Bilevel Model Based Discriminative Dictionary Learning for Recognition

– Revision Summary and Replies to Review Comments

Zhouchen Lin, Pan Zhou, and Chao Zhang

First of all, the authors would like to thank the reviewers for bringing up valuable questions/suggestions to improve our paper. We have tried our best to revise the manuscript. All the comments have been addressed appropriately.

In the following, we first provide a "Revision Summary" in Section 1 to list the differences between the original manuscript and the revised one, so that the reviewers and the editor can easily identify what changes we have made. We further provide detailed replies to the review comments in Section 2.

1 Revision Summary

Below are the summary of differences between the original manuscript and the revised version.

1. Following the suggestions from Reviewers #1, #2, #3 and AE, we discuss in detail the differences between our bilevel method with other bilevel dictionary learning methods, including (Yang et al. 2012), (Mairal et al. 2012), (Tao et al. 2014), and (Lobel et al. 2015), in the sixth paragraph of Section 1. Note that we submitted our manuscript **before** the publication of (Lobel et al. 2015).

2. Following the suggestions from Reviewers #2, we discuss the differences between our method with (Guo et al. 2012). Though the model by Guo et al. can be regarded as the unilevel (single-level) version of our bilevel model to some degree, there are differences between these two methods. Please refer to the seventh paragraph of Section 1.

3. Due to the suggestions from Reviewer #1 and page limit, we move the details of the mathematical deduction of Algorithms 1 and 2 to Supplementary Material.

4. Following the suggestions from Reviewers #1, #2, and #3, we add Section 3.5 to discuss the convergence of our optimization method. Admittedly, our method cannot be theoretically proven to converge. But we conduct experiments and report the objective value and find that the objective values reduce reasonably well. Due to page limit, we only report the objective value $\mathcal{L}_2$ in Fig. 1 (a) and (b) on Extended YaleB and Fifteen Scene Categories, respectively. And in the future work, we will further explore the convergence issue of ADM when solving nonconvex optimization problems that have nonlinear linear constraints and K ($K \geq 3$) blocks of variables. This is mentioned in Section 6.

5. Following the suggestions from Reviewers #1, #2 and #3, to make the advantages of our bilevel model more clear, we add Section 4 to discuss the differences between unilevel model and bilevel model.

6. Following the suggestions from Reviewers #3, we discuss the connections and differences between our method and supervised neural networks at the end of Section 4. And we give more details in Section III of Supplementary Material.

7. Following the suggestions from Reviewers #1, #2, and #3, in Section 5.6, we conduct experiments to compare our bilvel method BMDDL with DDL-PC (Guo et al. 2012) and our unilevel version SMDDL to verify the advantages of our bilevel model. Our BMDDL outperforms both DDL-PC and SMDDL, which demonstrates the benefits of bilevel models.

8. Following the suggestions from Reviewers #1, #2, and #3, to demonstrate the advantages of the Laplacian regularization, we add two sentences at the end of Section 5.4.2. We quote them below:

"Note that TDDL [32] is also a bilevel model based dictionary learning method and it replaces the Laplacian term tr(ALA^T) in problem (4) with a regularization $||A||_F^2$. From Tables 3$\sim$8 and Fig. 8, BMDDL achieves better performance than TDDL on the six benchmarks, which also demonstrates the advantages of the Laplacian regularization that encourages similar samples to have similar sparse codes."

And in Section IV of Supplementary Material, we further conduct other experiments to verify the contribution of the Laplacian regularization. We find that in BMDDL, the improvements of recognition rates are mainly achieved by the Laplacian term and our optimization method is more efficient in speed than the stochastic gradient descent algorithm.

9. Following the suggestions from Reviewers #1, #2, and #3, to demonstrate the advantages of our optimization method, we add two sentences at the end of the second paragraph of Section 5.5. We quote them below:

"Note that TDDL [32] replaces the Laplacian term tr(ALA^T) in problem (4) with a regularization $||A||_F^2$, which results in a subproblem that is easier to solve for the subgradient with respect to D via implicit differentiation. From Table 9, we can see that though our optimization method solves a more complex problem, our method is still faster than TDDL, which demonstrates that our optimization method runs faster than the stochastic subgradient descent algorithm."

And in Section IV of Supplementary Material, we explains this in more detail.

10. Following the suggestions from AE and Reviewer #1, we also compare our method with (Lobel et al. 2015), which is also a bilevel model based dictionary learning method, and we report the results in Tables 4 and 5. It should be pointed out that in (Lobel et al. 2015), it needs to extract two kinds of features, HOG and LBP, while our method and other compared methods, such as SRC, DKSVD, LC-KSVD, TDDL, etc., only use one kind of feature, SIFT. But our method still outperforms it. Note that we submitted our manuscript **before** the publication of (Lobel et al. 2015).

11. Due to page limit, we remove Figure 1 in our original version, which shows examples of Extended YaleB. We also only show 5 examples, instead of 15 examples, of Fifteen Scene Categories database in Figure 3. Also, we only show 5 examples, rather than 10 examples, of UCF50 and HMDB51 in Figure 5. (a) and (b), respectively. Finally, we resize Figure 8, which shows performance on the six testing databases with varying dictionary sizes, to a smaller one.

12. All the typos, improper presentations, and other minor comments from reviewers have been properly addressed in the new version. All the suggested references are cited appropriately.

2 Detailed Replies to Review Comments

The replies below are ordered as the questions appearing in the comments.

2.1 To AE:

1. *The specific bi-level model is very similar to existing models, which are not cited or compared to. R1 provides some examples that should be discussed and compared against, since many of the experiments are similar and even on the same databases. In addition, the ECCV 2014 paper "Sparse Dictionaries for Semantic Segmentation" uses a similar strategy of jointly learning the classifier parameters and the sparse dictionary and shows how the subgradient of the upper level objective function with respect to the dictionary D can be computed. The new think in the present paper is the Laplacian regularization, which as pointed out by R3 is very minor. Overall, the novelty of the model needs to be more clearly established with respect to the state of the art on bi-level discriminative sparse dictionary learning.*

Reply: Admittedly, (Mairal et al. 2012, PAMI), (Lobel et al. 2015, PAMI), and (Tao et al. 2014, ECCV) all propose bilevel model based dictionary learning methods. However, there are obvious differences, which we have discussed in the sixth paragraph of Section 1 in the new manuscript.

We also conduct experiments to compare our method with these methods. As the model proposed in (Tao et al. 2014, ECCV) is for semantic segmentation and is not very related to recognition tasks (they adopt the conditional random field energy function, which is usually used in semantic segmentation instead of recognition tasks, as the loss function, and do not do any image classification experiment in their paper), we do not conduct recognition experiments of (Tao et al. 2014, ECCV) in our revised manuscript and only discuss it in the sixth paragraph of Section 1. (Lobel et al. 2015, PAMI) divides an image into L regions and extracts several kinds of features in the regions, such as HOG and LBP. Then in the lower level, they employ the max-pooling method to find a few feature words to construct compact features. But our method uses the whole image rather than patches for dictionary learning and we employ the ℓ_1 norm and the Laplacian term to promote the group sparsity of features in the lower level. The upper level in (Lobel et al. 2015, PAMI) minimizes the combination of the loss function of a linear SVM and the regularization of dictionary. Thus, it also has the second drawback we mentioned previously that the learnt dictionary is optimal for the combination, rather than the classification loss. Accordingly, the dictionary may not be the most discriminative for recognition tasks. Our upper level only minimizes the classification loss, thus it does not have the drawback. So, obviously, (Lobel et al. 2015, PAMI) is very different from ours. Besides, (Lobel et al. 2015, PAMI) is difficult to be reimplemented, since it depends on the number and the regions of patches and the overlap between patches, etc., which is not provided in the paper. The classification framework is not presented in the paper either. Namely, they only present the training process, but not the testing phase. We sent emails to the authors, but they did not provide their code. So we only report the experimental results on Caltech 101 (75.4%) and 15 scenes categories (86.3%) shown in their paper. It should be pointed out that in their paper, they extract two kinds of features, HOG and LBP, while our method and other compared methods, such as SRC, DKSVD, LC-KSVD, TDDL, etc., only use one kind of feature, SIFT. But our method still outperforms (Lobel et al.

2015, PAMI). As for TDDL (Mairal et al. 2012, PAMI), we compare our method with it on the six testing datasets. We can see from Tables 3∼8 that our method outperforms (Mairal et al. 2012, PAMI).

2. *There is lack of theoretical analysis or justification as to why the proposed optimization strategy (apply ADMM to non-convex problems with non-linear constraints) is valid and why it is advantageous with respect block-coordinate descent strategies from a theoretical point of view. This is a serious limitation of the present paper and should be clearly addressed in the revised version.*

Reply: In the revised manuscript, we added Section 3.5 to discuss this issue. We quote it below:

“Admittedly, there is no theoretical convergence support when we apply ADM to solve problem (11). Typically, ADM for less than three blocks of variables usually converges when the problem is convex. Recently, some scholars propose theories to extend the scope of the convergence of ADM. For example, Hong and Luo [45] point out that ADM with K $(K \geq 3)$ blocks of variables can converge when minimizing the sum of two or more nonsmooth convex separable functions which are subject to linear constraints. Hong et al. [46] also prove that ADM is convergent for a family of sharing problems, regardless of the number of blocks or the convexity of the objective function. Those works have extended the scope of ADM with theoretical guarantee. However, as for more complex optimization problems, which contain nonlinear equality constraints, are nonconvex and have K $(K \geq 3)$ blocks of variables, there is no theory that supports the convergence of ADM. But this does not mean that ADM cannot converge. Boyd et al. [47] point out that when solving nonconvex problems by ADM, ADM may not converge, but when it does converge, it will possibly have better convergence properties than other local optimization methods. On the other hand, many scholars have also adopted ADM to solve nonconvex problems with nonlinear equality constraints and more than three blocks of variables and they report state-of-the-art experimental results, such as [9]. To illustrate the convergence of ADM in solving problem (11), we conduct experiments and report in Fig. 1 (a) and (b) the objective value $\mathcal{L}_2$ on Extended YaleB [48] and Fifteen Scene Categories [49], respectively. We can see that the objective values reduce reasonably well."

The reason that we do not use other methods that are mentioned is that those methods cannot, or are difficult to, be applied to solve our optimization problem.

1) Lobel et al. use an alternating minimization algorithm based on the CCCP algorithm (Yuille et al. 2003), which is designed for unconstrained optimization problems whose objective is decomposed as the sum of a convex and a concave term. But our optimization problem is a constrained problem and does not satisfy the condition that the objective should consist of a convex and a concave term either . Thus, the alternating minimization algorithm based on the CCCP algorithm cannot solve our optimization problem.

2) The block coordinate descent methods can easily stuck at non-critical points for nonsmooth problems, such as ours, even for convex problems (Xu et al. 2013). Besides, the block coordinate descent methods are also for unconstrained problems. Finally, Xu and Yin (2013) proposed a block coordinate descent method to solve regularized block multiconvex optimization. But it can converge only when the feasible set and objective function are convex in each block of variables and they can be generally nonconvex. Obviously, for our problem, the constraints with respect to the variable D is not convex. So, those methods cannot solve our optimization problem either.

3) As for the stochastic subgradient descent algorithm, actually it is difficult to be applied to solve our bilevel model. This is because deducing the gradient with respect to D via nonsmooth implicit functions is difficult. We will explain this in the reply to the part II of Question 2 of Reviewer #1.

Xu Y, Yin W. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion[J]. SIAM Journal on imaging sciences, 2013, 6(3): 1758-1789.

3. *There is a lack of detail in the experiments and the comparisons to the point that it is not clear to the reviewers whether the comparison is fair, how parameters were set, what code was used for comparison, etc. Again, this is a serious limitation of the present paper and should be clearly addressed in the revised version.*

Reply: The implementations of D-KSVD, LC-KSVD1, LC-KSVD2, SRC, KSVD, and LLC are provided by their authors. We sent emails to the authors of LRRDL, SLRRDL, TDDL, LSC, and DDL-PC, but for some reasons, the authors did not provide their codes. So we implemented their codes by ourselves. It should be pointed out that LRRDL and SLRRDL are proposed in the same paper. TDDL is mainly based on the LARS algorithm (Efron et al. 2010) which is provided to us by its authors. And the LSC and DDL-PC are mainly based on the feature-sign search algorithm (Lee et al. 2007), which is also provided by the authors. In the experiments, we have tried our best to tune the parameters of these methods.

In all experiments, for fairness, D-KSVD, LRRDL, SLRRDL, TDDL, LC-KSVD, SRC, KSVD, LSC*, LLC* and our BMDDL all use the same feature and the dictionary sizes are the same, too.

When we tuned the hyper-parameters of other methods, we use the grid, which is the SAME as ours, to search the best parameters. In the experiments, we have tried our best to tune the parameters of these methods.

When testing a dataset, we firstly introduce the dataset, then present what kind of features and how many dimensions of the features we used in the experiment, even where the features can be downloaded or how the features are produced. We also point out the parameter settings in our method. And in the second paragraph of Section 5 in the revised manuscript, we add a sentence " Note that the parameters in our model are fixed for each database and determined by n-fold cross validation and the detailed settings are presented in each experimental section.".

2.2 To Reviewer #1:

1. *Authors claim 3 main contributions: 1) Joint learning of mid-level dictionary and top-level classifier, where model is consistent (training and testing problems are the same), 2) Use of group sparsity constraints to take advantage of structure in the data, 3) A novel optimization scheme to solve the resulting model. Respect to the 2 first contributions above, there is extensive previous work in terms of discriminative joint learning of dictionary and classifier, as well as, use of class labels to implement group sparsity constraints. As an example, Lobel et al. "Learning Shared, Discriminative, and Compact Representations for Visual Recognition", PAMI, 2015, propose a bilevel model with group sparsity constraints for visual recognition. Earlier, [30] also proposes a bilevel model, although without group sparsity constraints.*

Reply: We have answered this question in the reply to Question 1 of AE.

Please refer to it.

2. *The authors make a good effort to test their method on several benchmark datasets. However, most of the experimental part is dedicated to present overall recognition rates, without further analysis to support the individual contributions of the paper. As an example, authors claim as a relevant contribution the optimization technique proposed to solve the resulting model. In particular, they indicate that the proposed model can be also solved by using the stochastic subgradient descent algorithm, but with a slower convergence rate and a more complex solution. It will be great if they can support this claim including an experiment that compares the convergence rate of both techniques. Similarly, it will be good to know the contribution of the group sparsity constraint, and the joint learning scheme of the dictionary and the classifier. In terms of this last point, the main differences with respect to the work in [30] are the use of a group sparsity constraint and an alternative optimization scheme, which of these factors explain the slightly superior performance with respect to [30]?*

Reply: We will answer your questions in turn.

I. A comparison between our optimization method and the stochastic subgradient descent algorithm to verify the convergence rate

Actually, it is difficult to apply the stochastic subgradient descent algorithm to solve our model. Typically, when applying the stochastic subgradient descent algorithm to solve bilevel model based dictionary learning, one can follow the following sketch (Mairal et al. 2012, PAMI and Yang et al. 2012, CVPR). Consider the general bilevel model based dictionary learning problem (1):

$$\begin{aligned} &\min_{W,D} f(A,W), \\ &\text{s.t.} A = \min_{A} g(A,D), \end{aligned} \tag{1}$$

where A, W, and D are three variables which need to be solved. A denotes the sparse representation. W represents the parameter matrix of a classifier or other parameters. D is a dictionary. To solve problem (1), firstly, fixing D, we solve the lower level problem and compute the variable A. Then we compute the gradient of $f(A,W)$ with respect to D by the chain rule:

$$\frac{\partial f}{\partial D} = \frac{\partial f}{\partial A}\frac{\partial A}{\partial D}, \tag{2}$$

where $\frac{\partial A}{\partial D}$ can be computed via implicit function. Since A is the optimization solution to the lower level problem, A satisfies the following implicit function.

$$\frac{\partial g}{\partial A} = 0. \tag{3}$$

Thus, we can utilize Eq. (3) to compute $\frac{\partial A}{\partial D}$. Unfortunately, when facing a dictionary learning problem, the lower level problem is usually non-differentiable, e.g., involving the ℓ_1 norm for sparsity. Thus, we can only obtain the subgradient $\frac{\partial A}{\partial D}$, which may lead to a slow convergence rate.

When we use the stochastic subgradient descent algorithm to solve our model, we can obtain the following implicit function:

$$-D^T(Y - DA) + \alpha \text{sign}(A) + \beta AL = 0, \tag{4}$$

where sign(A) carries the signs of A. However, for general readers, it is difficult to compute $\frac{\partial A}{\partial D}$ from (4). Thus, applying the stochastic subgradient descent algorithm to solve complex bilevel models, such as ours, is not easy, as it requires significant mathematical skills.

On the other hand, in our manuscript, we have compared our method with T-DDL (Mairal et al. 2012, PAMI), which is also a bilevel model based dictionary learning method. Indeed, its model is easier than ours, since its lower level problem does not consider the data structure and it has no Laplacian term. Actually, TDDL replaces the Laplacian term tr(ALA^T) in problem (4) in our manuscript with a regularization $\|A\|_F^2$. Thus, when using the stochastic subgradient descent algorithm to solve the model, the L in the implicit function (4) becomes an identity matrix and it is much easier to compute $\frac{\partial A}{\partial D}$. In Table 9 in the revised manuscript, though solving a more complex problem, we can still see that our method is much faster than TDDL, which demonstrates that using our optimization method to solve this kind of problem is faster than using the stochastic subgradient descent algorithm. In our revised manuscript, we mentioned this at the end of the first paragraph of Section 5.5 (" Note that TDDL [32] replaces the Laplacian term tr(ALA^T) in problem (4) with a regularization $\|A\|_F^2$, which results in a subproblem that is easier to solve for the subgradient with respect to D via implicit differentiation. From Table 9, we can see that though our optimization method solves a more complex problem, our method is still faster than TDDL, which demonstrates that our optimization method runs faster than the stochastic subgradient descent algorithm.").

II. The contribution of the group sparsity constraint

We discuss this issue in Supplementary Material. We quote it below. Note that Table 1 above is Table 1 in Supplementary Material. We just copy it here.

"In this section, we conduct experiments to verify the advantages of the Lapla-

Table 1: The effects to recognition rates ($\%$) of the Laplacian term on the three databases. (This table is adapted from Table 1 in Supplementary Material)

Method	Extended YaleB	15 Scene Categories	Caltech 101
TDDL (Mairal et al. 2012)	94.6	92.1	71.5
Our method without Laplacian term	94.8	92.9	71.8
Our method with Laplacian term	**95.5**	**96.9**	**75.5**

cian term. In our paper, we have compared our method with TDDL [5]. As we have mentioned, TDDL is also a bilevel model based dictionary learning method, but it does not consider the intrinsic data structure. Actually, TDDL [5] replaces the Laplacian term tr(ALA^T) in problem (4) with a regularization $\|A\|_F^2$, which results in a subproblem that is easier to solve for the subgradient with respect to D via implicit differentiation. From Tables 3$\sim$8 in the paper, by comparison, we can see that our method outperforms TDDL on the six testing datasets.

To further demonstrate the contribution of the group sparsity constraint, we discard the Laplacian term in our model (set $\beta = 0$) and add a regularization $\|A\|_F^2$ to enhance the convexity of the lower level problem. To accommodate this change, we only need to set $L = I$ in problem (4) in our paper, where I is the identity matrix. Now, our model is the same as the model in TDDL [5]. Then we first replace the lower level with its KKT conditions and then apply ADM to solve the new model. We also use the same initialization strategy in our paper. Namely, we first use KSVD [6] to initialize D, then solve the lower level problem to initialize other variables. We report the experimental results in Table 1. The experimental settings in this section are as described in corresponding subsections in the paper, respectively. We can see that our original BMDDL can achieve better recognition performance than the BMDDL without the Laplacian term, which demonstrates the benefits of the Laplacian term. From Table 1, we can also see that when the models are the same, our method without the Laplacian term only outperforms TDDL slightly. Thus, the improvements of recognition rates are mainly achieved by the Laplacian term. But from Table 9 in the paper, our method is much faster than TDDL, which employs the stochastic gradient descent algorithm to solve its model. Thus, our optimization method is more efficient in speed than the stochastic gradient descent algorithm."

Due to page limit, we only report the results of TDDL on the six testing datasets in Table 3∼8 and do not report our method without Laplacian term in the revised manuscript. We point out this in the last paragraph of Section 5.4 (" Note that TDDL [32] replaces the Laplacian term tr(ALA^T) in problem (4) with a regularization $\|A\|_F^2$. From Tables 3∼8 and Fig. 8, BMDDL achieves better performance than TDDL on the six benchmarks, which also demonstrates the advantages of Laplacian regularization that encourages similar samples to have similar sparse codes.").

III. Joint learning scheme of the dictionary and the classifier

We add Section 5.6 in the revised manuscript to discuss this issue. We conduct experiments and verify its advantage. In the experiment, we transform our bilevel model into a unilevel model, named SMDDL:

$$\begin{aligned}&\min_{W,D,A} \|H - WA\|_F^2 + \frac{\lambda}{2}\|W\|_F^2 + \frac{\gamma}{2}\|Y - DA\|_F^2 + \alpha\|A\|_1 + \frac{\beta}{2}\text{tr}\left(ALA^T\right),\\ &\text{s.t. } \|D_i\|_2^2 \le 1,\ \forall i \in \{1, 2, \cdots, k\}.\end{aligned} \tag{5}$$

Table 2: The comparison of recognition rates ($\%$) between unilevel and bilevel on the four databases. (This table is adapted from Table 11 in the revised manuscript)

Type	Method	Extended YaleB	15 Scene Categories	Caltech 101	Caltech 256
Unilevel	DDL-PC (Guo et al.)	95.3	92.0	71.3	58.3
Unilevel	SMDDL (ours)	95.2	93.3	72.4	58.7
Bilevel	BMDDL (ours)	**95.5**	**96.9**	**75.5**	**59.3**

Then we use ADM to solve it. For fairness, we also adopt the same initialization strategy in our paper and solve problem (22) to compute the sparse representation of testing samples. The experimental results are summarized in Table 2. Note that Table 2 here is Table 11 in the revised manuscript. We just copy it here.

It should be noted that we also report a similar work DDL-PC (Guo et al. 2012, ACCV) in Table 2. This work also solves the problem (5) to compute the dictionary D, classifier W, and sparse representations A_{tr}. But it adopts the feature-sign search algorithm (Lee et al. 2007, NIPS) to solve problem (5). Then it solves problem (6)

$$\min_{A} \frac{\gamma}{2}\|Y - DA\|_F^2 + \alpha\|A\|_1. \tag{6}$$

to compute the sparse representations A_{ts} and uses W to classify A_{ts}. From Table 2, we can see that our method also performs better than DDL-PC (Guo et al. 2012, ACCV). The reason why our bilevel model outperforms SMDDL and DDL-PC is presented in more detail in Section 4 in the revised manuscript. We report the experimental results of unilevel and bilevel model based methods in Section 5.6.

IV. The main differences with respect to the work in [30] are the use of a group sparsity constraint and an alternative optimization scheme, which of these *factors explain the slightly superior performance with respect to [30]?*

We have discussed this issue in the reply to the part II of your Question 1. Please refer to it.

3. *Initialization of the proposed model seems to be a relevant issue that lacks of analysis in the paper, it will be great to show how sensitive is the proposed method to this step.*

Reply: Admittedly, nonconvex problems always have the initialization issue. But the initialization strategy in our manuscript is empirically good. So, there is no need for trying other initialization. In all the experiments, we first use KSVD (Aharon et al. 2006) to initialize D, then solve the lower level problem to initialize other variables. And our method achieves the best recognition rates. It should be pointed out that LC-KSVD1, LC-KSVD2 (Jiang et al. 2013), and TDDL (Mairal et al. 2012) all utilize KSVD to initialize their dictionaries.

4. *How do you initialize the alternative method during the experimental validation?. For the alternative methods, do you use your own implementations?. In all experiments, do you use the same features for all the methods (this is indicated in just some of the cases)?*

Reply: After we trained our model, we can obtain the dictionary D and the classifier parameter matrix W. Then we can follow Section 3.3 in the manuscript to compute the sparse representations of validation samples. Actually, we only need to solve the problem (22) in the manuscript. Following TDDL (Mairals et al. 2012), we directly use the LARS algorithm (Efron et al. 2010) to solve problem (22) and we do not initialize by ourselves.

We have discussed the implementation issue when we answer Question 3 of AE. Please refer to the reply.

5. *Authors indicates that a novel contribution of the method is to subordinate the learning of the mid-level dictionary to the learning of the upper level classifier. It is not clear to me, how is this different to previous approaches that jointly learn the dictionary and classifier (ex. using a coordinate descent optimization approach).*

Reply: In Section 4 in the new manuscript, we present the differences between unilevel model (jointly) and bilevel model (subordinate). We quote it below:

"In this section, we will discuss the advantages of bilevel models. As we mentioned in Section 2.2, most supervised methods directly incorporate discriminative term $F(D, A, S)$ into the objective functions of unsupervised methods and the general supervised dictionary learning model can be formulated as (2). Such a mechanism leads to two drawbacks.

1) Undoubtedly, in recognition tasks, the classification error is our ultimate goal and we need to minimize it directly. However, these unilevel model based supervised methods [6], [7], [10], [19], [20], [21], [22], [23], [24] minimize combinations of the reconstruction error and the discriminative terms, such as the classification error. In this way, the learnt dictionary is an optimal dictionary to the combined terms, rather than the classification error. Accordingly, the performance on recognition tasks may be limited. On the contrary, bilevel models can overcome this drawback as they directly minimize the classification error. The upper level minimizes the classification loss, while the lower level characterizes the intrinsic data structure. The objective of lower level is subordinate to that of the upper level. Therefore, bilevel models achieve an overall optimality in that the dictionary learning is directly connected to recognition.

2) Another drawback of those unilevel model based methods [6], [7], [19], [21], [22], [24] is that the problems for computing the sparse codes in the training and the testing phases are different, making the models inconsistent. The sketch of these methods for recognition tasks can be summarized as following three steps. Firstly, these supervised methods solve problem (2) to learn a dictionary D, the sparse codes A_{tr} of the training samples, and other variables S, such as the classifier parameters in [6], [7], [19]. Then, in the testing phase, since there is no supervised information, those methods have to discard the discriminative term $F(D, A, S)$ in (2) and fix dictionary D to compute the sparse codes A_{ts} of testing samples. Finally, these methods feed the feature A_{tr} of training samples into a classifier to learn its parameters W, then use W to identify the feature A_{ts} of testing samples. Or in [6], [7], [19], they directly use the previously learnt classifier S of (2) in the training phase to classify testing samples. These methods solve different problems to learn the sparse representations A_{tr} of training samples and the sparse representations A_{ts} of testing samples. By this way, the new feature A_{ts} may not be optimal for the classifier W or S which is learnt on the feature A_{tr} of training samples. In contrast, bilevel models do not have the above problem. In the training phase, they solve the lower level optimization problem to compute the sparse representations A_{tr} of training samples, and in the testing phase, they still use the lower level model to compute the feature A_{ts} of testing samples. Thus, the classifier trained on the feature A_{tr} can perform on the feature A_{ts} of testing samples. So, in bilevel models, the problems for computing the sparse codes in the training and the testing phases are consistent."

To better understand, we take our model for example. If we jointly learn the dictionary and classifier in unilevel, the model can be written as:

$$\begin{aligned} &\min_{W,D,A} \|H - WA\|_F^2 + \frac{\lambda}{2}\|W\|_F^2 + \frac{\gamma}{2}\|Y - DA\|_F^2 + \alpha\|A\|_1 + \frac{\beta}{2}\text{tr}\left(ALA^T\right), \\ &\text{s.t. } \|D_i\|_2^2 \leq 1,\ \forall i \in \{1, 2, \cdots, k\}. \end{aligned} \tag{7}$$

Accordingly, we use the model (7) to learn the dictionary D, the sparse representations A_{tr} of training samples, and the classifier parameters W. Then we have to solve the following problem to compute the sparse representations A_{ts} of testing samples:

$$a^* = \operatorname*{argmin}_a \frac{\gamma}{2}\|y - Da\|_F^2 + \alpha\|a\|_1 + \frac{\beta}{2}\sum_{i \in N_k(y)} q_i\|a - A_i\|_2^2, \tag{8}$$

Finally, we use the classifier W to classify the sparse representations A_{ts} of testing samples. We can see that when we learn the dictionary D, we minimize the combination of the reconstruction error, the classification error and the group s-parse penalty. Thus, the dictionary D is optimal for the combination to some degree. However, in recognition tasks, the final goal is to minimize the classification error only. Accordingly, the learnt dictionary D may not be the optimal dictionary for recognition tasks. We also see that when we learn the sparse representations A_{tr} of training samples and the classifier parameters W, we solve problem (7). But when we learn the sparse representations A_{ts} of testing samples, we use the model (8). In other words, the problems for computing the sparse codes in the training and the testing phases have to be different, making the models inconsistent. This leads to a problem that the new feature A_{ts} may not be optimal for the classifier W, which may limit the performance of the method.

6. *How do you manage the different complexity among the target classes, in the sense that each class (easy ones and hard ones) has the same number of atoms* *assigned. In this sense, a relevant missing reference is the following, where the authors manage dictionary coherence. Yang et al. ”Latent Dictionary Learning for Sparse Representation based Classification”, CVPR 2014.*

Reply: Actually, we can only manage the total number of the dictionary atoms. In the learning process, we don’t explicitly allot more atoms to the hard classes and less atoms to the easy ones. But we optimize the representation ability of the dictionary holistically on all data, i.e., the atoms can linearly represent samples well. Thus, such a mechanism may implicitly allot more atoms to the hard classes and less atoms to the easy ones. Note that other dictionary learning methods, such as (Jiang et al. 2013, Mairal et al. 2012 and so on), can only manage the total number of the dictionary atoms, too. We delete those inaccurate sentences in the revised manuscript, e.g., “the trained dictionary has 15 atoms for each person”, to avoid misunderstandings.

7. *In some of the tested datasets the proposed method presents a relevant increase in performance with respect to the runner-up (ex. 5%), but in other cases the increase is marginal, less than 1%. Could you comment on what conditions one can expect to take advantages of the proposed method.*

Reply: We think that the complexity of a dataset affects the performance of our method. If the data are linearly distributed, it is easy for our method to handle it. Otherwise, if the data are nonlinearly distributed, it is relatively hard to deal with the data. Also, the class number is also a factor that affects the complexity of a dataset. When the class number increases, the complexity of a dataset will increase and it is more difficult for our method to deal with the data. YaleB is an easy dataset, since (Liu et al. 2013, PAMI) point out that faces are approximately linearly distributed and its class number is relative small (it has only 38 classes). Thus, most methods can handle YaleB easily. Actually, the baselines on it are very high. Most methods can achieve 94%. In contrast, Caltech 256 is very complex, since the data are nonlinearly distributed due to its different poses, shapes and illuminations, and the class number is 257. Thus, it is quite difficult for most methods, including ours. Thus, our method only performs a little better on YaleB and Caltech 256. As for 15 Scene Categories, UCF50, HDBM51 and Caltech 101, their data are also nonlinearly distributed, but their class numbers are not very large. Thus, their complexities are less than Caltech 256. So, compared with Caltech 256, our method can handle these four datasets more easily. But other methods cannot handle them well. So our method achieves good improvements.

8. *The mathematical deduction of the optimization method is rather elaborated, maybe some part can be in an Appendix. Also some intuitions about the different steps can help to follow the method.*

Reply: Thank for your suggestion. Due to page limit, we move the mathematical deduction of the optimization method to Sections I and II of Supplementary Material.

2.3 To Reviewer #2:

1. *At a high level, I didn't find the innovation of the method to be particularly well developed or described. Specifically, the primary formulation of this paper (eq. 4) seems to be identical to the formulation considered by Guo et al in Discriminative Dictionary Learning with Pairwise Constraints (see eqs. 3-5). This paper is not cited, nor is the novelty of the current work relative to this exist-*

ing work discussed. As such, the main innovation of this work appears to be the proposed bilevel optimization strategy, about which I have several concerns (described below).

Reply: In the revised manuscript, we have discussed the difference between our method and (Guo et al. 2012) in the seventh paragraph of Section 1. We quote it below:

"Guo et al. [19] propose a pairwise constraint based discriminative dictionary learning method, named DDL-PC. They also incorporate a Laplacian term with a linear classifier to jointly learn a discriminative dictionary and a classifier. However, their model is unilevel, which cannot avoid the second drawback we mentioned above. 1) It minimizes the combination of the reconstruction error, the classification error, and the group sparse penalty, not the final goal in recognition tasks, i.e., the classification error. In this way, the classification error using the learnt dictionary may not be optimal. 2) The problems for computing the sparse codes in the training and testing phases are not consistent. That is, they use one model to learn a classifier, a dictionary, and the sparse representations of training samples, while they solve a different optimization problem to compute the sparse representation of testing samples. Finally, they adopt the learnt classifier to classify the sparse representation of testing samples. In this way, the learnt classifier is not discriminative to the sparse representation of testing samples, s-ince the classifier is learnt on the feature of training samples and the models for computing the feature of training samples and testing samples are different. We will discuss the differences between unilevel model and bilevel model in more detail in Section 4. Another difference is that in the testing phase, Guo et al. [19] solve a LASSO problem to compute the sparse codes of testing samples, while we further consider the data structure and solve the lower level optimization problem, i.e., problem (22), to compute the sparse representation of testing samples."

And in Section 4 of the new manuscript, we further discuss the differences between unilevel models, including (Guo et al. 2012), and bilevel models. When answering Question 5 of Reviewer $\#1$, we take an example to explain the differences. Please refer to Section 4 of the new manuscript and the reply to Question 5 of Reviewer $\#1$.

Besides, the optimization methods are very different. In (Guo et al. 2012), their model is unilevel and they just adopt the feature-sign search algorithm (Lee et al. 2007) to solve it. But our model is bilevel and is much more complex. Thus, we develop a novel optimization method to solve it. Compared with other optimization algorithms which can solve the bilevel models, such as the stochastic subgradient descent algorithm, our method is much more efficient. We deem that changing unilevel models to bilevel models is significant enough. For example, (Mairal et al. 2012, PAMI) is a bilevel version of (Zhang et al. 2010, CVPR) and Mairal et al. simply adopt the stochastic subgradient descent algorithm to optimize the problem.

2. *While the dictionary learning problem is understandably inherently non-convex, the optimization approach seems somewhat ad hoc with little theoretical support. For example, the application of ADM to problem (11) is well beyond the typical scope for which ADM has been developed and analyzed. First, the convergence of ADM for more than 2 variable blocks has only very recently been proven even in the convex case (which the authors do not mention) and the implementation must be done with some care to ensure convergence depending on* *the particular objective. Second (and more seriously), every ADM analysis of which I am aware is only developed for linear equality constraints. The equality constraints in the formulation the authors propose to solve (11) contain bilinearities and even quadratic terms composed with bilinearities. As such, problem (11) is well outside the scope of problems for which ADM is typically used with very little known about how ADM will perform on such a problem, and I would expect to see significant discussion of this issue and why the authors consider this approach to be justified (which they do not provide).*

Reply: When we answer Question 2 of AE, we have discussed this issue. Please refer to the reply.

3. *Given points 1 and 2, why the bilevel approach is advantageous over a more traditional optimization strategy such as alternating minimization used by Guo et al is not well established. Alternating minimization and block coordinate descent methods can be shown to be globally convergent to a critical point for the proposed objective function (see, for example, A Block Coordinate Descent Method for Regularized Multiconvex Optimization with Applications to Nonnegative Tensor Factorization and Completion by Xu and Yin and related works), and as a result are arguably more justified from a theoretical perspective. If the authors wish to establish the benefit of their method over existing techniques I would at least expect to see significant experimental results showing the superiority of bilevel optimization over something like block coordinate descent or alternating minimization. I am skeptical there will be much benefit. In fact, Guo et al performed face classification experiments on the Extended YaleB database and achieved almost identical performance (95.3% vs 95.5% accuracy).*

Reply: As for the advantages of our method when comparing with (Gao et al.

2012, ACCV), we have answered this question in your Question 1. Please refer to it.

When we answer Question 2 of AE, we have answered the reasons that we do not use other methods that are mentioned. Please refer to the reply.

Indeed, all compared methods achieve excellent recognition rates on Extended YaleB since the dataset is relatively easy. Thus, that Guo et al. obtain 95.3% cannot demonstrate that bilevel models are not beneficial to recognition rates. In our manuscript, we conducted other experiments and reported the experimental results of (Guo et al. 2012, ACCV) in Table 11. We can see that our method performs much better than DDL-PC (Guo et al. 2012, ACCV). Please refer to Section 5.6. As for why sometimes the improvements of our model over existing ones are large and sometimes small, please refer to the reply to Question 7 of Reviewer #1.

4. *How the experimental comparison is being done is somewhat unclear and some key results from well-known competing methods seem to be missing. For example, in the experiments on the YaleB database, SRC [ref 5] is listed as an unsupervised method with worse performance than the proposed method; however, in section 4 of the SRC paper the method achieves 98.1% accuracy (compared to the current methods 95.5%) in an almost identical experimental setup (in fact slightly harder as SRC used only 504 random projections compared to the current studys 540). Why is this result not listed for comparison and why is SRC listed as an unsupervised method? Further, the authors claim to compare with other state-of-the-art methods for that task (pg. 7, lines 43-45, right column) for each particular task. However, there are some methods that significantly outperform the current method which are not reported. For example, the authors of DeCAF:*

A Deep Convolutional Activation Feature for Generic Visual Recognition achieve almost 87% accuracy on Caltech 101 (compared to 75.5% in the current paper), but this is never mentioned.

Reply: When we answer Question 3 of AE, we have discussed the details of experiments and the implementations of other methods. Please refer to it.

In the original SRC paper, the authors adopt a downsampling method to downsample the images, the downsampling ratio is 1/8, while in our manuscript, following LC-KSVD (Jiang et al. 2013), we employ the random projection method to reduce the dimension of the images. About the dimension of our feature on Extended YaleB, it is also 504. We have revised it in the new manuscript and the feature for Extended YaleB is provided by (Jiang et al. 2013) and can be downloaded at http://www.umiacs.umd.edu/ zhuolin/projectlcksvd.html. Another difference is that in the original SRC paper, the authors randomly select half of the images (i.e., about 32 images per subject) to construct a dictionary and the other half for testing, while in our manuscript, for fairness, we randomly select 15 images per subject to construct a dictionary and half for testing.

SRC randomly selects the same number of images from each class to construct the dictionary. So, actually it has used the label information. Thus, we agree that SRC is a supervised method. We have fixed this error in our new manuscript.

We don't need to compare our method with DeCaf, because we are studying how to learn discriminative dictionaries for visual recognition. So it is reasonable to compare with other dictionary learning based methods or other methods with similar framework, which have been mentioned at the end of first paragraph of Section 6 ("In each specific task, we further compare with other state-of-the-art methods with similar framework for that task, such as the classic locality-constrained linear coding (LLC) method [37].”). The framework of DeCaf is very different from ours and it is not on dictionary learning. Thus, we do not compare with it.

5. *Similar to 4, how are the other methods being evaluated? Presumably, the authors must be running their own experiments from implementations of these methods as some reported results do not match those in the original papers (for example, the reported results on the Caltech 101 dataset for LC-KSVD1 and LC-KSVD2 do not match those in the original paper [ref 7, table 5]). As such, the details for how these methods are being implemented need to be provided. For example, it is a somewhat unfair comparison if the hyper-parameters for the proposed method are tuned via a large grid search while the parameters for the competing methods are only evaluated at one fixed value.*

Reply: In (Jiang et al. 2013), they reduce the dimension of spatial pyramid feature to 3000 by PCA, while we reduce the dimension of same feature to 1500 by PCA, which is presented in Section 5.3.1. When the data scale is large, some methods, such as TDDL, SRC, etc., are very time consuming. So we reduce the dimension smaller. Thus, the reported results on the Caltech 101 dataset for LC-KSVD1 and LC-KSVD2 are not the same as those in the original paper.

About the implementations of other methods, we have answered it in Question 3 of AE. Please refer to the reply.

2.4 To Reviewer #3:

1. *The paper makes many comparisons with existing dictionary learning ap-*

proaches, but gives very few details about the context of this comparison. Where these methods reimplemented by the authors, coming from some open-source software? How were chosen the parameters of all these approaches?

Reply: We have answered this question in Question 3 of AE. Please refer to the reply.

2. *The beginning of Section 3.2 contains many statements that are vague and that lack some proper justification. First, the use of the terminology "subgradient" is not appropriate. Subgradients are only defined for convex functions. Second, the claim that "its convergence speed is relatively slow" is unjustified. Stochastic gradient descent may be difficult to use because it requires tuning a step size whose optimal value is unknown in advance, but I am not aware of any other approach that is theoretically faster. The next sentence is also a bit obscure. What does mean " it is difficult to deduce the subgradient"?*

Reply: Actually, subgradient is defined not only for convex functions but also for nonconvex functions. For example, in (Attouch et al. 2013), Attouch et al. quoted the definition of "subdifferential" which is also called "subgradient". When a function is convex, the "subdifferential" can be defined in the usual way for convex functions.

We have compared our method with TDDL (Mairal et al. 2012, PAMI), which is also a bilevel model based dictionary learning method. Indeed, its model is easier than ours, since its lower level problem does not consider the data structure and it has no the Laplacian term. Actually, TDDL replaces the Laplacian term $\mathrm{tr}(ALA^T)$ in problem (4) in our manuscript with a regularization $\|A\|_F^2$. Thus, when using the stochastic subgradient descent algorithm to solve the model, the L in the implicit function (4) becomes an identity matrix and it is much easier to compute $\frac{\partial A}{\partial D}$. About the step size, we also adopt the step size choosing strategy in (Mairal et al. 2012, PAMI). In Table 9, though solving a more complex problem, we still can see that our method is much faster than TDDL, which demonstrates that using our optimization method to solve this kind of problem is faster than using the stochastic gradient descent algorithm. From the theoretical aspect, subgradient descent for nonsmooth problems is known to have a convergence rate $O(\frac{1}{\sqrt{K}})$ (Boyd et al. 2011), where K is the iteration number, while ADM for convex problems has a rate of $O(\frac{1}{K})$ in an ergodic sense (Lin et al. 2015). So ADM could be faster than subgradient descent, although our problem is nonconvex.

We have answered why it is difficult to deduce the subgradient in the part I in Question 2 of Reviewer $\#1$. Please refer to the reply.

H. Attouch, J. Bolte, B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forwardbackward splitting, and regularized GaussSeidel methods[J]. Mathematical Programming, 2013, 137(1-2): 91-129.

S. Boyd, N. Parikh, E. Chu, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends in Machine Learning, 2011, 3(1): 1-122.

Z. Lin, R. liu, H. Li. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning[J]. Machine Learning, 2015, 99(2): 287-325.

3. *The proposed approach is heuristic for several reasons, which is fine, but this should be made clear in the paper. For instance, (10) is non-convex and involves nonlinear constraints. It is thus intractable and cannot be “solved” by ADM. Applying augemented Lagrangian techniques to non-convex problems may* *be a good heuristic, but noboby knows exactly what it does in theory with respect to the original primal problem. Similarly, some of the sub-problems of the ADM algorithm are themselves intractable (updating D for instance), which requires further heuristics. This raises several questions regarding the convergence of the algorithm, which should be discussed in the paper.*

Reply: We have answered this question in the reply to Question 2 of AE. Please refer to it.

It should be pointed out that in (Zhang et al. 2013, CVPR), ADM has been successfully applied to solve an optimization problem which is a nonconvex problem with nonlinear equality constraints and has more than three blocks of variables. So their situations are similar to ours. Zhang et al. also adopt a similar strategy to update D. They report state-of-the-art experimental results.

4. *The choice of parameters in the experimental section is not clear to me. As far as I understand the paper, they seem to be optimized on the test set. Is that correct?*

Reply: Following (Jiang et al. 2013, PAMI) and (Lobel et al. 2015, PAMI), the parameters in our model are fixed for each dataset and determined by n-fold cross validation. We mention this in the second paragraph of Section 5 in the revised manuscript (" Note that the parameters in our model are fixed for each database and determined by n-fold cross validation and the detailed settings are presented in each experiment section.").

5. *The optimization technique is novel, even though the formulation builds upon those of [30,31]. Adding the Laplacian regularization can for instance not be considered a major modification: since this regularization is quadratic, the inner-problem remains a Lasso, and is thus trivial to modify [30] to use this* *regularization and compute the modified gradient.*

Reply: Actually, computing the modified gradient is not easy. Please refer to the reply to part I in Question 2 of Reviewer $\#1$. And we discuss the differences between our method and other methods, such as DDL-PC (Guo et al. 2012), TDDL (Mairal et al. 2012), (Yang et al. 2012), (Tao et al. 2014), and (Lobel et al. 2015). Please refer to the reply to Question 1 of AE.

6. *There is a link between these "task-driven" approaches and neural networks with backpropagation. Given the current popularity of neural networks, I think the readers may be interested in drawing this link, which can be found for instance in Section 4.5 of [A].*

[A]. J. Mairal, F. Bach, and J. Ponce. "Sparse Modeling for Image and Vision Processing", Foundations and Trends in Computer Graphics and Vision. 2014.

Reply: Thank for your suggestion. We have discussed this issue in our revised manuscript and the reference is also cited appropriately. But, due to page limit, we only mention this connection briefly in the third paragraph of Section 4. We quote it below:

"From the above viewpoints, there are connections between BMDDL and supervised neural networks [50], [51], [52]. Both BMDDL and supervised neural networks are multi-level recognition-driven feature learning schemes. In recognition tasks, they adopt the classification loss as their optimization goal and at each level, they use a feature extractor, such as the lower level problem (5) in BMDDL, to learn more discriminative features and feed them into the next level as input. But, since the feature extractor used in BMDDL is much more complex than that (linear mapping and nonlinear mapping) in neural networks, BMDDL can only be a network with two levels. Please refer to Supplementary Material for more details."

And we discuss this connection in more detail in Section III of Supplementary Material. We quote it below:

"There are connections between BMDDL and supervised neural networks [2], [3], [4]. Both BMDDL and supervised neural networks are task-driven feature learning schemes. In recognition tasks, minimizing the classification loss is the final task. So BMDDL and neural networks adopt it as their optimization goal. They can be formulated as a general multi-level model:

$$
\begin{aligned}
&\min_{\{W^i\},\{A^i\}} \sum_{i=1}^{n} \Phi(h_i, f(A_i^{m-1}, W^m)),\\
&\text{s.t. } A^{m-1} = \operatorname*{argmin}_A G^{m-1}(A^{m-2}, W^{m-1}, A),\\
&\qquad\qquad \text{s.t. } A^{m-2} = \operatorname*{argmin}_A G^{m-2}(A^{m-3}, W^{m-2}, A),\\
&\qquad\qquad\qquad\qquad \cdots\cdots\\
&\qquad\qquad\qquad\qquad \text{s.t. } A^{1} = \operatorname*{argmin}_A G^{1}(A^{0}, W^{1}, A),
\end{aligned}
\tag{9}
$$

where $G^i(A^{i-1}, W^i, A)$ is a feature extractor, W^i is its parameters, and A is a variable representing the extracted feature. A^0 is the input, A^i is feature of the ith level extracted by $G^i(A^{i-1}, W^i, A)$. $\Phi(h_i, f(A_i^{n-1}, W^m))$ is a classification loss function. $f(A_i^{m-1}, W^m)$ is a classifier, such as multinomial logistic regression or a linear classifier. A_i is a vector and denotes ith sample. $H = [h_1, \cdots, h_n]$ is the label of A^{n-1}. n is the number of training samples.

In BMDDL, the lower level optimization problem (5) in our paper can be regarded as the first level, which extracts group sparse feature from training samples and feeds them into the second level, i.e., a classification loss function $\Phi(\cdot)$. In this way, BMDDL is only a two-level based feature learning network. The reason why BMDDL can only stack two levels is that its feature extractor $G^i(A^{i-1}, W^i)$ is too complex. This is the very difference between BMDDL and neural networks. In BMDDL, $A^i = \operatorname{argmin}_A G^i(A^{i-1}, W^i, A)$ has no closed-form solution and we have to solve it to obtain A^i by iterative algorithms. If we stack K $(K \geq 3)$ levels in BMDDL, it will be too difficult to solve the optimization problem. In contrast, in neural networks, the new feature can be directly obtained, since we usually set $A^i = \operatorname{argmin}_A G^i(A^{i-1}, W^i, A) = \operatorname{argmin}_A \|A - \Psi(W^i A^{i-1})\|_F^2 = \Psi(W^i A^{i-1})$, where $\Psi(\cdot)$ is an activation function. Then we can use the backpropagation algorithm (based on the chain rule) to update the parameters W^i in turn. Thus, both BMDDL and neural networks are task-driven feature learning methods. But, due to the optimization difficulty, BMDDL can only be a network with two levels."