accuracy, and DC for the triplet method as well as for our baselines. We find that while the triplet method has the highest DC and strong MRR and KNN accuracy, providing consistent performance across all manifold metrics, Deep CCA with the addition of Procrustes analysis has the highest MRR and KNN, at the cost of a $1.9\times$ lower DC compared to our new approach. This disparity in performance means that Deep CCA with

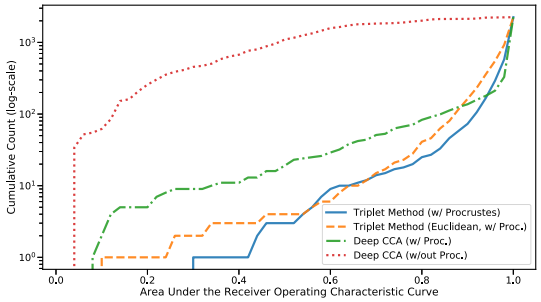


Figure 3: Grounded language task cumulative counts over AUCs. Perfect classification lies in the bottom-right corner, so curves toward the lower-right are preferred. Every text description datum can be considered a separate classification task (with its own AUC) where the goal is the binary classification of all RGB-D images as relevant or not relevant given the text description. The x-axis represents AUC score values, and the y-axis represents the number of classification tasks with an AUC less than a particular value. For any particular AUC score, our triplet method has more better scoring tasks than Deep CCA.

Algorithm	Avg Micro F1	Avg Macro F1
Triplet Method	**0.983**	0.725
Trip. Met. (w/out Procrustes)	0.978	**0.757**
Trip. Met. (Euclidean)	0.969	0.727
Trip. Met. (Eucl. w/out Proc.)	0.952	0.714
Cosine Baseline (w/ Proc.)	0.441	0.318
Cosine Baseline (w/out Proc.)	0.542	0.337
CCA (w/ Procrustes)	0.567	0.294
CCA (w/out Procrustes)	0.455	0.331
Deep CCA (w/ Procrustes)	0.855	0.716
Deep CCA (w/out Procrustes)	0.026	0.025
Richards and Matuszek (2019)	Not Reported	0.689

Table 1: Metrics for grounded language task evaluated on held out test set. Best results are **bolded**.

Procrustes is not learning a holistically useful manifold. As we saw in subsection 5.1, this translates to worse performance for grounded language learning. Deep CCA without Procrustes has a significantly reduced, and in fact the worst, MRR and KNN accuracy. CCA with and without Procrustes analysis both have poor performance. These results demonstrate the value of using Procrustes to improve the quality of a manifold alignment at little effort. We also note that while Procrustes is crucial for CCA and Deep CCA, our triplet method remains strong with only a slight decrease in MRR and KNN accuracy when Procrustes analysis is ablated.

To help confirm that our approach learns good manifolds, we would expect a visualization of the vision and language domains to have similar structure. We do this using UMAP [26], which preserves global structure. Figure 4 shows the UMAP for the triplet method. Ten randomly selected classes are plotted for legibility purposes. We observe that classes are generally well clustered (items are close to other items from the same class and classes are separated) and are projected to similar locations across both the language and vision domains. Note that using our new approach, classes with wide dispersion (e.g., *water bottle*) or compactness (e.g., *cell phone*) share this structure across domains. Figure 5 shows the UMAP for Deep CCA with Procrustes. In contrast with the triplet method, we observe that while data is well clustered in the language domain, data is less well clustered in the vision domain, in particular when it comes to class separation. Class alignment across domains is also less evident. While classes such as *cell phone* and *food bag* are well aligned, other classes such as *kleenex* and *calculator* are not. In these cases the structure is not successfully shared between the domains, indicating a lesser quality as a manifold.

Additionally, we can gain more insight into the DC results by plotting the vision space distances and the language space distances to compare their relationships. Subplots (a) and (b) in Figure 6 respectively show the distance relationships for the triplet method and Deep CCA with Procrustes.

Algorithm	MRR	KNN	DC
Triplet Method	0.802	0.787	0.686
Trip. Met. (w/out Procrustes)	0.758	0.742	0.692
Trip. Met. (Euclidean)	0.724	0.702	**0.693**
Trip. Met. (Eucl. w/out Proc.)	0.673	0.648	0.685
Cosine Baseline (w/ Proc.)	0.113	0.097	-0.001
Cosine Baseline (w/out Proc.)	0.208	0.181	0.031
CCA (w/ Procrustes)	0.144	0.122	0.067
CCA (w/out Procrustes)	0.035	0.027	0.040
Deep CCA (w/ Procrustes)	**0.870**	**0.860**	0.359
Deep CCA (w/out Procrustes)	0.023	0.012	0.377

Table 2: Evaluation of manifolds using Mean Reciprocal Rank (MRR), K-Nearest Neighbors (KNN), and Distance Correlation (DC) as metrics. Higher is better for all metrics.

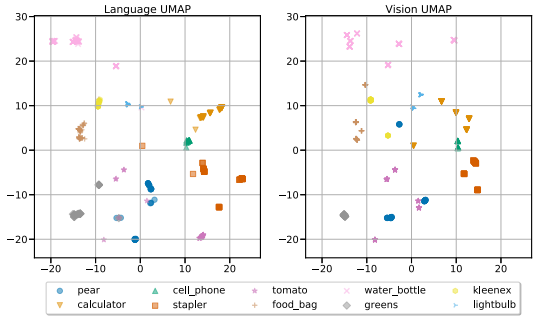


Figure 4: Test set UMAP of the Triplet Method. 10 randomly selected classes are plotted.

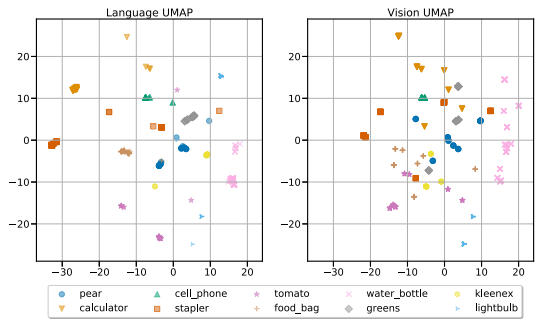
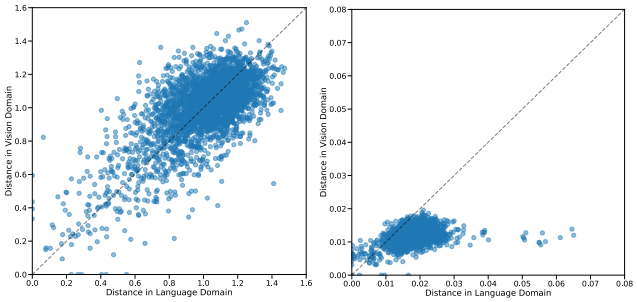


Figure 5: Test set UMAP of Deep CCA with Procrustes. 10 randomly selected classes are plotted.



(a) Triplet Method (b) Deep CCA with Procrustes

Figure 6: Distance Correlation visualization for the Triplet Method and for Deep CCA with Procrustes. Pairs of image and text pairs are randomly selected and the distance between them is plotted, with the x-axis representing the distance in the language domain and the y-axis representing the distance in the vision domain. The dashed line represents where points should lie under perfect manifold alignment.

While the triplet method has the desired linear relationship between distances, Deep CCA with Procrustes lacks the desired relationship that would indicate well aligned manifolds.

5.3. Understanding the Contribution of Procrustes Analysis and Triplets

To understand the role played by Procrustes analysis, we run ablation experiments, separately removing each of the Procrustes analysis components (translation, scaling, and rotation) one by one. Table 3 shows metrics for Procrustes analysis ablations on the triplet method. Metrics stay relatively similar when translation or scaling are removed. When rotation is removed, a decrease in MRR and KNN accuracy is observed without a decrease in DC. Similar ablation experiments can be run for Deep CCA with Procrustes analysis. Table 4 suggests that both rotation and scaling are needed for Deep CCA to achieve high MRR and KNN accuracy.

Algorithm	MRR	KNN	DC
Triplet Method	0.802	0.787	0.686
No Translation	**0.806**	**0.790**	0.679
No Scaling	0.801	0.786	**0.696**
No Rotation	0.750	0.733	0.693

Table 3: Ablation metrics where various components of Procrustes analysis are disabled for the Triplet Method.

Algorithm	MRR	KNN	DC
Deep CCA w/ Procrustes	0.870	0.860	0.359
No Translation	**0.871**	**0.862**	0.363
No Scaling	0.021	0.011	**0.378**
No Rotation	0.034	0.021	0.352

Table 4: Ablation metrics where various components of Procrustes analysis are disabled for Deep CCA.

Algorithm	MRR	KNN	DC
Triplet Method	0.724	0.702	**0.693**
No Translation	**0.729**	**0.707**	0.688
No Scaling	0.045	0.039	0.680
No Rotation	0.680	0.658	0.684

Table 5: Ablation metrics where various components of Procrustes analysis are disabled for the Triplet Method with Euclidean distance.

Table 5 shows metrics for Procrustes analysis ablations on a variant of the triplet method that uses Euclidean distance instead of cosine distance. Metrics stay similar when translation or rotation are removed. When scaling is removed, a significant decrease in MRR and KNN accuracy is observed. This suggests that the Euclidean distance without Procrustes maps data in each domain to similarly shaped manifolds of different scales. This result is consistent with the formulation of the Euclidean triplet loss, as differently scaled but otherwise similar manifolds can satisfy the relative distance constraints encouraged by the Euclidean triplet loss. This result demonstrates an advantage of the use of cosine distance in this context. A comparison of the performance of the triplet method with its Euclidean variant in Table 1, Table 2, and Figure 3 confirms this advantage.

We also explore the contribution of using triplets by adding a baseline which seeks to simply minimize the cosine distance between the positive and anchor points in the shared space. Table 1 and Table 2 show the performance of the cosine distance baseline, with and without Procrustes analysis. Overall, the triplet method performs significantly better than the cosine distance baselines. We note that our

Algorithm	Micro F1	Macro F1	MRR	KNN	DC
Triplet Met. (BERT)	**0.984**	0.735	0.816	0.804	0.686
Triplet Met. (SBERT)	0.982	**0.748**	0.745	0.731	0.678
Triplet Met. (SBERT fine-tuned)	**0.984**	0.734	**0.834**	**0.823**	**0.731**

Table 6: Metrics for grounded language task and manifold evaluation comparing BERT, SBERT, and a fine-tuned SBERT. We report average F1 scores.

Algorithm	Micro F1	Macro F1	MRR	KNN	DC
Triplet Method	**0.983**	**0.725**	**0.802**	**0.787**	0.686
Trip. Met. (unsup.)	0.963	0.698	0.754	0.736	**0.773**
Trip. Met. (unsup. w/out Proc.)	0.941	0.685	0.688	0.665	0.725

Table 7: Metrics for grounded language task and evaluation of manifolds in the unsupervised setting. We report average F1 scores.

cosine baseline is similar to the approach taken by Nguyen et al. [27].

5.4. Comparison of Language Embeddings

Next, we investigate the effect of better feature extraction. Sentence-BERT (SBERT) is a sentence embedding oriented modification of BERT that achieves better performance on Semantic Textual Similarity (STS) tasks [32]. We compare a BERT-based version of our triplet method to an off-the-shelf SBERT version and a fine-tuned SBERT version. We fine-tune SBERT using pairs of object descriptions from the same extended University of Washington dataset. Pairs describing the same instance of an object are given a score of 5 while pairs describing different instances of an object are given a score of 2.5, and pairs describing different objects are given a score of 0.

Table 6 summarizes the comparative performance of the language embeddings on the grounded language task and manifolds. Fine-tuned SBERT leads to the highest quality manifold. This follows intuition and suggests that the use of higher quality original embeddings of sensor data leads to higher quality aligned representations. Note that we retrained the BERT based triplet method for this experiment, hence the slightly different (but nearly identical) metrics when compared to Table 1 and Table 2.

6. Generalizability to Other Settings

We now investigate if our approach generalizes to situations where unsupervised manifold alignment is needed, and to another dataset with more limited labeled data.

6.1. Sampling Negative Examples in an Unsupervised Setting

So far, the training of the triplet method has assumed the availability of class labels for triplet selection. However, the triplet method can still be trained when class ground truth is not available using unsupervised negative example selection. In this setting, the triplets are fixed to have a vision anchor and language negatives and positives. The positive is selected to be the anchor's paired text, and the negative example is chosen through a semantic distance based technique similar to that used in [29]. In particular, the cosine distances between all natural language descriptions can be computed, and the negative is sampled from the 25% of descriptions furthest away from the positive description. This can be interpreted as aligning vision to the manifold induced by the language embedding. Table 7 summarizes the performance of the triplet method in this unsupervised setting. While there is a decrease in MRR and KNN accuracy, DC remains strong and even increases. On the grounded language task, performance also remains strong with only a 2% decrease in average micro F1 and a 4% decrease in average macro F1, compared to the triplet method results.

6.2. Effectiveness on a Smaller Dataset

We also test our triplet method on a dataset [29] containing fewer classes and fewer instances per class, with a lower computational cost vision extraction method, depth kernel descriptors [4] and average values for RGB channel values. Prior work [29, 30] used these same visual feature extraction methods with a word-as-classifier model. Pillai et al. [30] combined depth kernel descriptors and averaged RGB channel values. We concatenate the kernel descriptors and the average RGB channel values into a single vision embedding vector. Each vision vector is paired with a natural language description of the object. On this dataset, the triplet method with Procrustes achieves a mean macro F1 score of 0.722, and the triplet method without Procrustes achieves a mean macro F1 score of 0.729, both of which are better than but still comparable to the reported 0.714 for the non-category based model from previous works.

7. Conclusions

We explored the use of the triplet loss enhanced with Procrustes analysis for manifold alignment in the context of grounded language. Our approach to alignment achieves state-of-the-art performance on two datasets, and integration with existing robot sensors and models would likely have minimal additional overhead. Next steps include the alignment of more than two modalities, integration with a robot system, and evaluation on a wider variety of tasks.

References

[1] Max Aalto and Nakul Verma. Metric learning on manifolds. *arXiv preprint arXiv:1902.01738*, 2019. 5

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen

Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255, 2013. 4

[3] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3, 2016. 2

[4] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox. Object recognition with hierarchical kernel descriptors. In *Computer Vision and Pattern Recognition*, 2011. 8

[5] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018. 2

[6] Sungjoon Choi, Matthew Pan, and Joohyung Kim. Nonparametric motion retargeting for humanoid robots on shared latent space. In *Proceedings of Robotics: Science and Systems (R:SS) 2020*. Robotics: Science and Systems (RSS), 2020. 2

[7] Vanya Cohen, Benjamin Burchfiel, Thao Nguyen, Nakul Gopalan, Stefanie Tellex, and George Konidaris. Grounding Language Attributes to Objects using Bayesian Eigenobjects. 2019. 2

[8] Nick Craswell. Mean reciprocal rank. *Encyclopedia of database systems*, 1703, 2009. 5

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. pages 4171–4186, 2018. 4

[10] Fernando Diaz and Donald Metzler. Pseudo-aligned multilingual corpora. In *IJCAI*, pages 2727–2732, 2007. 5

[11] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin A Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust {RGB-D} object recognition. *International Conference on Intelligent Robots and Systems (IROS)*, pages 681–687, 2015. 4

[12] Mikhail Fain, Andrey Ponikar, Ryan Fox, and Danushka Bollegala. Dividing and Conquering Cross-Modal Recipe Retrieval: from Nearest Neighbours Baselines to SoTA. 2019. 2

[13] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 3

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

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[16] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 3

[17] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992. 4

[18] Mahmut Kaya and Hasan Sakir Bilge. Deep metric learning: A survey, 2019. ISSN 20738994. 2

[19] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations*, 2015. 4

[20] Ross A Knepper, Stefanie Tellex, Adrian Li, Nicholas Roy, and Daniela Rus. Recovering from failure by asking for help. *Autonomous Robots*, 39(3):347–362, 10 2015. 1

[21] 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*, 123:32–73, 2016. 1

[22] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view {RGB-D} object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE, 2011. 2, 3

[23] Angeliki Lazaridou, Elia Bruni, and Marco Baroni. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1414, 2014. 2

[24] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. 2019. 2, 4

[25] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1

[26] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2018. 6

[27] Thao Nguyen, Nakul Gopalan, Roma Patel, Matt Corsaro, Ellie Pavlick, and Stefanie Tellex. Robot object retrieval with contextual natural language queries. In *Proceedings of Robotics: Science and Systems (R:SS) 2020*. Robotics: Science and Systems (RSS), 2020. 8

[28] Rohan Paul, Jacob Arkin, Derya Aksaray, Nicholas Roy, and Thomas M Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *The International Journal of Robotics Research*, 37 (10):1269–1299, 2018. 1

[29] Nisha Pillai and Cynthia Matuszek. Unsupervised Selection of Negative Examples for Grounded Language Learning. In *Proceedings of the 32nd National Conference on Artificial Intelligence (AAAI)*, New Orleans, USA, 2018. 1, 2, 8

[30] Nisha Pillai, Cynthia Matuszek, and Francis Ferraro. Deep Learning for Category-Free Grounded Language Acquisition. In *Proc. of the NAACL Combined Workshop on Spatial Language Understanding and Grounded Communication for Robotics (NAACL-SpLU-RoboNLP)*, Minneapolis, MI, USA, 6 2019. 8

[31] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. 1

[32] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410. 8

[33] Luke E Richards and Cynthia Matuszek. Learning to Understand Non-Categorical Physical Language for Human-Robot Interactions. In *Proc. of the R:SS 2019 workshop on AI and Its Alternatives in Assistive and Collaborative Robotics (RSS: AI+ACR)*, Freiburg, Germnany, 6 2019. 1, 2, 3, 5

[34] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3020–3028, 2017. 1, 2

[35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015. 2

[36] Lanbo She and Joyce Yue Chai. Interactive Learning of Grounded Verb Semantics towards Human-Robot Communication. In *ACL*, 2017. 1

[37] Jivko Sinapov, Priyanka Khante, Maxwell Svetlik, and Peter Stone. Learning to order objects using haptic and proprioceptive exploratory behaviors. In *IJCAI*, pages 3462–3468, 2016. 1

[38] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *ICLR*, 2020. 2

[39] Jesse Thomason, Jivko Sinapov, Maxwell Svetlik, Peter Stone, and Raymond J Mooney. Learning Multi-Modal Grounded Linguistic Semantics by Playing" I Spy". In *IJCAI*, pages 3477–3483, 2016. 1

[40] Chang Wang and Sridhar Mahadevan. Manifold alignment preserving global geometry. In *23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013. 2

[41] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Backpropagation-friendly eigendecomposition. In *Advances in Neural Information Processing Systems*, pages 3162–3170, 2019. 4

[42] Weiran Wang, Raman Arora, Karen Livescu, and Nathan Srebro. Stochastic Optimization for Deep CCA via Nonlinear Orthogonal Iterations. 10 2015. 4