Dear Editor and Reviewers,

RE: TPAMI-2014-07-0536, "Transductive Multi-view Zero-Shot Learning" by Yanwei Fu, Timothy M. Hospedales, Tao Xiang and Shaogang Gong

We would like to thank the editor and the reviewers for their constructive comments and suggestions. We have revised our manuscript in line with the comments and suggestions. Details of our changes to the manuscript and our responses (in italics) are elaborated below. We hope that our revised manuscript is acceptable for publication in IEEE TPAMI.

Response to Editor

1. The manuscript has been reviewed by three experts, who all agree that the contribution is relevant and interesting. However, they identified several issues that should be addressed in a revision. Therefore, I recommend that the manuscript undergoes a major revision before being reviewed again. I would encourage the authors to take the reviewer comments as an opportunity for improving their manuscript. In particular, all reviewers mention that the clarity of presentation should be improved. It might be advisable to address this by rewriting at least the critical sections, not just making minor changes. This would also make it clearer that the manuscript has a scientific contribution on its own, and is not just a extended version of the ECCV paper.

   We have revised the presentation extensively for clarity and to ensure all details are better explained. Specifically, 1) we have rewritten some the critical sections highlighted by each reviewer with clearer explanations, e.g. the 'transductive' setting in introduction, related work, as well as the experimental section. 2) A new schematic diagram (Figure 2 in the revised manuscript) was added to illustrate the whole framework of our method. 3) We have made extensive efforts to addressing all comments, in particular the very thorough list of comments raised by Reviewer 3, which are very helpful in improving the clarity and readability of the paper. 4) Numerous validation experiments have been carried out to address the doubts of all aspects from each reviewer. Note that due to the page limit, we are not able to include these new experiments to the main manuscript. They have now been presented in a new supplementary material document.

   More detailed response to each reviewer's comments can be found below. Note that because of the new illustrations and more detailed explanations, we have to move a couple of less important experimental results in the original submission to the supplementary material.

Response to Reviewer 1

1. Unclear contributions w.r.t to preliminary version [15], and why a different method is used. “I think, the paper will improve if it explains more clearly the contributions over [15] in the intro and not hidden in section 2.”

   Thanks for your suggestions. We have now clarified the contributions over the preliminary version [1] into the Introduction section as suggested (see Page 2). Below we elaborate why a different graph-based label propagation method is proposed in this work.

   In the preliminary version [1], for zero-shot classification in the transductive embedding space, a graph-based label propagation (TMV-BLP) method is formulated, which is based on a conventional 2-graph formulation. In this work, we propose a hypergraph based label propagation method termed TMV-HLP. The proposed novel cross-view heterogeneous hypergraph provides additional robustness against noise and better exploits the complementarity of different views.

   More specifically, there are two main differences between TMV-BLP and TMV-HLP: hypergraph is used in place of 2-graph; and heterogeneous graph is used instead of homogeneous graph. The objective of using hypergraph is to make the algorithm more robust against the data noise. A hypergraph is the generalisation of a 2-graph with edges connecting many nodes/vertices, rather than connecting only two nodes in conventional 2-graphs. This enables it to cope better with noisy nodes and thus achieve better performance than conventional graphs as demonstrated extensively by previous works [2, 3, 4], as well as in our experiments (see Fig. 6 in the revised manuscript). On the other hand, the heterogeneous formulation of the hypergraph is a novel contribution of this work and is designed to better combine multiple complementary views of the data. More specifically, most existing works including the preliminary version [1] construct a homogeneous graph in each view and then fuse them. In the proposed formulation, the combination/fusion of different views happens at an earlier stage, namely the hypergraph construction stage. This early fusion approach can be considered as a distributed representation [5, 6] which can potentially lead to better data clustering; importantly, it has been empirically shown to be better than late fusion of multiple homogeneous graphs (see Fig. 6).

2. Experiments lack insight into the proposed paradigm (eg, influence of # projection dimensions, influence of # samples used for CCA).

   (a) Why CCA is selected, not another method?

   We have multiple views and want to embed all these views into a common space so that they can be compared directly (e.g., for multi-view label propagation). Multi-view CCA is a good option due to (i) supporting an arbitrary number of views, and (ii) a computationally efficient closed-formed solution [7] that can be used for large-scale datasets [8].

   Some other methods can also be used to project two spaces into a common one, for example, partial least squares (PLS). Nevertheless, our framework needs to fuse 3 or more views (up to 9 in the largest case), so PLS cannot be directly applied without non-trivial generalisations. There are also other alternatives which are suitable for 3 views such as the work of Wang et al [9] which could potentially further improve our results. However in any case multi-view CCA is efficient and already works well in practice as we demonstrate, so we will leave the exploitation of alternative embedding methods to the future work.

   This is now explained in the Related Work section (Sec. 2).

   (b) What is the influence of projection dimensionality in CCA?

   This is an interesting question and was not discussed clearly in the original submission. In the revised version, we have provided a better explanation (see Sec. 3.3).

   Indeed, selecting the right number for the projection dimensionality in CCA is important. Conventional approaches to CCA [7] need to manually set a number or estimate it via, say cross validation for the projection dimensionality. However, in this work we follow a recent work [8] which avoids the need to explicitly select a dimensionality by using a soft-weighting strategy. Specifically, the dimensionality of our embedding space is the sum of the input dimensions (so no information is lost, unlike in the conventional case of selecting a lower dimensional subspace). We then use the CCA eigenvalues to ***weight*** *the high dimensional embedding space Γ, so that more important dimensions are automatically weighted more highly. Thus conventional CCA can be seen as using a manually fixed hard-weighting (1 for all selected dimensions, 0 for all non-selected dimensions), that is a special case of our soft weighting (each dimension continuously weighted).*

More specifically, from the manuscript (Eq. (1)), we have:

$$\Psi^i = \Phi^i W^i \left[D^i\right]^\lambda = \Phi^i W^i \tilde{D}^i, \tag{1}$$

Here Φ are the input data, which are transformed to the CCA representation Ψ via the CCA projection matrix W and the weighted eigenvalues $\left[D^i\right]^\lambda$. To be precise, if the dimension of each input view is m_i, then the dimension of the CCA space is $\sum_{i=1}^{n_V} m_i$, and the projection matrices are thus of size $W^i \in R^{m_i \times \sum_{i=1}^{n_V} m_i}$. This is a very high dimension and would contain redundant information for sure. However, the sharpening parameter λ (suggested by [8] and also used in [1]) controls how much to prefer high eigenvalue dimensions over low-eigenvalue dimensions. This soft-weighting by eigenvalue strategy simplifies the whole framework by avoiding explicit dimension selection for CCA space. Although the sharpening parameter λ plays an analogous role to conventional CCA subspace dimension selection (larger λ prefers more low-weight dimensions), the whole framework is not very sensitive to this parameter, which was set to 4 throughout our experiments on different datasets.

To validate this point, we conduct the following experiment: We use the hand-crafted features (dimension: 10925) of the AwA dataset with semantic word vector $\mathcal{V}$ (dimension: 1000) and semantic attribute $\mathcal{A}$ (dimension: 85). We compare the CCA embedding space with conventional "hard" selection of number of dimensions against our soft-weighting strategy (applied on the maximum number of 10925+1000+85=12010 dimensions). As shown in Fig. 1 below, it is true that for conventional CCA without eigenvalue weighting, a good selection of subspace dimension (around 50 is the optimal value) gives better results than using all the dimensions. Nevertheless, our soft-weighting strategy applied to the full 12010 dimensional has better results than the best unweighted choice of subspace dimension. Moreover, it is not very sensitive to the weighting parameter, with choices of $\lambda > 2$ all working well. In fact better results can be obtained with $\lambda = 5$ than 4 used in the reported results.

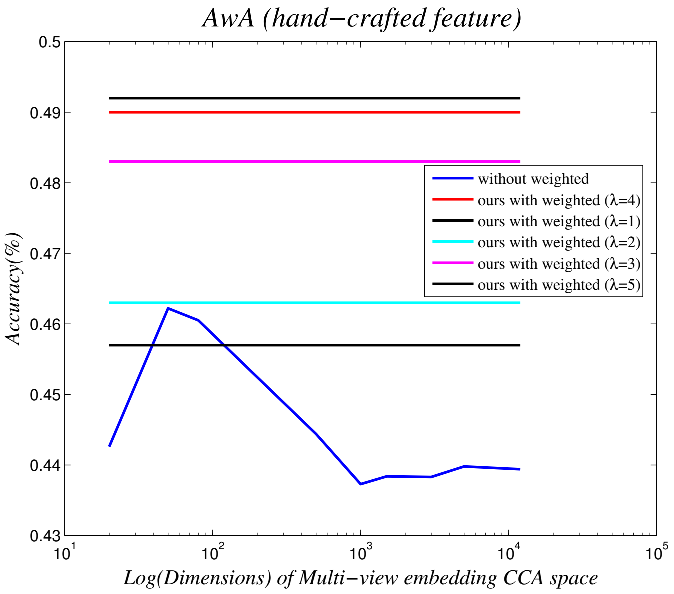


Figure 1: Varying the dimensionaility of CCA for AwA ZSL. Unweighted CCA spaces of dimension 20 to 12010 are compared against our soft-weighting strategy within the full 12010 dimensional space.

(c) how many examples are required to construct CCA? Knowing all test items beforehand is a rather restrictive assumption, but knowing a few could make a lot of sense.

In our paper, we use all the examples to construct the transductive embedding CCA space. However, we follow the reviewer's suggestion to investigate how many examples are required to construct a reasonable embedding. More specifically, we use hand-crafted features (dimension: 10925) of AwA dataset with semantic word vector $\mathcal{V}$ (dimension: 1000) and semantic attribute $\mathcal{A}$ (dimension: 85) to construct the CCA space. We randomly *select 1%, 3%, 5%, 20%, 40%, 60%, and 80% of the unlabelled testing instances to construct the CCA space, followed by our TMV-HLP for ZSL. Random sampling is repeated 10 times. The results shown in Figure 2 below demonstrate that only about 5% of the full set of samples (300 in the case of AwA) are sufficient to learn a good embedding space. Reasonable results can be obtained with even only 1% of the test data. This result has now been included in the supplementary material.*

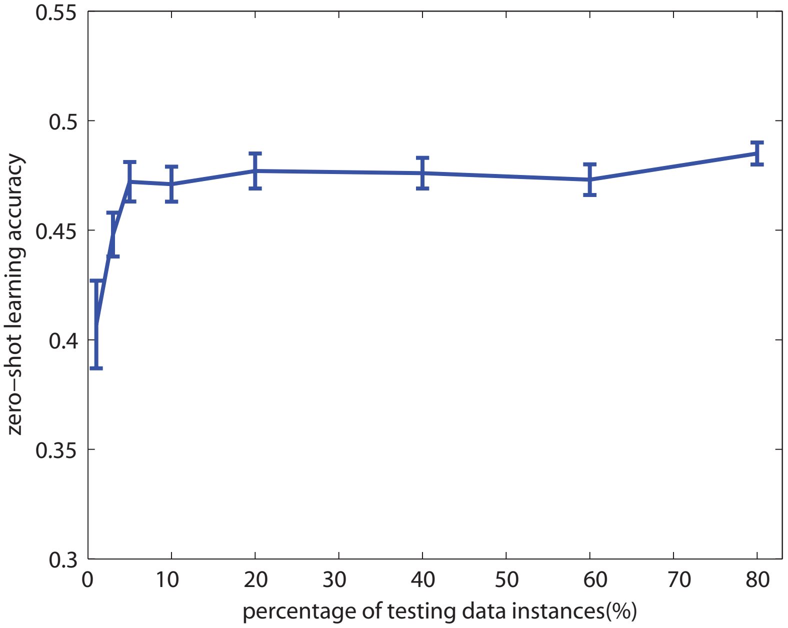


Figure 2: Influence of the number of unlabelled test samples used to learn the CCA space.

(d) Can the CCA be performed using the validation images of the train images (auxiliary images)?

Yes, it can be performed. However, it will then be less useful to rectify the projection domain shift problem since it would then be applied prior to observing the domain-shift. To validate this point, we conduct experiments by using the hand-crafted features (dimension: 10925) of AwA dataset with semantic word vector $\mathcal{V}$ (dimension: 1000) and semantic attribute $\mathcal{A}$ (dimension: 85). Auxiliary data from AwA are now used to learn the multi-view CCA, and we then project the testing data into this CCA space. We compare the results of our TMV-HLP on AwA using CCA spaces learned from auxiliary validation images versus unlabeled test images in Fig. 3. CCA learned by training images gets reasonable performance; however, it does not perform as well as our method which learned CCA transductively using testing images. This is likely due to not observing, and thus not being able to learn to rectify, the projection domain shift. This result has now been included in the supplementary material.

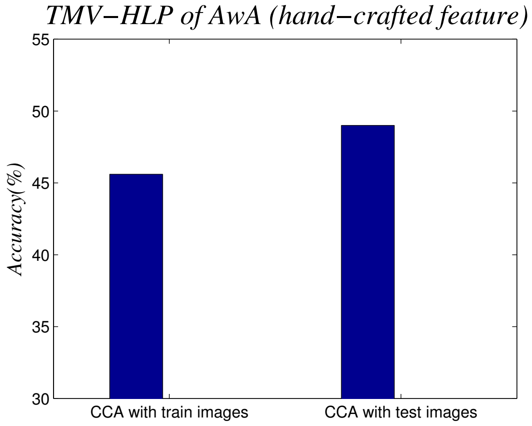


Figure 3: Comparing ZSL performance using CCA learned from auxiliary training images versus unlabelled testing images.

(e) Which part (CCA or Label propagation) is responsible for the improvement? i.e what is the performance of DAP/kNN/SVMs on the CCA space? What is the performance of the label propagation methods on the original space(s)?

Each part of our framework contributes to the improved results. As explained in the Introduction section, we identify three problem for ZSL: the projection domain shift problem, prototype sparsity problem and the challenge of exploiting multiple intermediate semantic views. Multi-view CCA embedding is trying to solve the projection domain shift problem; our label propagation is designed for solving the other two problems.

To validate this, as suggested we conduct experiments to investigate the performance of Nearest neighbour in CCA space and label propagation methods for the original space for all the datasets. The AwA dataset is used with hand-crafted features. From Fig. 4 we can draw the following conclusions: (i) label propagation can slightly improve the zero-shot accuracy in the original semantic word/attribute space (top subplots). (ii) Multi-view CCA embedding improves the performance of simple NN based ZSL compared to the original space (blue versus green bars in lower plots). (iii) Within the embedding space, label propagation further improves the results compared to simple NN based ZSL (green versus red bars in the lower plots). This result thus demonstrates that both the CCA and label propagation component of our framework contribute to the performance.

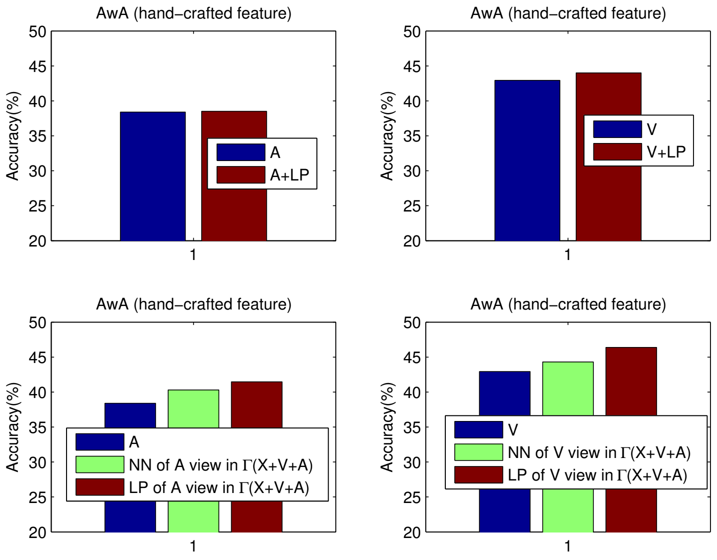


Figure 4: Contribution of CCA and label propagation on the AwA dataset

3. The transductive view on zero-shot learning is interesting, although rather different from the scenario used in the traditional zero-shot methods such as DAP [28/29] and ALE [1]. I'm unsure if a direct comparison is "fair", therefore claiming to have the "best results" so far should be taken with a grain of salt. The chosen setup is really different and that should be highlighted in the experimental section.

   *Thanks for pointing this out. Yes. We do assume that a pool of test instances are given in order to use our framework, i.e., the transductive assumption. However this not an unreasonable assumption in many applications, e.g., unless real-time response to a data stream is required. The new result in Fig. 2 above also suggests that a small portion of the test set would be sufficient to learn a good transductive embedding space. We have now highlighted the 'transductive' requirement in experimental settings (In Table 1 we have added: "***: requires unlabelled data, i.e. a transductive setting."), as well as in Related Work (the 'tranductive zero-shot learning' subsection) and Introduction.*

4. How are the visual feature vector prototypes of each class defined?

   The visual feature vector prototypes of each class are explained at the beginning of Sec. 4 in the manuscript: We have $\psi_c^{\mathcal{A}}$ *and* $\psi_c^{\mathcal{V}}$ *for the attribute and word vector prototypes of each target class* c *in* Γ. *In the absence of a prototype for the (non-semantic) low-level feature view* $\mathcal{X}$, *we synthesise it as* $\psi_c^{\mathcal{X}} = (\psi_c^{\mathcal{A}} + \psi_c^{\mathcal{V}})/2$. *For empirical validation about the visual feature vector prototypes, please refer to our response to R2-Q1 for more details.*

5. How is the result of the random walk used as classification?

   The ZSL problem scenario is that the embedding space is populated with a set of unlabelled test images, as well as a handful of labelled zero-shot prototypes (one for each class). Our work (also [10, 1]) interprets this ZSL problem as a semi-supervised learning problem: A prototype provides a single labelled "instance" of each zero-shot class. These should be used to classify the unlabelled data in a way that reflects its manifold structure. Graph-based semi-supervised learning methods provide a mechanism to achieve this by propagating labels from the labelled prototypes to the unlabeled images along the graph. A theoretically well-studied approach to realising graph-based semi-supervised learning is the random walk model [11, 12], which has a closed form solution and is thus efficient to compute. Given the constructed graph (Sec. 4.1), transition probabilities $p\,(k \rightarrow l)$ *along edges are computed. Labelled nodes (in our case prototypes), are initialised with their true labels. These labels and then propagated to all the unlabelled nodes through random walk. Unlabelled instances are classified according to the label they get after iterating the random walk transitions. This has a simple closed form solution* $\hat{Z} = \eta(\eta\Pi + \mathcal{L})^{-1}\Pi Z$ *[12].*

   We have now explained this more intuitively by revising the start of Sec. 4.2. We have also now included a framework pipeline illustration to put this in context (Fig. 2 in the revised manuscript).

6. More thorough comparison with [37], as well as other transductive methods?

   There are mainly two works on zero-shot learning in a transductive setting, i.e., Rohrbarch et al. [10] and Fu et al. *[13, 14]. We compare these two works in zero-shot learning (Table 1) and N-shot learning settings (Fig. 7).*

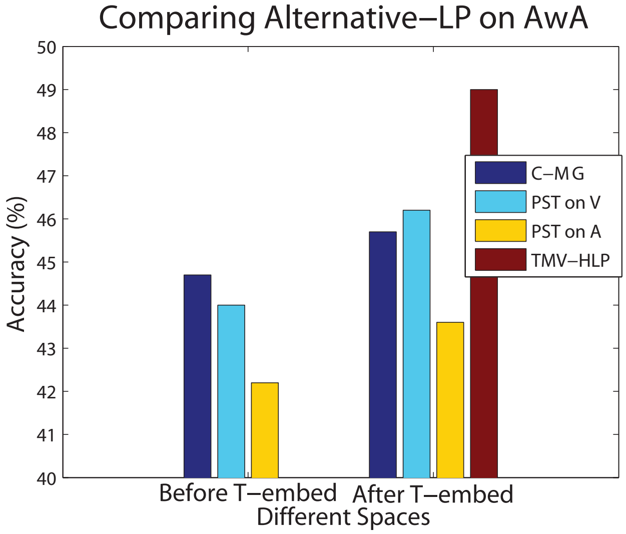


Figure 5: Comparing our transductive learning method against C-MG and PST before and after transductive embedding.

   To more thoroughly compare with PST (Robarch et al. [10], i.e. the mentioned [37]), we add one more group experiment similar to Fig. 2 in Fu et al. [1]. Specifically, we use AwA dataset with handcrafted features, semantic word vector $\mathcal{V}$, and semantic attribute $\mathcal{A}$. We compare with the graph-based semi-supervised learning methods C-MG [12] and PST [10] before and after our transductive embedding (T-embed). We use equal weights for each graph for C-MG and the same parameters from [10] for PST. From Fig. 5, we make the following observations: (1) TMV-HLP in our embedding space outperforms both alternatives. (2) The embedding also improves both C-MG and PST, due to its ability to alleviate projection domain shift via aligning the semantic projections and low-level features.

   There are a number of reasons why our transductive learning framework outperforms PST: (1) Using multiple semantic views (PST is defined only on one), (2) Using hypergraph-based label propagation (PST uses only conventional 2-graphs), (3) Less dependence on good quality initial labelling heuristics required by PST – our TMV-HLP uses the trivial initial labels (each prototype labelled according to its class as in Eq. (17)).

   This more thorough comparison and discussion have been included in the supplementary material.

7. The differences between domain shift in general transfer learning and projection domain shift? And a statement like: "renders any efforts to align the two domains directly unfruitful" (in the related work section) asks for an (experimental) validation.

   The projection domain shift problem tackled in this work is a related but different problem to the conventional domain shift in general transfer learning. It can be considered as a non-trivial generalisation of the general domain shift problem in computer vision. Classic domain shift problems are caused by the different distributions of low-level features for the same task (e.g., high vs. low resolution images, $p(X_t) \neq p(X_s)$ where X_t and X_s are the target and source data features respectively [15, 16]). In contrast, in the projection domain shift problem: (1) from the perspective of the low-level feature to semantic space projection, the low-level feature distribution has indeed changed ($p(X_t) \neq p(X_s)$), because the input images from which we should predict, e.g., hasTail attribute have changed from pigs to previously unseen zebras). (2) This type of change moreover means that the conditional distributions have also changed $p(Y_t|X_t) \neq p(Y_s|X_s)$ where Y_t and Y_s are the target and source data labels respectively [15]). In addition, (3) since we are using the predicted attribute/word vectors as input (to the main ZSL task), the effects of the changing input image distribution are amplified and complicated by being propagated through the (non-linear) attribute/word-vector projections.

   To validate this and demonstrate that for this new problem, new solution is required, new experiments are carried out. Note that in the ZSL problem context, there is no labelled data in the target domain. This greatly limits the suitable domain adaptation methods as most of them require labelled data from the target domain. One of the most classic *methods to address domain shift is the Kernel mean matching method [17, 18]. This method re-weighs the training data such that its distribution more closely matches that of the test data. (However like most domain adaptation methods, it assumes that the conditional probabilities $p(Y|X)$ are unchanged for training and testing data, where X indicates the low-level feature and Y, the labels.) We evaluate KMM as an alternative on the AwA dataset: we use the low-level features of auxiliary (training data) and testing data to compute the weights by KMM (kernel mean matching). These are in turn used for adapting each low-level feature dimension of the AwA testing data to make it more similar to the training set. Using AwA dataset hand-crafted features, we generate kernel matrices following Lampert* et al. *[19, 20]. We can thus compare attribute detection AUC values with/without KMM in Fig. 6. For some attributes, KMM methods do give improvement of AUC, probably because the assumption that conditional probabilities $p(Y|X)$ are unchanged is satisfied on these attributes. However, for most of attributes, this assumption is invalid, since the auxiliary and test data come from very different classes (e.g., Pig tail versus Zebra tail). As a result, attempting to rectify the domain shift with KMM actually results in worse performance compared to doing nothing. Specifically, using KMM leads to an average drop of 3.98% in the AUC value for attribute classification, and in turn a drop of 4% in the ZSL classification accuracy. In contrast, the transductive embedding method proposed in this work has shown to be effective in solving the projection domain shift problem (See Fig. 4 in the revised manuscript and Fig. 4 above in this document.)*

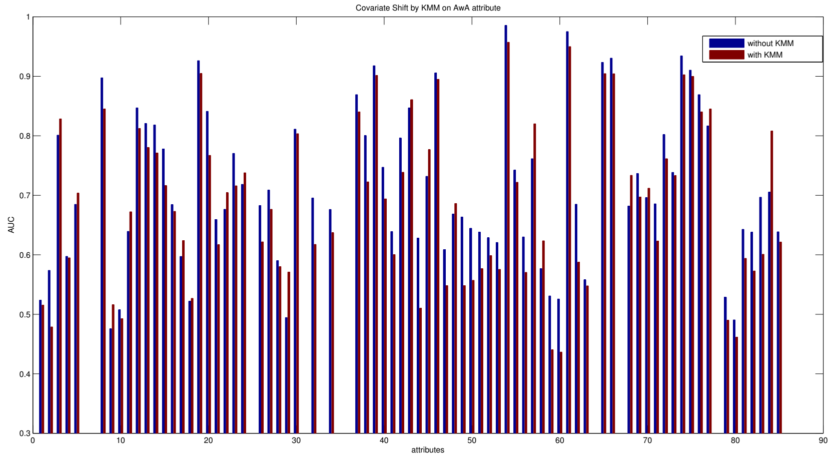


Figure 6: Semantic attribute detection accuracy with/without KMM on AwA dataset.

8. why the notation p(k->l) is used not p(l|k)?

   p(k->l) indicates the probability that a random walker currently at vertex (k) should jump to another vertex (l) at the next walk iteration. It is different from the standard definition p(l|k) which would refer to the probability of random variable l taking on a particular value given the value of random variable k. Note that the notation p(k->l) was first used in the seminar work of Zhou et al. [12], which laid some theoretical foundations of random walk.

9. Section 4.2: the symbol for complexity seems off, and other minor comments

   Thanks for the suggestions. We rectified all the minor comments in the updated version.

   We have now better explained the complexity in the revision: "The computational cost of our TMV-HLP is $\mathcal{O}\left((c_T + n_T)^2 \cdot n_V^2 +\right.$ where n_V is the number of views, and c_T is the number of target classes. It costs $\mathcal{O}((c_T+n_T)^2 \cdot n_V^2)$ to construct the heterogeneous graph, while the inverse matrix of Laplacian matrix $\mathcal{L}$ in label propagation step will take $\mathcal{O}((c_T+n_T)^3)$. However, the efficient label propagation method in the recent work of Fujiwara *et al.* [21] could be used to reduce this term to $\mathcal{O}(c_T n_T t)$, where $t \ll n_T$ is an iteration parameter."

Response to Reviewer 2

1. Since $\psi_c^{\mathcal{X}}$ is synthetically defined, is it possible to understand a posteriori how much it is reliable as a prototype?

   Thanks for the question. In our existing experiments, the synthetic prototype in the low-level feature space has been used with other (real) prototypes in the semantic spaces to achieve state-of-the-arts results. To evaluate its usefulness on its own, we have done additional experiments. Specifically, the synthesised visual prototype $\psi_c^{\mathcal{X}}$ is used to do ZSL via simple nearest neighbour classification solely within the low-level feature view in the embedding spaces. The result in Table 1 shows that the synthesised prototypes give competitive results in a simple 1-NN classification context, demonstrating that they are meaningful and useful. Note that this synthesis is possible due to our embedding space step, which makes all views comparable and represented with the same number of dimensions.

	AwA($\mathcal{H}$ [20])	*AwA ($\mathcal{O}$)*	*USAA*	*CUB ($\mathcal{O}$)*
$\psi_c^{\mathcal{X}}$	*45.3*	*64.6*	*43.6*	*39.6*

   Table 1: Evaluation on the usefulness of the synthesised prototype in the low-level feature view

2. I would suggest to explicitly indicate the dimensionality of Φ^i (i.e. $\mathbb{R}^{n_T \times m_i}$) and W^i (in lines 10-12 of page 5). In particular a paragraph may be dedicated to explain why the embedding space dimensionality should be the sum of the separate original space dimensionalities.

   Thanks. We now explicitly indicate the dimension of Φ^i and W^i in Sec. 3.3 ("The dimensionality of the embedding space is the sum of that of Φ^i, i.e. $\sum_{i=1}^{n_V} m_i$; so $W^i \in \mathbb{R}^{m_i \times \sum_{i=1}^{n_V} m_i}$"). There we also have a paragraph to explain why with a soft weighting strategy we can work with the sum of all dimensionalities of different views rather than performing hard dimensionality reduction. We have also performed new experiments to validate that this soft weighting strategy is better than the conventional hard dimensionality reduction. Please Refer to R1-Q2(b) for detailed explanations and experimental results.

3. What about projecting all the views to a shared subspace and reducing the dimensionality instead of enlarging it? Subspace methods have proven to be effective for different domain adaptation problems (e.g. [12]) and can be good solutions also for heterogeneous zero-shot learning.

   Thanks for the comment. As we explained in R1-Q2, although we technically enlarge the space, our dimension weighting method effectively provides a soft dimensionality reduction within this enlarged space. We have obtained new experimental results to show that this soft weighting is better than the conventional approach of hard dimensionality reduction. Re: domain-adaptation versus domain-shift, please refer to R1-Q7.

4. I find a bit confusing the use of "k" in section 4.1. At the beginning it refers to every node in view i (lines 32-33 page 6), but then in the second column of page 6 it is used in the sentence "We select the first k highest values from ...". Please rephrase the sentences to explain better what "k" refers to.

   Yes, thanks for pointing it out. We have changed it to "We select the first K highest values from ...". Capital K is now consistently used throughout to indicate the number of nearest neighbours. Lower-case k and l are used to index nodes in Sec. 4.

5. - In line 35 of page 6 "k = 1, ...n_T+c_T": if I understand correctly both the unlabelled samples and the prototypes are considered as nodes. Please add an explicative sentence here.

   Yes, we have now stated this explicitly.

Response to Reviewer 3

1. Readability. The grammar and English usage is good, but I found it took several reads, significant effort, and the reading of several reference papers to understand and gain an intuition for what is being done, and I am still uncertain about key aspects. One pattern in the manuscript is that the technical explanations start abstract and heavy with notation or equations, and the intuition or examples come much further in the manuscript, if at all. Perhaps it reads fine if you already understand the work, but as a first-time reader, I might have given up on it if I weren't a reviewer, and for maximum impact you want your paper to be accessible to TPAMI readers. This is a difficult thing to fix, I've made many suggestions below that I hope help.

   Thanks for the suggestions! We have revised the typos and made special efforts to address all the comments of Reviewer 3 (Please refer to the following points for specific problems). We also put the intuitive explanations before the heavy and abstract equations or notations for most of the technical sections.

2. Comments on CCA.

   (a) The paper makes strong claims about being the only paper to identify and address domain shift, but the manuscript doesn't provide an intuition for how the way they applied CCA rectifies domain shift...and for readers without experience in this area, the claim seems dubious. And the experiments don't give experimental evidence that it does.

   Thanks. Indeed, the paper will benefit from an intuitive explanation on how the domain shift problem is rectified. First, we need to emphasise that the problem at hand is projection domain shift, rather than the conventional domain shift. It is caused by a projection function learned from the source domain and applied to project different classes in the target domain in the same semantic space. As shown clearly in Fig. 1 and Fig. 5 in the revised manuscript (Fig. 4 in the old version), this projection domain shift has two negative effects on classifying the target domain classes using the target class prototypes. First, the prototypes are far away from the projected data points (see Fig. 1 the Pig class). Second, the target classes are inseparable after the projection (see the data distribution in the attribute and word vector views in Fig. 5).

   So how can our CCA framework rectify this two negative effects? It is easy to see why CCA embedding can rectify the second negative effect. In particular, after transductively (using the unlabelled target data) aligning the semantic views with each other, and with the low-level feature view, the complementarity of different views are exploited in a shared subspace. The end result is that data belonging to different classes become more separable – they may be inseparable in one view, but with three or nine in the case of Fig. 5, they have a much better chance to be separable after the views are aligned. This (multi-view CCA embedding making data more separable thus easier to classify) has been demonstrated extensively by previous works [2, 3, 4], as well as in our experiments (see Fig. 6 in the revised manuscript). It is also illustrated clearly in Fig. 5 (Fig. 4 in the old version). As for the first effect, i.e. the target class prototypes and the projected class members are far away, it is hard to come up with a theoretical explanation why by correlating different views, the data projections in the CCA space should be closer to the projection of their prototype. One intuitive explanation is that since the CCA projections for different views are learned together from the target class data, when the learned projection matrices are used to project a prototype point on to the same space, it would naturally draw them closer to its class members. From another perspective, when the biased target data projections (semantic representations) are correlated/aligned with their (unbiased) low-level feature representations, the bias/projection domain shift can be alleviated. Importantly, empirically we found it is indeed the case, that is, the prototypes are much closer to their class members after the CCA embedding.

   We have now added a more intuitive explanation of this in Introduction and at the beginning of Sec. 3.3. For provide additional experimental support, we have also included a new set of experiments to thoroughly validate the efficacy of CCA in the supplementary material. Please refer to R1-Q2 for more details.

   (b) As for the experiments, the text for Fig 3 says the experiments show that domain shift is rectified, but those experiments do not isolate CCA as beneficial; for the experimental settings where CCA is used, they also use TMVHLP (the hypergraph algorithm), which conflates the two parts of their approach. Because the experiment always combines CCA and TMVHLP, it doesn't clearly show that domain shift existed or is being addressed by CCA. Could you instead compare nearest neighbours with and without CCA? This would demonstrate that NN before and after transduction has addressed domain shift in the way the paper claims it should.

   Thanks for the suggestion. The same has been asked by Reviewer 1 as well. We have now add new experiments to isolate the benefits of CCA by comparing nearest neighbour-based ZSL with/without CCA. The result is shown in Fig. 4 in this document and has been included in the supplementary material. Please see R1-Q2(e) for details.

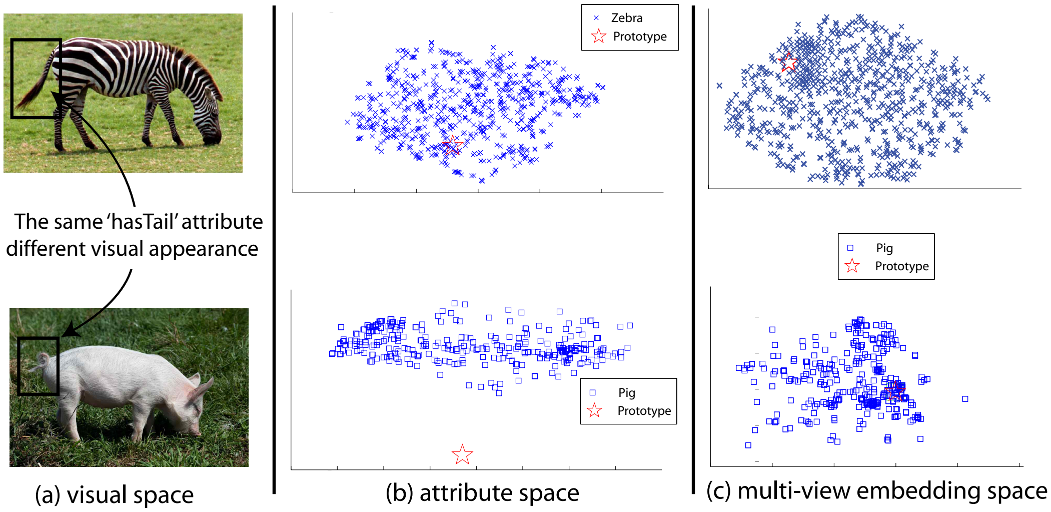
(c) It also isn't clear how the control experiment (e.g. 'V' data point) was set up (e.g. is f^A(X_T) used?).

In Fig. 3, the 'V' experiment is NN within the raw word-vector space, so it (solely) uses $f^{\mathcal{V}}(X_T)$ to represent each image, and matches them against the word vector prototypes $\Psi^{\mathcal{V}}$ (so $f^{\mathcal{A}}$ is not used). Similarly the 'A' experiment solely uses $f^{\mathcal{A}}(X_T)$.

3. Related to (2) above I have major misgivings about how well the model could still classify the nonzeroshot (auxiliary classes). My best guess is that CCA skews the embedding space in favor of the zeroshot classes at the expense of the target classes. For example,

(a) what would Zebra look like in Fig 1 if you showed it after CCA?

Thanks. We updated Fig. 1 (see Fig. 7 below) to include the Zebra representation within CCA space. There is not significant detriment to the representation of zebra in terms of alignment of the prototype with its class members.



(a) visual space

(b) attribute space

(c) multi-view embedding space

Figure 7: Illustration of the projection domain shift problem.

(b) In most real world settings where one has an instance to categorize, one would not be told whether it is from a known or unknown class. .... If this is a drawback of the approach, it makes me very uncomfortable that the paper doesn't address this or even mention it. There is a sentence in the conclusion that appears to hint at this but doesn't directly address it ("The current framework needs to be extended to first distinguish these two types of data before performing zeroshot recognition"). If this is a drawback, be upfront and honest about it, similar to Socher et al. NIPS 2013 on Zeroshot.

Indeed, it is a limitation, shared with most existing studies. We have now stated this limitation up-front in Related work: "all these studies and ours (as with most previous work [32], [31], [41]) only consider recognition among the novel classes: unifying zero-shot with supervised learning is an open challenge [47]."

(c) I have major misgivings about how well the model could still classify the nonzeroshot (auxiliary classes). My best guess is that CCA skews the embedding space in favor of the zeroshot classes at the expense of the auxiliary classes

This is an interesting questions. To answer that, we conduct the following experiments on AwA dataset with overfeat features (three views: 4096 dim overfeat, 1000 dimensional semantic word vector, and 85 dimensional semantic attribute views). The CCA embedding space is learned transductively using testing (target class) data as usual. To test the validity of CCA space for the auxiliary instances, we do a 50:50 train:test split for the images for auxiliary classes (in total 24295) and perform conventional SVM-based supervised learning with/without the CCA representation. We use this setting because: (1) overfeat features are a stronger representation so a simple classifier such as linear SVM will achieve reasonable results. In contrast, hand-crafted features need multi-kernel learning to have good performance. (2) Since the number of all auxiliary instances is 24295, linear SVM is needed for efficient classification.The experiment is repeated 10 times to reduce the variance. The results in Fig. 8 show that in the CCA space, supervised learning of auxiliary classes is only slightly worse than in the *original space. This demonstrates that CCA does not significantly skew the embedding space at the expense of the auxiliary classes.*

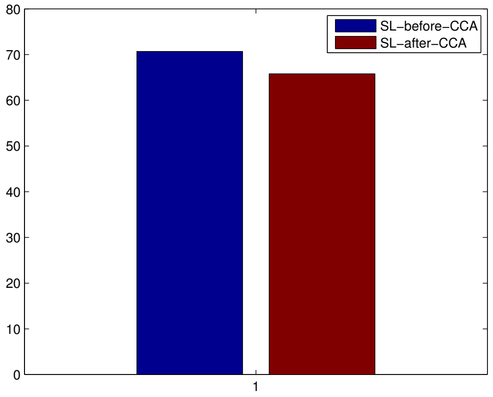


Figure 8: Supervised learning (SL) to classify auxiliary classes before/after projection to the CCA space learned from target classes.

4. explain "semantic representation" and "prototype" in introduction as well as highlighting the 'transductive'

   We have now explained "semantic representation" and "prototype" earlier in the paper (Intro Para 2). Specifically, for semantic representation, we added "Common semantic representations include binary vectors of visual attributes [31], [35], [17] (e.g. 'hasTail' in Fig. 1) and continuous word vectors [36], [13], [47] encoding linguistic context.". For prototype, we have "Depending on the semantic space, the class prototype could be a binary attribute vector listing class properties (e.g., hasTail) [31] or a word vector describing the linguistic context of the textual class name [13]."

   *We have also highlighted 'transductive' in the introduction (Para 6): "we propose to solve the projection domain shift problem using a transductive multi-view embedding framework. This transductive setting means using the unlabelled test data to improve generalisation accuracy." We also added another sub-section in Related Work for transductive zero-shot learning, and highlighted transductive settings in the experiments section (Table 1: "***: requires unlabelled data, i.e. a transductive setting.").*

5. Comments on P2 L29-43.

   (a) Various questions to Fig. 1 and its caption description.
   Thanks for the suggestions. We have now rephrased 'image feature projection' to 'predicted semantic attribute projections (defined in Sec. 3.2)'.

   (b) Is the domain shift a failure of the regressor?
   This is an interesting question. To a certain extent, the answer is yes. A significant change in image statistics (e.g., due to moving from one animal category to another such as pig and zebra) clearly introduces a domain shift from the perspective of the attribute classifier/word vector regressor. e.g. the tail classifier. Therefore if attribute classifier is so robust and generalises so well such that it works perfectly for any unseen animal (e.g. we have a super tail classifier which can recognise any forms of tails), then the projection domain shift could be eliminated. However we also know that the existing computer vision systems are still struggling with basic object classes such as car, person; the harder problem of attribute classification is far from being solved. Therefore, we have to live with the imperfect regressor/classifier and deal with the resultant projection domain shift problem.

   As requested, we have updated Fig. 1 (main manuscript)/Fig 7(this document) to include the Zebra in the embedding space.

6. P2 L49-52: This can be written to more clearly... P2 L54-56. This was one of the focuses of Prototype Theory that was studied by psychophysicists before Computer Vision researchers, e.g. Eleanor Rosch in the 1970s. It will rub many readers the wrong way for you to claim you are the first to identify this problem.

   Thanks. We rephrased this part and add a reference to the work of Prof. Rosch.

7. P2 R8: This is the first time the term "views" has been introduced, and instead of being defined it is left up to the reader to understand what you mean from context.

   Thanks. We rephrased this point to make this explicit.

8. P2 R48: The solution to the prototype sparsity problem is to "explore" the manifold structure. That's a vague term, it would be better to use a word that provides some better intuition as you introduce this part of your approach.

   The manifold structure is referring to standard manifold assumption in semi-supervised learning [11], i.e. The data lie approximately on a manifold of lower dimension than the input space, and that modelling the manifold for each class can help guide classification.

9. P2 R52-58: The description of heterogeneous hypergraphs at this point in the text is heavy on terms but doesn't convey any meaning to a reader new to the concepts, so this section of the paragraph doesn't provide useful information to many of your readers. If you want to stay at a high level here, find a way to convey the idea of what's being done without relying on terms that will only become clear later.

   Thanks for the suggestion. We have now revised that paragraph by adding more intuitive explanations to better explain the intuitive motivation of the heterogeneous hypergraph.

   More specifically, we highlighted that hypergraph is used in place of 2-graph to gain robustness against data noise. In addition, the way we construct the heterogeneous hypergraph allows us to exploit the complementarity of different semantic and low-level feature views, as well as the manifold structure of the target data to compensate for the impoverished supervision available in the form of the sparse prototypes. Later we further explain (Sec. 4.1) that (1) Hypergraphs typically gain robustness at the cost of losing discriminative power – it essentially blurs the boundary of different clusters/classes by taking average over hyperedges. (2) Heterogeneous hypergraphs can explicitly explore among different views the complementary information which is additional to the discriminative information of each individual view.

10. P4 R22-23: I could not find a description of how attribute words are turned into vectors. I only found here that "A is typically manually defined using a standard ontology". I was unclear through the paper whether each attribute is embedded individually or whether they were combined in some way and embedded as a single attribute vector per class. Please make that explicit at this point in the manuscript.

    The attribute vectors are defined by expert designed ontology following standard practice [19, 22, 23]. For example, suppose there are three attributes: 'has tail', 'has fur', 'living in land'. Then the classes Whale and Zebra will be assigned the attribute vectors as (1,0,0) and (1,1,1). This is a manual process relying on human knowledge rather than the text strings of the attribute names. In contrast, the word vectors are defined automatically by a skip-gram neural network model learned using a large unannotated text corpus, which maps any English word into a high-dimensional word vector. For example, using the text string 'leopard' as input, the skip-gram model will give us a 1000-dimension word vector.

    We now add such description in the Introduction: "the semantic representation used in existing approaches varies from visual attributes [30], [11], [34], [17] (e.g. 'hasTail' in Fig. 1) to semantic word vectors [35], [13], [46] and semantic relatedness [40]."

11. P4 R40-41: the SV regressors used to map instances into semantic space are introduced here, and they play a key role as the starting point, the basis for learning a multi-view space, and a means for experimental comparison, but how they were trained is not mentioned at all. They are also the regressors that are used to demonstrate the key idea in Fig. 1. The text doesn't even say what dimensionality the regressors project to.

    SVMs/SVRs are trained as follows: We train a SVM/SVR predictor for each dimension of the attribute/word vector separately. The attribute ontology for AwA defines an 85 dimensional binary vector for each class, and the word-vector neural network defines a 1000 dimensional continuous vector for each class. This means for AwA we train 85 independent SV classifiers and 1000 independent SV regressors to map each image to the corresponding dimension of the attribute/word vector. So overall the bank of SVMs projects to 85 and 1000 dimensions respectively. This is the standard approach/pipeline used by [19, 23, 22, 10, 14] to map low-level features to attribute/word vectors.

    We have now better illustrated this point by adding a new pipeline figure in the paper (Fig. 2). In Sec. 3.2 we also better explained the SVC/SVR training: "In this paper, using the auxiliary set S, we train support vector classifiers $f^{\mathcal{A}}(\cdot)$ and support vector regressors $f^{\mathcal{V}}(\cdot)$ for each dimension of the attribute and word vectors respectively", and in Sec. 6.1 the dimension of the word vector space (1000 throughout) is given, as well as the attribute space for each dataset (AwA: 85, CUB: 312, USAA: 69).

12. P4 R51-56: This section describes the multi-view embedding, but it starts by defining variables and follows with an optimization equation. The section does not give an intuition for what CCA is doing in this setting, which is very useful for making sense of the math. Please address in an introductory explanation how it could be that associating the domain-shifted semantic views with the zero-shot instances improves the domain shift for zero shot classes.

    Thanks. For an intuitive explanation on how CCA can address the projection domain shift problem, please see our response to Point 2(a) above. We have now added some intuitive explanations about CCA in Sec. 3.3.

13. P5 L11-12: Either somewhere around here or in the experimental section it would be good to say what the dimension of the embedding space is, e.g. the size of W_i that you use. I could not find this number anywhere in the paper. P5 L19-20: I found it confusing that it says "The dimensionality of the embedding space is the sum of that of \Phi^i, i.e. \sum_{i=1}^n_V m_i". The dimensionality of the space is determined by the size of the Ws. Do you mean that the points lie in a subspace of that dimension? Or do you size W_i to map to that dimensionality?

    Yes, we size W_i to map to an embedding space whose dimension is the sum of the input dimensions. We have now explained more clearly in the updated manuscript Sec. 3.3 between Eq(1) and Eq(2):"Compared to the classic approach to CCA [22] which projects to a lower dimension space, this retains all the input information including uncorrelated dimensions which may be valuable and complementary. Side-stepping the task of explicitly selecting a subspace dimension, we use a more stable and effective soft-weighting strategy to implicitly emphasise significant dimensions in the embedding space. This can be seen as a generalisation of standard dimension reducing approaches to CCA, which implicitly define a binary weight vector that activates a subset of dimensions and deactivates others." Please refer to Q1-R2 for further explanation and evaluation about the soft-weighting strategy we adopted for the CCA space.

14. P5 L54: "After alleviating the projection domain shift problem... ". At this point in the paper I am entirely unconvinced as I don't have an intuition for the algorithm and I have seen no experimental results. I suggest toning down the claim here.

    Please refer to our response earlier to Point 2 above. We hope the intuitive explanation and new experiments have convinced you that CCA does address the projection domain shift problem.

15. P5 R17-21: At this point on my first read, I began to lose track of what vectors go through which functions and combined in which ways. You could address this by adding a box and arrow diagram to the paper illustrating how the data is being processed.

    Thanks for the suggestion. We have added a new pipeline figure (Fig. 2, new manuscript) that provides an overview. Specifically, the images X_t are projected into word-vector and attribute space, resulting in three sets of vectors $[X_t, f^A(X_t), f^V(X_t)]$. All three of these are projected into the CCA space, producing respectively the three projections $[\Psi^X, \Psi^A, \Psi^V]$. These form the unlabelled data points to be classified. The prototypes $[\psi_{c,}^{X}\psi_c^A, \psi_c^V]$ are then introduced as labelled data. At this point simple KNN classification could be used (see also R1-Q2(e)). However, to leverage the manifold structure of the unlabelled data to further improve the results, we use graph-based semi-supervised learning to do the classification.

16. P5 footnote 4: "the prototypes are updated by one-step self-training as in [16]". This seems like potentially important information. As a reader, I need to read [16] to understand what self-training even is and why it's useful. Can you at least say something at a high level about what it is and why you do it? Is it a common step in these approaches?

    Thanks for the comment. Recall that our interpretation of the ZSL task is as a semi-supervised learning problem with one labelled "instance" (the prototype) for each class. Self-training (updating a class mean to be the the average of its unlabelled nearest-neighbours) is a classic semi-supervised learning strategy – the simplest one available [11, 24]. However it is clearly very coarse, and applying it for many iterations risks model drift [11, 24]. Thus our strategy is to use it once in order to get a good starting point before applying the significantly more elaborate TMV-HLP hypergraph label propagation which is one of our main semi-supervised learning contribution.

17. P6 L17-18: having the same number of similar and dissimilar pairs is a strange assumption given that you have several classes.

    This strategy is inspired by [25] where it is applied to determine Gaussian-kernel scales for SVMs. The objective is simply to ensure that the kernel scale and the data scale are well matched and to avoid the situation where every data point seems near or every data point seems far, which could happen if the scale is not set properly (i.e. too big or too small). Note that this does not imply that the "near" and "far" points need to correspond to the same class and different class data.

18. P6 L36-37: "includes the nodes in view j that are the nearest to node ..." - how is "nearest" defined? A distance cut-off? A number k of nearest nodes? Everything else appears to build on this, but it's left vague.

    *Thanks. It is defined as the K nearest nodes according to a similarity measure. To avoid confusion, we have now replaced the word 'nearest' with 'most similar', which is defined by the similarity. We have also rewritten this subsection and the word 'nearest' is explained clearly in the next subsection (subsection "**Similarity strength of hyperedge** $e_{\psi_k^i}^{ij}$").*

19. Did you try experiments on ImageNet using just word vectors (no attributes)? There are several other works that show results on the 800/200 split, e.g. PST, DeVise, and use features to DECAF.

    We have now conducted some initial experiments on ImageNet dataset. Specifically, we used the ILSVRC 2013 1K data as in [26] and the 800/200 class split for zero-shot learning. Overfeat implementation of CNN features is used together with a word vector space of 100 dimension. The projection between the two spaces is learned using SVR. This experiment is designed to validate that our CCA embedding framework can alleviate the projection domain shift problem. We compare the results with/without CCA embedding using simple nearest neighbour classifier. Without CCA embedding, NN classifier achieves 9.32% *mean accuracy while in our CCA embedding space, the NN classifier can achieves* 19.21% *mean accuracy. If we use the Flat hit@5 accuracy, the number becomes* 41.57%*. This further supports the effectiveness of our multi-view CCA spaces in alleviating the projection domain shift problem. Note that the results is not directly comparable to that in [26] because (1) different CNN implementation is used; and (2) the Overfeat CNN features were learned used the full ILSVRC 2013 1K rather than the 800 classes in the training set. Since we do not have the hardware (i.e. GPU) to re-run overfeat for the 800 classes, we could not conduct experiments under the same setting as in [26].*

20. P6 R59-60: "With the pairwise similarity, one can now create the hypergraphs", which comes after a full page of describing hyperedge formation and hyperedge similarity, which sounded like it was hypergraph creation. If all that is left is a pruning, please describe it that way. Or if the process up to this point was described declaratively, then it may be easier for the reader to understand to also have an imperative description. Have you considered putting in some pseudocode?

    Sec.4.1 as a whole describes all the elements for hypergraph creation. We have now improved the clarity of this paragraph in the revised version. We would love to include a pseudocode, but are already struggling with page limit. We will release our source code so the algorithm can be better understood, if the manuscript is accepted.

21. P7 Fig 2: (1) The image here could be helpful in introducing all the math in the previous section instead of having it follow. (2) typo in the superscripts at "and G^[ji] at the right". (3) I expected here to see the query node clustered with the rest of the hyperedge, was that a misreading or an inaccuracy in the picture?

    Thanks. (1) We have now put the image in the previous section and added more intuitive explanation. (2) we rectified the typo. (3) This image illustrates the process of constructing the heterogeneous hypergraphs. Note that in a heterogeneous hypergraph, a query node only serves to compute the distance from this query node in one view to all the nodes in another view. The query node itself is not used for constructing hyperedges.

22. P7 R24-26: "A typical solution to this is to fuse hypergraphs..." - this again seems like an important detail that's been left unexplained.

    In our framework, we define different types of hypergraphs within and across views. Nevertheless, we still need to combine these hypergraphs to compute a single pairwise transition probability matrix for each pair of nodes in order to apply label propagation. This sentence is an intuitive top level description. In the next section this is explained more elaborately using a more detailed mathematical formulation,

$$\begin{aligned} p\left(k\rightarrow l\right) = & \sum_{i\in\{\mathcal{X},\mathcal{V},\mathcal{A}\}} p\left(k\rightarrow l\mid\mathcal{G}^{i}\right)\cdot p\left(\mathcal{G}^{i}\mid k\right)+ \\ & \sum_{i,j\in\{\mathcal{X},\mathcal{V},\mathcal{A}\},i\neq j} p\left(k\rightarrow l\mid\mathcal{G}^{ij}\right)\cdot p\left(\mathcal{G}^{ij}\mid k\right) \end{aligned} \tag{2}$$

    which explains how hypergraphs (bottom line) are combined with 2-graphs (top line).

23. P7 R31: "Now we have two types... and 2-graphs G^P = {G^i}" - I don't recall seeing an explanation of what the 2-graph is.

    Thanks. The 2-graph here is referring to the standard K-nearest-neighbour graph in the embedding space Γ *for each view. Mathematically, a 2-graph is the conventional type of graph which has edges connecting two nodes. Thus*

2-graph is a special case of hypergraph. We have added one footnote to explain what 2-graph is (Page 8). We have also revised and better explained the 2-graph in the related work (Sec. 2): "a hypergraph is the generalisation of a 2-graph (i.e. one edge connecting two nodes) with each hyperedge connecting a set of nodes (vertices)...".

24. P7 R33-35: "To propagate information from prototypes/labelled nodes ... a classic strategy is random walk". Please add something here that gives an intuition about how doing a random walk results in labels being assigned, before going into the math.

    The intuitive explanation of random walk: Assume a random walk with the current position being at a vertex in one graph. Then, in the next step, the walker may stay, or jump to another node according to the transition probabilities. A subset of vertices is regarded as a cluster or sharing the same class label if during the random walk, the probability of leaving this subset is small while the stationary probability mass of the same subset is large.

    We have now added some intuitive explanation on random walk (Sec. 4.2 Page 8).

25. P8 R37-38: Perhaps because I never fully got how the attributes are represented, I was surprised here that you can reverse the mapping to give a set of attributes. Wouldn't that be a lossy or irreversible mapping? Or is this only giving a single attribute?

    Attributes are represented as a binary vector whose length is the number of attributes in the ontology, and elements assume value one or zero according to if the attribute is present. The CCA mapping is indeed invertible, and it can be used for regression [27], i.e. given one view, one can infer the other view. Conventional (dimensionality reduction based) CCA may be lossy, however since we take a soft-weighting strategy and save all the dimensions in the CCA space (see our response to R3-Q13 above), it not lossy. Therefore the reversed mapping is exact.

26. P9 R8:

    (a) Somewhere in here you need to give the parameters that you used.

    We have now specified the values of all the parameters. Note that the values for some of them e.g. ϖ, and K are specified at first introduction of each. Others are given in the Experiments section.

    (b) What is the dimensionality of the word vectors? How are the attributes turned into vectors?

    The word vector space is trained using the skip-gram model [28] with 1000 *dimensions. The attribute dimensionality varies with the dataset. See R3-Q10 for more explanations about how the attribute vectors are constructed.*

    (c) What dimensionality is the multi-view embedding space? Do these choices make a difference in the results? How did you choose them? Did you put together a validation set?

    The dimensionality of multi-view embedding space is the sum of the dimension of each single view. Our framework does not need to explicitly select a dimension. Please see R1-Q2(b) for explanation and validation of this.

27. P9 R15: In the apples-to-apples comparison using the features that come with AwA, please give your number in the text. The only number written here is the other technique's result. As for the result here, which is 49% vs. 48.3%, did you check whether the 0.7% difference is significant?

    Thanks. We have now put our results in the text as well. For comparing with Yu et al [29]'s 48.3% result, unfortunately we can't check statistical significance without their implementation. However, please note that the reported 48.3% in [29] has a significant constraint not shared by the other methods. It requires additional human annotations to relabel the similarities between auxiliary and target classes. As explained in [29]: "Ten graduate students, who were not aware of the zero-shot learning experiments, were included in the study. When performing the the tasks, they were asked to think about visual similarities, rather than similarities otherwise. The time spent for the task ranges from 15 *to* 25 *minutes." That introduced significant additional external knowledge to obtain the result of 48.3%. Therefore we conclude that it is unfair to compare their result directly with other method's results.*

28. P11 Fig 4: I'm not convinced the Overfeat t-SNE plot is a fair apples-to-apples comparison in 2 dimensions. How many dimensions in the Overfeat features? How many in the multi-view embedding space? They are more separable in 2 dimensions,

    There are 4096 *dimensions of both Overfeat [30] and Decaf. As explained earlier, the dimension of multi-view embedding space is the sum of all views. To visualise the data distribution in these extremely high-dimensional space, some visualisation tool is required. t-SNE is one such tool which has been used widely to visualise data of high* *dimension. We thus believe that it is acceptable to use t-SNE to qualitatively show how separable the data are in different embedding spaces.*

29. P11 L33, "There is no validation set" - there is always a validation set, you just remove some of your training data while you tune parameters.

    Yes, one could do that. However, for the zero-shot learning problem the test set have completely different classes as the training set, so using part of the training set as validation would not yield suitable parameters for classifying the target classes.

30. P12 L16-17: "Our TMV-HLP algorithm is computationally efficient": It's a matter of application whether squared in the number of images is computationally efficient. If one is running on ImageNet, this will not be efficient enough. The scale differential increases from 30 minutes on AwA to about 18 days on ImageNet. You need to soften your claim here.

    Thanks. We have now rephrased this part and added more explanations about the computational cost (please see also R1-Q9).

31. P11 Fig 5: The legend says "Before T-embed", how is that done? My understanding was that the multi-view embedding must be done in order to run TMV-HLP.

    Yes, TMV-HLP makes more sense in the CCA space, as the cross-view heterogeneous graph is constructed based on querying nodes across views; and this cross view query is more meaningful when the views are aligned by CCA. Nevertheless, heterogeneous hypergraphs can still be computed before the CCA embedding. The results show that embedding is very important to make TMV-HLP work – the performance of TMV-HLP without/before embedding degrades dramatically. As shown in Fig. 5, other homogeneous graph based algorithms also produce weaker performance before embedding but not by such a big margin as those graphs are not constructed across views.

32. P12 L32-33: "contrast it to the conventional N-shot learning without the prototypes" - this is confusing, the prototypes can be word vectors, e.g., and are how labels are assigned to the images. How do you do zero-shot without any prototypes? I found this section to be quite confusing.

    Thanks. We have now revised the text to make it clearer. To clarify, we cannot do zero-shot without any prototypes. The +/− symbols are related to N-shot learning and indicate whether solely the N-labelled data per class are used as training data, or they are used together with a class prototype. With N-shot per class, one could learn a classifier without the prototype and this is done by most previous work. The experiments in that section is designed to investigate whether N-shot learning (conventionally used without any prototype) can be improved by the addition of a prototype to compensate for the sparsity of labelled data points in the case of small N.

33. P12 L38: "we modify the SVM- used in [29] into SVM+" - please say a little more about how.

    Same as Q32 above, we use (+) to indicate adding the prototype to the pool of N labelled data and the learned SVM is termed SVM+. So modifying SVM- to SVM+ means simply adding one more instance (actually a prototype rather than real data) per class to the labelled set for SVM training.

34. More explanations of P14 Table 3:

    (a) I found this table and the text describing it very confusing. The query is a list of attributes, but the table shows a class as the query. This is somewhat explained, but the indirection should be removed from the table or describe the columns differently (e.g. ground truth instead of query).

    The indirection in Table 3(a) was necessary because the full list of attributes is 69 elements long in the case of USAA, so cannot fit in there. To avoid confusion, we clarified this by modifying the column header in (a) to '(a) Query by GT attrb. of' and providing better explanation in the caption: "(a) Querying class names by the groundtruth (GT) attribute definitions of classes.". We also modified the text: "Table 3(a) shows queries by the groundtruth attribute definitions of some classes in USAA and the top-4 ranked list of classes returned."

    (b) How many attributes were added in the 3rd row? Can they be listed in the caption?
    Two more were added in the 3rd row. We have made this explicit in the modified version (Table 3(b)).

35. P6 R8-22, R47-50: These sections describe the normalizations and manipulations done to make the hypergraph work. While I'm glad the manuscript includes the details, it reads as though these are hacks that were thrown in to make it work. How much are these needed? Is there there a more principled intuition behind them?

Thanks for the question. Nevertheless, we do not think there are hacks. Some normalisation steps are commonly required to assure reasonable results for almost all machine learning algorithms. For example, the idea of normalisation is also used in SVM in order to maintain reasonable data scale in the feature matrix [31, 32]. Similarly, we do need to do some normalisation to ensure reasonable data-scale when we construct the heterogeneous hypergraphs, since different view comes from heterogeneous data sources and they may have distinctive scales.

Like previous works [33, 34, 4, 35], we use the weight $\delta^{ij}_{\boldsymbol{\psi}^i_k}$ to indicate the similarity strength of nodes connected within each heterogeneous hyperedge $e^{ij}_{\boldsymbol{\psi}^i_k}$. we define $\delta^{ij}_{\boldsymbol{\psi}^i_k}$ based on the mean similarity of the set $\Delta^{ij}_{\boldsymbol{\psi}^i_k}$ for the hyperedge

$$\delta^{ij}_{\boldsymbol{\psi}^i_k} = \frac{1}{\mid e^{ij}_{\boldsymbol{\psi}^i_k} \mid} \sum_{\omega\left(\boldsymbol{\psi}^i_k, \boldsymbol{\psi}^j_l\right) \in \Delta^{ij}_{\boldsymbol{\psi}^i_k}, \boldsymbol{\psi}^j_l \in e^{ij}_{\boldsymbol{\psi}^i_k}} \omega\left(\boldsymbol{\psi}^i_k, \boldsymbol{\psi}^j_l\right), \qquad (3)$$

where $\mid e^{ij}_{\boldsymbol{\psi}^i_k} \mid$ is the cardinality of hyperedge $e^{ij}_{\boldsymbol{\psi}^i_k}$. Such weight $\delta^{ij}_{\boldsymbol{\psi}^i_k}$ intrinsically is used to measure the influence of one hyperedges.

In our framework, the heterogeneous graph is constructed by cross-view querying in multi-view CCA embedding space. Such cross-view querying heterogeneous hypergraph construction however may lead to a varying scale of the weights $\delta^{ij}_{\boldsymbol{\psi}^i_k}$. Dramatic variations in weight scale is bad for the stability of our algorithm. An elegant way to normalise such weights is to enforce the Normal distribution to the pairwise similarities between $\boldsymbol{\psi}^j_l$ and all query nodes from view i by zero-score normalisation as introduced in our methodology. This strategy, also commonly used as an SVM preprocessing step, will improve robustness of later computation,

To illustrate this, we conduct experiments to show that zero-score normalisation makes a more robust cross-view similarity matrix. We only use the cross-view pairwise similarity matrix to construct the random walk graph and then do label propagation. The zero-shot learning accuracy on AwA and USAA dataset of with/without zeros-score normalisation are compared in Table 2. Specifically, we compare 6 different cross-view hypergraphs $\mathcal{G}^{\mathcal{XV}}, \mathcal{G}^{\mathcal{XA}}, \mathcal{G}^{\mathcal{VA}}, \mathcal{G}^{\mathcal{VX}}, \mathcal{G}^{\mathcal{AX}}$, $\mathcal{G}^{\mathcal{AV}}$. For some pairs of views normalisation is not necessary, but for other pairs which may otherwise have suffered from scale mismatch, it greatly improves accuracy. Overall it is consistently beneficial.

AwA	*Norm*	*No Norm*	*USAA*	*Norm*	*No Norm*
$\mathcal{G}^{\mathcal{XV}}$	*46.3*	*43.5*	$\mathcal{G}^{\mathcal{XV}}$	*46.2*	*42.9*
$\mathcal{G}^{\mathcal{XA}}$	*44.1*	*43.7*	$\mathcal{G}^{\mathcal{XA}}$	*49.1*	*44.8*
$\mathcal{G}^{\mathcal{VA}}$	*46.1*	*31.3*	$\mathcal{G}^{\mathcal{VA}}$	*46.9*	*27.8*
$\mathcal{G}^{\mathcal{VX}}$	*45.7*	*13.6*	$\mathcal{G}^{\mathcal{VX}}$	*44.1*	*37.9*
$\mathcal{G}^{\mathcal{AX}}$	*42.9*	*21.9*	$\mathcal{G}^{\mathcal{AX}}$	*50.1*	*37.3*
$\mathcal{G}^{\mathcal{AV}}$	*44.3*	*30.3*	$\mathcal{G}^{\mathcal{AV}}$	*45.3*	*33.3*

Table 2: Comparing classification using cross-view hypergraphs with/without zero-score normalisation in $\Gamma(\mathcal{X}+\mathcal{V}+\mathcal{A})$ on (1000) dimensional word vector.

36. Other minors

    (a) P13 L22: "Zero-attribute learning": this isn't a good name for what is being described here, which is assigning a class from a list of attributes. The same indicates that attributes are not available.

    Yes, indeed. We have changed the "zero-attribute learning" to "zero prototype learning".

    (b) Other typos

    Thanks. We have rectified all the other typos.

References

[1] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong, "Transductive multi-view embedding for zero-shot recognition and annotation," in *ECCV*, 2014.

[2] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *CVPR*, 2009.

[3] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *CVPR*, 2010.

[4] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-H. Zhou, "Multi-view video summarization," *IEEE Trans. on MM*, vol. 12, no. 7, pp. 717–729, 2010.

[5] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, 1998.

[6] D. J. Watts, *Small Worlds: The Dynamics of Networks Between Order and Randomness*, 8th ed. University Presses of California, 2004.

[7] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis; an overview with application to learning methods," in *Neural Computation*, 2004.

[8] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *IJCV*, 2013.

[9] K. Wang, R. He, W. Wang, L. Wang, and T. Tan, "Learning coupled feature spaces for cross-modal matching," in *ICCV*, 2013.

[10] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *NIPS*, 2013.

[11] X. Zhu, "Semi-supervised learning literature survey." University of Wisconsin-Madison Department of Computer Science, Tech. Rep. 1530, 2007.

[12] D. Zhou and C. J. C. Burges, "Spectral clustering and transductive learning with multiple views," *ICML 07*, 2007.

[13] Y. Fu, T. Hospedales, T. Xiang, and S. Gong, "Attribute learning for understanding unstructured social activity," in *ECCV*, 2012.

[14] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Learning multi-modal latent attributes," *TPAMI*, 2013.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Data and Knowledge Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[16] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *ECCV*, 2012.

[17] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *NIPS*, 2007.

[18] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Scholkopf, "Covariate shift by kernel mean matching," 2011.

[19] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE TPAMI*, 2013.

[20] ——, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009.

[21] Y. Fujiwara and G. Irie, "Efficient label propagation," in *ICML*, 2014.

[22] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele, "What helps where–and why semantic relatedness for knowledge transfer," in *CVPR*, 2010.

[23] M. Rohrbach, M. Stark, and B. Schiele, "Evaluating knowledge transfer and zero-shot learning in a large-scale setting," in *CVPR*, 2012.

[24] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, 2009. [Online]. Available: http://www.morganclaypool.com/doi/abs/10.2200/S00196ED1V01Y200906AIM006

[25] C. H. Lampert, "Kernel methods in computer vision," *Foundations and Trends in Computer Graphics and Vision*, 2009.

[26] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Devise: A deep visual-semantic embedding model andrea," in *NIPS*, 2013.

[27] S. M. Kakade and D. P. Foster, "Multi-view regression via canonical correlation analysis," in *COLT*, 2007.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representation in vector space," in *Proceedings of Workshop at ICLR*, 2013.

[29] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang, "Designing category-level attributes for discriminative visual recognition," *CVPR*, 2013.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[31] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001.

[32] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," 2003.

[33] X. Li, Y. Li, C. Shen, A. R. Dick, and A. van den Hengel, "Contextual hypergraph modelling for salient object detection," *ICCV*, 2013.

[34] X. Li, W. Hu, C. Shen, A. Dick, and Z. Zhang, "Context-aware hypergraph construction for robust spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2013.

[35] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: clustering, classification, and embedding," in *NIPS*, 2006.