Learning Constrained Structured Spaces with Application to Multi-Graph Matching

Hedda Cohen Indelman
Technion - Israel Institute of Technology

Tamir Hazan
Technion - Israel Institute of Technology

Abstract

Multi-graph matching is a prominent structured prediction task, in which the predicted label is constrained to the space of cycle-consistent matchings. While direct loss minimization is an effective method for learning predictors over structured label spaces, it cannot be applied efficiently to the problem at hand, since executing a specialized solver across sets of matching predictions is computationally prohibitive. Moreover, there's no supervision on the ground-truth matchings over cycle-consistent prediction sets. Our key insight is to strictly enforce the matching constraints in pairwise matching predictions and softly enforce the cycle-consistency constraints by casting them as weighted loss terms, such that the severity of inconsistency with global predictions is tuned by a penalty parameter. Inspired by the classic penalty method, we prove that our method theoretically recovers the optimal multi-graph matching constrained solution. Our method's advantages are brought to light in experimental results on the popular keypoint matching task on the Pascal VOC and the Willow ObjectClass datasets.

1 INTRODUCTION

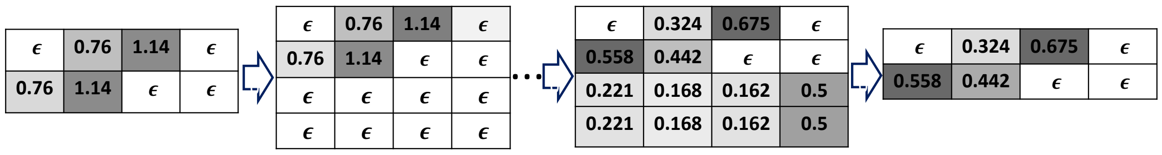
Deep graph matching is a prominent structured prediction task at the intersection between computer vision, combinatorial optimization, and machine learning. It involves predicting node correspondences between two graphs based on a parameterized node-to-node and edge-to-edge affinity.

Unfortunately, gradient methods are inefficient for learning such discrete predictions. The highest scoring structure is a piece-wise constant function of the parameters w, and its gradient w.r.t. w is zero almost everywhere. Previous research facilitated end-to-end graph matching learning by a continuous relaxation of node-to-node matching. Indeed, when graphs consist of the same set of nodes, the permutation may be continuously relaxed to a soft matching with the Sinkhorn operator (Sinkhorn, 1964). However, the set of nodes in graphs differs in practical scenarios. Unfortunately, continuously relaxing a partial permutation to a soft unbalanced matching is puzzling. This difficulty is typically addressed by generating a balanced graph matching with inlier nodes only, which is a degenerate setting. An alternative is to perform Sinkhorn normalization on a heuristically augmented rectangular scoring matrix. Yet, this practice may not result in a biproportional soft unbalanced matching, i.e., does not proportionally project onto the transportation polytope. A toy example in Figure 1 illustrates the adverse effect of this practice.

Multi-Graph Matching. Matching multiple graphs allows predicting matchings in a global fashion, as opposed to locally matching two graphs. Cycle-consistency denotes a condition in which the matching between two graphs is consistent when passed through any other graph. Prior research focused on two mechanisms for learning multi-graph matching: learning pairwise graph matching locally and enforcing cycle-consistency as a post-processing step, and learning pairwise graph matching while accounting for cycle-consistency globally. The latter is challenging since the structured label space is complexly constrained, i.e., it's the set of matchings that are cycle-consistent globally.

While the direct loss minimization technique allows learning predictors over structured label spaces, it requires an efficient prediction solver to estimate the gradient of a given loss. We extend the applicability of the direct loss minimization technique to complexly constrained structured spaces. To that end, we decompose matching constraints from cycle-consistency constraints, such that cycle-consistency constraints are softly enforced as loss functions and pairwise matchings are strictly enforced in predictions. We prove that this extension theoretically recovers the optimal multi-graph matching solution. Two-graph and hypergraph matching fit naturally in our method by limiting the structured label space to the set of matchings. Further, our method applies to balanced and unbalanced matching naturally and does

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.



(a) Add dummy rows and initialize them to a small epsilon

(b) Repeat normalizing rows and columns

(c) Remove dummy elements

Figure 1: A toy example illustrating the potential effect of Sinkhorn normalization with dummy elements. Consider matching 2 nodes to 4 nodes by normalizing a 2×4 node-to-node affinity matrix. (a) Add dummy rows and initialize their elements to a small ϵ. (b) Normalize rows and columns repeatedly until convergence. (c) Remove dummy elements and produce a soft unbalanced matching. However, this soft unbalanced matching is not biproprtional w.r.t. the input matrix.

not require continuous relaxation of the scoring function.

In summary, our contributions are the following:

1. Our method for end-to-end learning of multi-graph matching accounts for pairwise missing correspondences and allows for minimizing the structured loss without relaxing the matching prediction.
2. We extend the direct loss minimization to settings in which the prediction solvers are computationally inefficient, as in the setting of multi-graph matchings. Thus, our method allows learning cycle-consistent matchings, while not using a cycle-consistent matching solver.
3. This extension is proved to theoretically recover the constrained multi-graph matching optimal solution.
4. We demonstrate the effectiveness of our method in various balanced and unbalanced matching tasks (two-graph, hypergraph, and multi-graph matching).

2 RELATED WORK

Two-Graph Matching. Most graph matching methods aim at finding correspondences between two graphs (Leordeanu et al., 2012; Gold and Rangarajan, 1996; Caetano et al., 2007; Egozi et al., 2013). A line of research suggested graph matching models based on the quadratic assignment problem (QAP) formulation (Zhou and la Torre, 2016; Nowak et al., 2018). A similar formulation in Cho et al. (2010) expressed graph matching as an integer quadratic program and proposed a solver based on a random walk. Formulating graph matching as vertex classification on the association graph was suggested in Leordeanu and Hebert (2005); Lee et al. (2011). Solvers based on Lagrangian decomposition (Swoboda et al., 2017; Torresani et al., 2013; Zhang et al., 2016) were also suggested.

Zanfir and Sminchisescu (2018) introduced a novel neural graph matching solver based on spectral methods and deep hierarchical features learning with a quadratic objective. Many end-to-end differentiable models (Wang et al., 2019b; Zhang and Lee, 2019; Fey et al., 2020; Wang et al., 2020b; Yu et al., 2020; Chen et al., 2021; Jiang et al., 2022) rely on relaxing the node-to-node affinity with Sinkhorn normalization. In unbalanced matching, Sinkhorn normalization is often performed on dummy elements-padded affinity matrices. Our method differs as it allows minimizing the structured loss without relaxing the sought-after discrete prediction. Generalizing second-order affinity to higher-order affinity allowed hypergraph matching (Zass and Shashua, 2008; Lee et al., 2011; Feng et al., 2019). The method of blackbox differentiation (Pogančić et al., 2020) was recently applied to neural graph matching (Rol'inek et al., 2020), based on graph matching solvers (Swoboda et al., 2019). Interestingly, while the blackbox differentiation method's gradients are of a surrogate linearized loss, direct loss minimization estimates the gradients of the expected structured prediction loss itself.

Multi-Graph Matching. Cycle-consistency is often enforced as a post-synchronization step given pairwise matchings (Chen et al., 2014; Pachauri et al., 2013; Maset et al., 2017; Zhou et al., 2015; Rol'inek et al., 2020). Wang et al. (2021) allow multi-graph end-to-end matching accounting for cycle-consistency by a method of spectral fusion. However, the proposed spectral fusion method assumes all graphs are of equal size and is inapplicable to multi-graph unbalanced matching. An end-to-end multi-graph matching scheme based on soft pairwise matching is proposed by Wang et al. (2020a). Geometrical and unsupervised cycle-consistency-based loss was introduced by Phillips and Daniilidis (2019) to solve a balanced multi-graph matching problem. Our method differs from peer methods as cycle-consistency is enforced during learning of pairwise matchings, while accounting for discrete matching between multiple sets of nodes of potentially different sizes.

3 BACKGROUND

Discriminative learning from labeled data amounts to learning parameters w of a predictor $y_w : \mathcal{X} \rightarrow \mathcal{Y}$ from a data point x to its label y. The predictor $y_w(x)$ relies on a parameterized scoring function $s_w(x, y)$ and its parameters are learned to minimize a loss function $\ell(\cdot,\cdot)$ over the training data. In these settings, the predictor is the label with the highest score value $y^*_w(x) = \arg\max_{\hat{y}\in\mathcal{Y}} s_w(x,\hat{y})$, where $\mathcal{Y}$ is the set of admissible structures. In the rest of the paper we use $s(x,y)$ instead of $s_w(x,y)$ and y^* instead of $y^*_w(x)$ to simplify the notation.

3.1 Graph Matching

Let $G^i = (V^i, E^i)$ and $G^j = (V^j, E^j)$ be two graphs and denote by n_i the number of nodes in G^i, i.e., $n_i = |V^i|$, and similarly for n_j. The pair of graphs G^i and G^j form a data instance x^{ij}. Its corresponding label is a permutation matrix $y(x^{ij}) \in \{0,1\}^{n_i \times n_j}$ representing the node matching between graphs G^i, G^j. As such, $y(x^{ij})_{lm} = 1$ if node $l \in V^i$ is matched to node $m \in V^j$, and $y(x^{ij})_{lm} = 0$ otherwise.

Unbalanced Matching is formed when the set of nodes differs such that some nodes from V^i and V^j may not be matched. In this case, y is a partial permutation matrix. It is often a more realistic setting for matching problems, e.g., in machine vision the number of keypoints in each image is rarely equal, due to occlusions or different points of view. The set of pairwise unbalanced matchings is defined as:

$$\begin{aligned}\mathcal{M}^{ij} = \{y(x^{ij}) \in \{0,1\}^{n_i \times n_j} : y(x^{ij})\mathbf{1}_{n_j} \le \mathbf{1}_{n_i}, \\ y(x^{ij})^T \mathbf{1}_{n_i} \le \mathbf{1}_{n_j}\}.\end{aligned} \tag{1}$$

Balanced Matching is formed when the set of nodes V^i is equal to the set V^j. Balanced matchings correspond to full permutations, in which case Equation (1) holds with equality. Balanced matching carries important information in its label space since there is a one-to-one correspondence between V^i and V^j. This implicit bias typically allows deep nets to excel at this task.

Multi-Graph Matching imposes cycle-consistency constraints over three pairwise graph matchings or more. In multi-graph unbalanced matching, graphs may contain outlier nodes, hence cycle-consistency among a set of graphs G^i, G^j, G^k pairwise matchings suggests:

$$\mathcal{C}^{ij} = \left\{ y(x^{ij}) : y(x^{ij}) \ge y(x^{ik}) y(x^{kj}) \quad ,\forall k \right\}. \tag{2}$$

$y(x^{ik})y(x^{kj}) \in \{0,1\}^{n_i \times n_j}$ is the matrix multiplication between $y(x^{ik})$ and $y(x^{kj})$, representing matching between graphs G^i and G^j through graph G^k. We note that Equation (2) holds with equality in balanced matching, in which case the correspondence predictions are full permutations. However, we make no assumptions on the dimensions of the pairwise matchings.

In the following, we omit the superscript ij when it is obvious from context. The optimization function for learning multi-graph matching considers the predictor $y^*_w(x)$ over the space of cycle consistent matchings $\mathcal{M} \cap \mathcal{C}$ for each training sample $(x,y) \in \mathcal{S}$. The optimization program takes the form $\min_w \sum_{(x,y)\in\mathcal{S}} \ell(y^*_w(x), y)$ subject to $y^*_w(x) \in \mathcal{M} \cap \mathcal{C}$ for every $x \in \mathcal{S}$. Unfortunately, since $y^*_w(x)$ is not continuous everywhere and its derivative is zero almost everywhere, this function cannot be minimized by gradient descent without further assumptions.

3.2 Direct Loss Minimization

The direct loss minimization technique (Hazan et al., 2010; Keshet et al., 2011; Song et al., 2016; Cohen Indelman and Hazan, 2021) allows minimizing the loss of a predicted label, which corresponds to the structure that maximizes a score function. Although the maximal argument is not differentiable with respect to the parameters of the scoring function, the direct loss minimization framework ensures differentiability by adding a random perturbation to the scoring function $s(x,y)$ in the label space. In this setting, a random perturbation $\gamma(y)$ is added to each possible label. With this we generate random predictions:

$$\begin{aligned} y^* &= \arg\max_{\hat{y}\in\mathcal{Y}} \{s(x,\hat{y}) + \gamma(\hat{y})\}. && (3)\\ y^*(\epsilon) &= \arg\max_{\hat{y}\in Y} \{s(x,\hat{y}) + \gamma(\hat{y}) - \epsilon\ell(\hat{y},y)\}. && (4)\end{aligned}$$

The gradient of the direct loss minimization technique is composed of gradients of the scoring function at these random predictions: $\nabla_w \mathbb{E}_\gamma[\ell(y^*,y)] =$

$$\lim_{\epsilon\to 0} \frac{1}{\epsilon} \left(\mathbb{E}_\gamma\left[\nabla s_w(x,y^*) - \nabla s_w(x,y^*(\epsilon))\right]\right). \tag{5}$$

The gradient $\nabla s_w(x,y^*)$ reduces the scoring function at the prediction y^*, while the gradient $-\nabla s_w(x,y^*(\epsilon))$ increases the scoring function at the minimal loss prediction $y^*(\epsilon)$ as described in Equation (4). This notation appears implicitly in previous works, and is considered as "toward-better" gradient step in Hazan et al. (2010).

Efficiently using the direct loss minimization technique requires a black-box structured predictor to estimate the gradient of a given loss, i.e., to solve the optimization programs y^* and $y^*(\epsilon)$.

4 LEARNING CONSTRAINED DISCRETE LABEL SPACE

The direct loss minimization allows learning a model that best fits a predicted label y^* from any discrete space $\mathcal{Y}$. Thus, it holds promise for learning multi-graph matchings, for which the label space is the set of all cycle-consistent matchings $\mathcal{Y} = \mathcal{M} \cap \mathcal{C}$. While direct loss minimization can be applied to two-graph matching efficiently, it cannot be applied efficiently to multi-graph matching since executing such a specialized solver across sets of cycle-consistent matching predictions during training is computationally prohibitive. Moreover, the ground-truth labels only hold for pairwise matchings $\mathcal{Y} = \mathcal{M}$, and not for cycle-consistent sets of matchings. In the following, we present our method for efficiently extending the direct loss minimization to settings in which the black-box solvers are computationally inefficient, as in the setting of multi-graph matchings. As such, our method allows learning cycle-consistent matching, while not using a cycle-consistent matching solver.

The predicted label y^*, described in Equation (3), is the partial permutation that maximizes the randomly perturbed node-to-node correspondence scoring function $s_w(x, y)$. The scoring function assigns a real-valued number to each match, i.e., $s(x, y) = (s_{11}(x, y_{11}), ..., s_{n_i n_j}(x, y_{n_i n_j}))$. In this setting, a low-dimensional random perturbation is added to the scoring of each possible match, i.e., $\sum_{l=1}^{n_i} \sum_{m=1}^{n_j} (s_{lm}(x, \hat{y}_{lm}) + \gamma_{lm}(\hat{y}_{lm}))$. Each $\gamma_{lm}(y_{lm})$ is an independent random variable that follows the zero-mean Gumbel distribution, denoted $\mathcal{G}$. The scoring function need not be normalized, in contrast to the continuous relaxation-based approaches. Oftentimes (Wang et al., 2019a, 2021, 2020c,b), learning graph matchings uses the binary cross entropy loss (BCE) to measure the goodness of fit of a training-pair (x, y) and the predicted matching y^*. The BCE loss easily applies to matchings, where $y \in \{0, 1\}$. However, due to the discreteness of the prediction y^*, false negatives and false positives would be penalized equally. To generate a "towards-better" gradient step we adopt a weighted BCE loss ($\alpha > 1$). Please refer to Appendix A.1 for further details. As such, in our setting, the BCE loss function is

$$\ell^{bce}(y^*(x), y) = -\alpha y \log(y^*(x)) - (1-y)\log(1-y^*(x)). \tag{6}$$

In multi-graph matching, the predicted label $y^*(x)$ is constrained to the space of cycle-consistent matchings $\mathcal{Y} = \mathcal{M} \cap \mathcal{C}$. The optimization problem at hand is

$$\begin{aligned} w^* \in \arg\min_w & \sum_{(x,y)\in\mathcal{S}} \mathbb{E}_\gamma[\ell^{bce}(y^*(x), y)] \\ \text{s.t. } \forall x \in \mathcal{S} \quad & y^*(x) = \arg\max_{\hat{y}\in\mathcal{M}} \{s(x, \hat{y}) + \gamma(\hat{y}))\} \\ \forall x \in \mathcal{S} \quad & y^*(x) \in \mathcal{C}. \end{aligned} \tag{7}$$

The sets $\mathcal{M}$ and $\mathcal{C}$ involve the random variables γ and these conditions hold for every γ, except for a set of measure zero of γ. Formally, these conditions hold for almost every γ. We note that this condition is not affecting the objective, that assigns a value of zero to subsets of measure zero with respect to γ due to its expectation.

Optimizing this problem may not be practical without further assumptions, as predicting the label $y^*(x)$ over the space of cycle-consistent matchings $\mathcal{M} \cap \mathcal{C}$ requires calling a multi-graph matching solver for each training sample $(x, y) \in \mathcal{S}$ in every iteration.

To solve the optimization program in Equation (7) we recall that the cycle-consistency condition, defined in Equation (2), involves inequalities of the form $y(x^{ik})y(x^{kj}) \leq y(x^{ij})$. An inequality condition naturally translates to the penalty function $\max\{0, y(x^{ik})y(x^{kj}) - y(x^{ij})\}$. This penalty function is in fact a loss function: whenever the condition holds the loss function is zero, otherwise it has the value of one. The cycle-consistency loss of the predicted matching $y^*(x^{ij})$ between G^i and G^j takes the form:

$$\begin{aligned} \ell^{cycle}_{ij,k}(y^*(x^{ij}), y^*(x^{ik}), y^*(x^{kj})) &= \\ max\{0, y^*(x^{ik})y^*(x^{kj}) &- y^*(x^{ij})\}. \end{aligned} \tag{8}$$

We use the soft penalty function $\ell^{cycle}_{ij,k}$ instead of a hard constraint during optimization, and note that it decomposes to the scoring function dimension of x^{ij}. An example is illustrated in Figure 2.

Thus, the loss of a predicted matching between graphs G^i and G^j is a weighted sum between its BCE loss and cycle-inconsistency loss functions, controlled by a penalty parameter n:

$$\begin{aligned} \ell^{mgm}(y^*(x^{ij})) &= \ell^{bce}(y(x^{ij}), y^*(x^{ij})) \\ &+ n\sum_{i,j,k} \ell^{cyc}_{ij,k}(y^*(x^{ij}), y^*(x^{ik}), y^*(x^{kj}). \end{aligned} \tag{9}$$

We summarize our method in Algorithm 1.

To guarantee that the final solution satisfies the constraints in $\mathcal{C}$, we gradually increase the penalty, till convergence. This is summarized in the following theorem:

Theorem 1. *Consider the following optimization program:*

$$\begin{aligned} w_n \in \arg\min_w & \sum_{(x,y)\in\mathcal{S}} \mathbb{E}_\gamma[\ell^{bce}(y^*(x), y)] + \\ & n\sum_{i,j,k} \mathbb{E}_\gamma\left[\ell^{cycle}_{ij,k}\left(y^*(x^{ij}), y^*(x^{ik}), y^*(x^{kj})\right)\right] \\ s.t.\ \forall x \in \mathcal{S} \quad & y^*(x) = \arg\max_{\hat{y}\in\mathcal{M}} \{s(x, \hat{y}) + \gamma(\hat{y})\} \end{aligned} \tag{10}$$

Assume $s(x, y)$ is differentiable and w belong to a bounded set. If $n = 1, 2, ...$ is an increasing sequence of natural numbers then a subsequence of w_n has a limit point w^* which is the optimal argument of the optimization program in Equation (7), while w^* satisfies its constraints for almost every γ.

Proof. Let $f(w) = \sum_{(x,y)\in S} \mathbb{E}_\gamma[\ell^{bce}(y^*(x), y)]$ be the perturbed training loss over S and $g(w) = \sum_{i,j,k} \mathbb{E}_\gamma\left[\ell^{cycle}_{ij,k}\left(y^*(x^{ij}), y^*(x^{ik}), y^*(x^{kj})\right)\right]$ be the cycle inconsistency penalty. Then the theorem asserts that $w_n = \arg\min_w f(w) + ng(w)$ converge to a solution w^* of Equation (7).

The proof follows three steps: (i) the values of $f(w_n) + ng(w_n)$ are bounded and hence the values of $g(w_n)$ are

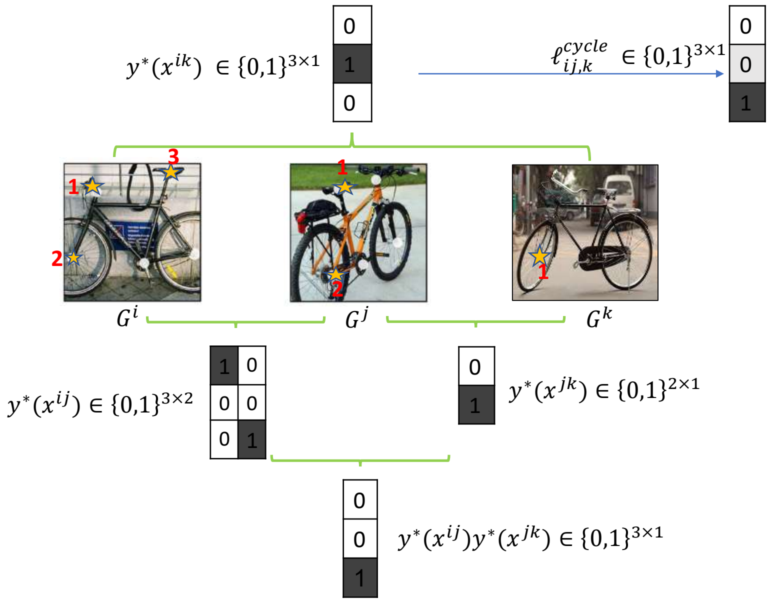

$y^*(x^{ik}) \in \{0,1\}^{3\times1}$
0
1
0
$\ell^{cycle}_{ij,k} \in \{0,1\}^{3\times1}$
0
0
1
1
3
2
1
2
1
G^i
G^j
G^k
$y^*(x^{ij}) \in \{0,1\}^{3\times2}$
1 0
0 0
0 1
0
1
$y^*(x^{jk}) \in \{0,1\}^{2\times1}$
0
0
1
$y^*(x^{ij})y^*(x^{jk}) \in \{0,1\}^{3\times1}$

Figure 2: A toy illustration of our cycle-consistency loss for 3-graph matching. Pairwise matchings $y^*(x^{ij}), y^*(x^{ik}), y^*(x^{jk})$ are predicted (Equation 3). In this setting, each pairwise matching will have one cycle-consistency loss term based on three pairwise matchings. Depicted is the cycle-inconsistency loss (Equation 8) of the predicted pairwise matching $y^*(x^{ik})$ between G^i and G^k w.r.t. passing the matching through image G^j. Multi-graph matching with larger sets allows robustness to such local pairwise matching noise.

converging to zero while $n \to \infty$; (ii) w_n converge to a limit point $\tilde{w}$ that satisfies $g(\tilde{w}) = 0$; (iii) $\tilde{w}$ is a solution to the optimization program of Equation (7).

We assume that $f(w_n), g(w_n) \geq 0$, which is a reasonable assumption for loss functions, and note that $\min_w f(w) + ng(w) = f(w_n) + ng(w_n) \leq f(w^*) + ng(w^*)$. Since w^* satisfies the conditions in Equation (7) then $g(w^*) = 0$ and consequently $0 \leq f(w_n) + ng(w_n) \leq f(w^*)$. We prove that $\lim_{n\to\infty} g(w_n) = 0$ by contradiction: assume $g(w_n)$ has a subsequence that converges to $\alpha > 0$ then $\infty > f(w^*) \geq \limsup_{n\to\infty} f(w_n) + ng(w_n) \geq \limsup_{n\to\infty} ng(w_n)$ since $f(w_n) \geq 0$ for every n. Since we assume by contradiction that $\limsup_{n\to\infty} g(w_n) \geq \alpha$ then $\limsup_{n\to\infty} ng(w_n) = \infty$ which reaches a contradiction.

Next, we show that w_n has a limit point $\tilde{w}$ that satisfy $g(\tilde{w}) = 0$. The function $g(w)$ is a continuous function (cf. Cohen Indelman and Hazan (2021), Corollary 1) therefore w_n belong to the closed set $\{w : g(w) \leq f(w^*)\}$. Since the set of all w is bounded by assumption then $\{w : g(w) \leq f(w^*)\}$ is a compact set and w_n has a limit point $\tilde{w}$. Since $g(w)$ is continuous then $g(\tilde{w}) = \lim_{n\to\infty} g(w_n) = 0$. Finally, we concluded $f(w_n) + ng(w_n) \leq f(w^*)$ for every n, hence $f(\tilde{w}) \leq f(w^*)$. However, since $g(\tilde{w}) = 0$ then $\tilde{w}$ satisfies the conditions of $\mathcal{C}$ for almost every γ and since w^* is the minimizer of this program, there holds $f(w^*) \leq f(\tilde{w})$ and consequently $f(w^*) = f(\tilde{w})$. □

```
Algorithm 1 Multi-graph matching
For each set of graphs:
  1. For each pair of graphs $G^i, G^j$:
      (a) Perturb the associated (unnormalized) scoring function with low-dimensional Gumbel i.i.d. random variables $\gamma(y(x^{ij})) = \sum_{l=1}^{n_i} \sum_{m=1}^{n_j} \gamma_{lm}(y_{lm})$.
      (b) Predict a matching $y^*(x^{ij})$ (Equation 3).
      (c) Compute the per-element BCE loss (Equation 6).
  2. Compute the cycle-inconsistency loss of each pairwise prediction $y^*(x^{ij})$ (Equation 8).
  3. Construct the multi-graph matching loss of $y^*(x^{ij})$ (Equation 9).
  4. Predict a loss-augmented matching $y^*(x^{ij})(\epsilon)$ (Equation 4).
The expected loss gradient (Equation 5) is
    $$\lim_{\epsilon\to0} \frac{1}{\epsilon}\Big(\mathbb{E}_{\gamma\sim\mathcal{G}}[\nabla_w s_w(x, y^*(\epsilon)) - \nabla_w s_w(x, y^*)]\Big).$$
```

We further note that as n increases, the cycle-consistency tends to dominate the loss-augmented prediction $y^*(\epsilon)$, provided that a cycle exists. Consequentially, for negative ϵ, the gradient step (Equation 5) increases the score of structures corresponding to the most cycle-consistent prediction. Thus, for a high enough penalty parameter, predictions are matchings that satisfy the global cycle-consistency constraints.

Convergence Properties. For each cycle-consistency constraint weighting factor n, gradient descent methods can be assumed to converge to a corresponding stationary point $\bar{w}_n$, under mild assumptions. Indeed, proving the theoretical convergence of gradient descent methods to a global minimum requires much stronger assumptions. In practice, we experienced that the optimization scheme converged well to almost zero loss and 100% accuracy (Figure 5).

Two-Graph Matching fits naturally in our method by limiting the structured label space to the set of matchings and hence setting the cycle-inconsistency penalty parameter n to zero. Similarly, generalizing the second-order affinity to higher-order affinity allows hypergraph matching.

5 EXPERIMENTS

We evaluate our method on the widely adopted datasets for keypoint matching Pascal VOC with Berkeley annotations and Willow ObjectClass. While the latter translates to balanced graph matching, the former is a natural unbalanced matching challenge as images have a varying number of keypoints.

Our architecture is based on our strongest competitor NGM-v2 (Wang et al., 2021), with key differences. Sinkhorn's normalization is not performed on the node-to-node affinity matrix prior to the matching prediction in all unbalanced matching settings. To accomplish this we change the vertex classification of the NGM-v2 architecture such that matching is predicted (y^*_w) on the unnormalized randomly perturbed scoring function $s_w(x, y)$. Our method allows estimating the gradient of the expected structured loss (Figure 3). In all experiments, a simple channel-wise normalization layer was added.

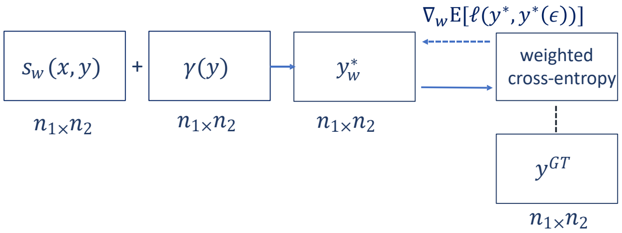


Figure 3: Illustration of our vertex classification prediction, performed on the unnormalized randomly perturbed scoring function $s_w(x, y)$, the associated loss and its derivative (Equation 5).

Multi-graph matching is based on our loss (Equation 9) and gradient step (Equation 5). The cycle-inconsistency parameter n increases with every epoch, such that when a cycle exists, it dominates the loss-augmented prediction. Our multi-graph matching shares the architecture of our two-graph matching pipeline. By that, we depart from the spectral fusion technique of Wang et al. (2021).

We compare prominent neural graph matching methods based on continuous matching relaxation and the blackbox differentiation method whenever applicable. We report the test set's average matching prediction accuracy (recall) and a per-class average matching accuracy. In the unbalanced matching experiment, we also report F_1 score, the harmonic mean between precision and recall.

Please refer to Appendix A.2 for considerations of setting the loss-augmentation parameter ϵ and to Appendix A.3 for further experiments details. Our code is publicly [1] available.

5.1 Unbalanced Matching

The Pascal VOC dataset with Berkeley annotations (Everingham et al., 2010; Bourdev and Malik, 2009) offers an unbalanced matching challenge. This natural image dataset comprises 20 instance classes with keypoint annotations. Objects vary in scale, pose and illumination, and the number of keypoints in each image varies from 6 to 23.

Keypoint Filtering. Following Wang et al. (2021); Rol'inek et al. (2020); Wang et al. (2019a), poorly annotated images and keypoints annotated as 'truncated', 'occluded', and 'difficult' are filtered. We *neither* filter keypoints to reach an intersection *nor* an inclusion between pairs of sampled images. The average imbalance between the number of effective keypoints is measured by sampling $6,385$ pairs of images. The average imbalance is 1.4, and the highest imbalance is 1.96 of class 'sheep'. Further details are in Appendix A.3.

Methods based on continuous matching relaxation were evaluated by performing Sinkhorn's normalization on a square node-to-node affinity matrix produced by adding dummy elements, as suggested in the baseline papers Wang et al. (2020a); Yu et al. (2020); Wang et al. (2021). We use ThinkMatch project's unified experiment setting and peer graph matching implementations.

Multi-Graph Unbalanced Matching. Instead of sampling pairs of images, we sample sets of five images. In such a setting, the loss for each pair of images comprises the BCE as well as three cycle-consistency loss terms. The model is trained with our gradient step (Equation 5) accounting for the corresponding multi-graph matching loss (Equation 9). For a set of $G_n \geq 3$ graphs, each pairwise matching forms $G_n - 2$ cycle-consistency loss functions.

Our scheme of tuning the penalty parameter n reflects the ratio of the BCE loss to set-wise multi-graph matching constraints (Figure 4). As n grows, fewer cycle-consistency losses in consensus are needed to dominate the multi-graph

[1] github.com/HeddaCohenIndelman/Learning-Constrained-Structured-Spaces-with-Application-to-Multi-Graph-Matching

Table 1: Training and inference time of multi-graph matching methods on the Pascal VOC dataset.

Method	Training average samples /s ↑	Inference total time (mins) ↓
GA-MGM	0.4	145
BB-GM-Multi	3.43	204
Ours	3.91	55

matching loss.

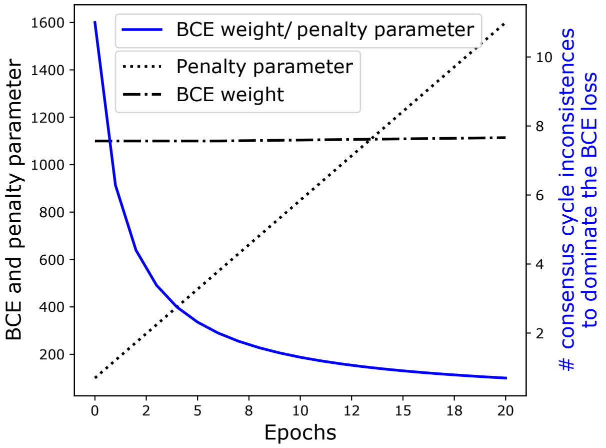


Figure 4: The cycle-inconsistency penalty parameter n tuning in the multi-graph unbalanced matching experiment. As n grows, fewer cycle-consistency losses in consensus are needed to dominate the multi-graph matching loss. Consequentially, the gradient step increases the scores of structures corresponding to the most cycle-consistent prediction.

We compare to peer methods whenever possible. Thus, we compare to BB-GM-Multi (Rol'inek et al., 2020) by using the authors' published code. We also compare to GA-MGM (Wang et al., 2020a). The method of Wang et al. (2021) is inapplicable to the unbalanced multi-graph matching setting. Our method's average test set accuracy is only slightly lower than BB-GM-Multi (Table 3a). A per-class comparison of average accuracy shows that our method (Ours) outperformed BB-GM-Multi in nine classes (Table 2a). Note that our method is learning-based while BB-GM-Multi makes use of a designated multi-graph matching solver during inference, which is also reflected in its long inference time (Table 1).

Two-Graph Unbalanced Matching. We compare to NGM-v2 (Wang et al., 2021), CIE-H (Yu et al., 2020) and qc-DGM (Gao et al., 2021), as well as to BB-GM (Rol'inek et al., 2020). As mentioned in Wang et al. (2021), BB-GM has a slightly favorable setting by filtering keypoints outside of the bounding box. For a fair comparison, we use the method's implementation in the ThinkMatch project (referred to as BB-GM*).

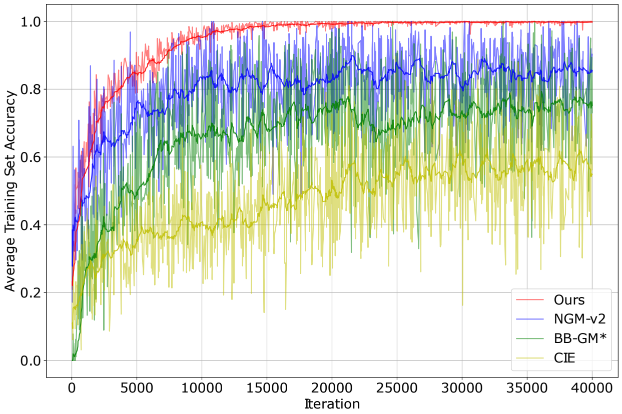


Figure 5: A comparison of average training set accuracy over learning iterations of the graph unbalanced matching experiment on Pascal VOC.

Results in Table 2 validate our method's notable advantage in predicting unbalanced node matching. The average test set accuracy and F_1 score results show that our method outperforms peer methods in both metrics by a considerable margin (Table 3b). A per-class average accuracy comparison in Table 2b further reveals that our method outperformed reference methods in 14 classes. In 6 classes the accuracy gain is more than 5% over the second performing method. Figure 5 compares the average training set accuracy over learning iterations. Note that our method is the most stable and reaches the highest average training set accuracy, compared to peer methods.

Hypergraph Unbalanced Matching. We compare to NHGM-v2 (Wang et al., 2021). Both models are pre-trained on the corresponding two-graph matching problem. Our method achieves 6.8% increase in average accuracy and almost 6% increase in F_1 score (Table 3c). A per-class average accuracy comparison reveals that our method outperforms the baseline in all but two classes (Table 2c).

5.2 Balanced Matching

The Willow ObjectClass dataset (Cho et al., 2013) offers a natural balanced matching challenge. This natural image dataset comprises five classes with keypoint annotations. Each class contains at least 40 images, and all class instances share the same 10 keypoints with no outliers.

We compare to GMN (Zanfir and Sminchisescu, 2018), PCA-GM Wang et al. (2019a), IPCA-GM (Wang et al., 2020b), BB-GM (Rol'inek et al., 2020), NGM (Wang et al., 2019b) and NGM-v2 (Wang et al., 2021). Our method's results are comparable to or slightly better than recent peer methods. This set of experiments shows that our balanced

Table 2: Graph unbalanced matching average accuracy per class on the Pascal VOC dataset. For fair comparison, we use the same keypoint filtering across all methods (preserving outlier keypoints in both images). Best results are in bold.

(a) Multi-graph unbalanced matching accua racy (%) per class.

Class	✈	🚲	🐦	🚢	🍾	🚌	🚗	🐱	🪑	🐄	🪑	🐕	🐎	🏍	🚶	🪴	🐑	🛋	🚆	🖥
GA-MGM	23.1	34.6	26.6	33.1	68.5	46.8	30.5	37.3	29.2	28.4	41.3	29.6	28.2	30.0	22.3	65.2	28.4	29.4	64.2	**85.6**
BB-GM-Multi	44.7	**74.8**	62.4	49.9	85.2	68.9	58.9	**68.1**	42.3	**66.2**	48.2	**67.8**	**66.1**	**73.1**	**47.5**	**97.2**	**59.9**	42.8	**83.3**	83.5
Ours	**52.1**	66.5	**64.4**	**50.9**	**87.0**	**70.5**	**70.3**	62.1	**44.5**	63.5	**55.8**	63.9	60.5	64.2	47.3	94.9	58.2	**42.9**	81.8	83.5

(b) Two-graph unbalanced matching accuracy (%) per class.

Class	✈	🚲	🐦	🚢	🍾	🚌	🚗	🐱	🪑	🐄	🪑	🐕	🐎	🏍	🚶	🪴	🐑	🛋	🚆	🖥
CIE	34.0	59.1	47.0	33.7	81.5	54.1	31.9	47.1	28.3	46.2	52.7	45.0	45.4	50.0	29.3	82.9	39.2	35.4	56.1	76.5
qc-DGN	30.9	59.8	48.8	40.5	79.6	51.7	32.5	55.8	27.5	52.1	48.0	50.7	57.3	60.3	28.1	90.8	51.0	35.5	71.5	79.9
BB-GM*	42.9	64.3	54.9	48.0	84.7	65.9	45.9	59.9	40.1	**63.6**	**49.1**	60.2	58.7	62.3	39.0	92.7	56.0	**40.6**	75.9	**86.4**
NGM-v2	45.4	**68.4**	54.3	48.8	**86.8**	64.6	55.1	57.0	40.8	57.7	44.8	55.9	54.7	55.9	43.4	89.7	47.7	30.8	70.2	77.1
Ours	**50.9**	64.4	**62.6**	**51.6**	85.0	**71.9**	**73.3**	**64.7**	**44.9**	62.9	48.8	**63.0**	**60.8**	**64.3**	**50.6**	**93.9**	**59.8**	39.5	**84.7**	84.7

(c) Two-hypergraph unbalanced matching average accuracy (%) per class.

Class	✈	🚲	🐦	🚢	🍾	🚌	🚗	🐱	🪑	🐄	🪑	🐕	🐎	🏍	🚶	🪴	🐑	🛋	🚆	🖥
NGM-v2	44.7	**67.1**	54.7	47.9	86.6	67.5	59.4	57.1	**42.3**	57.4	39.7	55.7	53.9	57.0	44.7	91.7	48.9	35.5	72.4	76.6
Ours	**50.8**	65.8	**60.4**	**54.8**	**86.8**	**72.2**	**67.7**	**63.8**	41.2	**64.0**	**60.3**	**62.9**	**62.6**	**64.6**	**51.3**	**93.9**	**59.1**	**44.4**	**82.7**	**82.5**

Table 3: Graph unbalanced matching average accuracy and F_1 score on the Pascal VOC dataset. Best results in bold.

(a) Multi-graph unbalanced matching.

Method	**Accuracy**	F_1 **score**
GA-MGM	39.13%	0.384
BB-GM-Multi	**64.54%**	**0.624**
Ours	64.25%	0.575

(b) Two-graph unbalanced matching.

Method	**Accuracy**	F_1 **score**
CIE	48.77%	0.459
qc-DGM	-	0.526
BB-GM*	59.55%	0.573
NGM-v2	57.46%	0.537
Ours	**64.11%**	**0.597**

(c) Two-hypergraph unbalanced matching.

Method	**Accuracy**	F_1 **score**
NHGM-v2	58.04%	0.541
Ours	**64.58%**	**0.601**

matching approach is effective and that our multi-graph matching improved average matching accuracy compared to the two-graph matching method.

Multi-Graph Balanced Matching. In this experiment, we sample sets of five images from the same class. Thus, the loss of each pair of images comprises its BCE loss and three cycle-consistency loss terms. Results are summarized in Table 4a.

Two-Graph Balanced Matching. This widely adopted setting allows a comparison to many peer methods. Results show that our method achieved the highest average test set accuracy and the highest per-class average accuracy in four out of five classes (Table 4b).

Two-Hypergraph Balanced Matching. Models are pre-trained on the corresponding two-graph matching problem. Our method's result is comparable to NHGM-v2 (Table 4c).

Table 4: Graph balanced matching average accuracy on the Willow ObjectClass dataset. Best results are in bold.

(a) Multi-graph balanced matching.

	🚗	🦆	🙂	🏍	🍾	**mean**
NMGM	78.5%	92.1%	**100%**	78.7%	94.8%	88.8%
NMGM-v2	97.6%	94.5%	**100%**	**100%**	99.0%	98.2%
Ours	**98.3%**	**98.2%**	**100%**	96.9%	**99.9%**	**98.7%**

(b) Two-graph balanced matching.

	🚗	🦆	🙂	🏍	🍾	**mean**
GMN	67.9%	76.7%	99.8%	69.2%	83.1%	79.3%
PCA-GM	87.6%	83.6%	**100%**	77.6%	88.4%	87.4%
IPCA-GM	90.4%	88.6%	**100%**	83.0%	88.3%	90.1%
BB-GM	96.8%	89.9%	**100%**	**99.8%**	99.4%	97.2%
qc-DGM	98.0%	92.8%	**100%**	98.8%	99.0%	**97.7%**
NGM	84.2%	77.6%	99.4%	76.8%	88.3%	85.3%
NGM-v2	97.4%	93.4%	**100%**	98.6%	98.3%	97.5%
Ours	**98.2%**	**94.3%**	**100%**	96.2%	**99.2%**	97.6%

(c) Two-hypergraph balanced matching.

	🚗	🦆	🙂	🏍	🍾	**mean**
NHGM	86.5%	72.2%	99.9%	79.3%	89.4%	85.5%
NHGM-v2	**97.4%**	93.9%	**100%**	**98.6%**	98.9%	97.8%
Ours	97.1%	**95.4%**	**100%**	98.4%	**99.6%**	**98.1%**

6 ABLATION STUDY

We set out to demonstrate the advantage of our approach. An alternative would be to strictly enforce the discreteness of predictions, and softly enforce adherence to matching constraints (Equation 1) as a loss function.

Thus, discreteness is enforced by row-wise $\arg\max$ predictions. A loss term is added, which distributes any target node excessive matching across the associated target column elements. In unbalanced matching, this loss translates to

$$\ell_{lm}(y^*(x^{ij})) = \frac{max(0, \sum_{t=1}^{n_i} y^*_{tm}(x^{ij}) - 1)}{\sum_{t=1}^{n_i} y^*_{tm}(x^{ij})} \ \forall l = 1,..,n_i. \tag{11}$$

This loss decomposes to the score function dimension, which allows computing the direct loss minimization gradient (Equation 5) easily. We compare between two methods of unbalanced keypoint matching on the Pascal VOC dataset: 'regular'- our approach which strictly enforces matching predictions (Equation 3) with a linear assignment solver, and 'soft matching'- strictly enforces discrete predictions and softly enforces matching constraints (Equation 11). It comes with no surprise that by softly enforcing matching constraints lower average accuracy is achieved (Figure 6) since matching constraints carry a combinatorial semantic which is hard to optimize. Please refer to Appendix A.3.6 for further details.

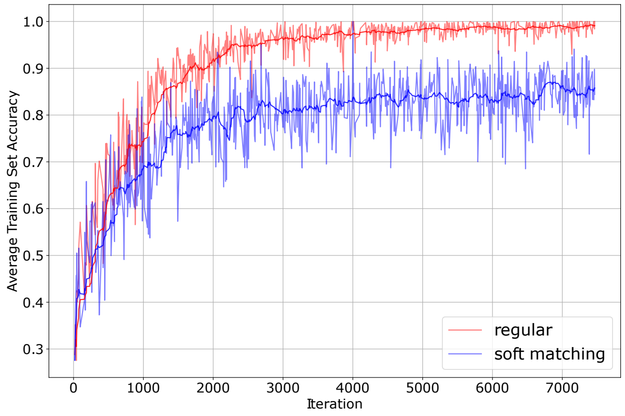


Figure 6: Average training accuracy on the Pascal VOC dataset for unbalanced matching. We compare between two methods of enforcing matching constraints: 'regular' strictly enforces matching predictions, and 'soft matching' strictly enforces discreteness and softly enforces matching constraints.

7 CONCLUSIONS

Our focus in this research has been addressing the challenges of learning multi-graph matching. In summary, our contribution is extending the applicability of the direct loss minimization technique to complexly constrained structured spaces. To that end, we use a penalty function to decompose matching constraints from cycle-consistency constraints, such that cycle-consistency constraints are softly enforced as loss functions and pairwise matchings are strictly enforced in predictions. We prove that this extension is not limiting the direct loss minimization as it theoretically can recover the optimal multi-graph matching solution (Theorem 1). Our method easily extends to two-hypergraph matching and two-graph matching and allows unbalanced and balanced matchings in a principled way.

Experimental validation demonstrates our method's notable advantage in predicting unbalanced two-graph and hypergraph matching. Further, our multi-graph unbalanced matching improves significantly compared to other learning-based methods (GA-MGM) and achieves similar accuracy to the multi-graph matching solver-based method (BB-GM-Multi) at roughly one-quarter of the inference time. Also, our method is effective for balanced graph matching, achieving comparable or slightly better than peer methods.

References

L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1365–1372, 2009. doi: 10.1109/ICCV.2009.5459303.

T. S. Caetano, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. doi: 10.1109/ICCV.2007.4408838.

H. Chen, Z. Luo, J. Zhang, L. Zhou, X. Bai, Z. Hu, C.-L. Tai, and L. Quan. Learning to match features with seeded graph matching network. *International Conference on Computer Vision (ICCV)*, 2021.

Y. Chen, L. Guibas, and Q. Huang. Near-optimal joint object matching via convex relaxation. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32. PMLR, 2014.

M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *CVPR 2011*, volume 6315, pages 492–505, 09 2010. ISBN 978-3-642-15554-3. doi: 10.1007/978-3-642-15555-0_36.

M. Cho, K. Alahari, and J. Ponce. Learning graphs to match. In *Proceedings of the IEEE Interational Conference on Computer Vision*, 2013.

H. Cohen Indelman and T. Hazan. Learning randomly perturbed structured predictors for direct loss minimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139. PMLR, 2021.

A. Egozi, Y. Keller, and H. Guterman. A probabilistic approach to spectral graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):18–27, 2013. doi: 10.1109/TPAMI.2012.51.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. Hypergraph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 2019.

M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege. Deep graph matching consensus. In *International Conference on Learning Representations (ICLR)*, 2020.

Q. Gao, F. Wang, N. Xue, J.-G. Yu, and G.-S. Xia. Deep graph matching under quadratic constraint. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5067–5074, 2021. doi: 10.1109/CVPR46437.2021.00503.

S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996. doi: 10.1109/34.491619.

T. Hazan, J. Keshet, and D. McAllester. Direct loss minimization for structured prediction. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

B. Jiang, P. Sun, and B. Luo. Glmnet: Graph learning-matching convolutional networks for feature matching. *Pattern Recogn.*, 121(C), 2022. doi: 10.1016/j.patcog.2021.108167.

J. Keshet, C.-C. Cheng, M. Stoehr, and D. A. McAllester. Direct error rate minimization of hidden markov models. In *INTERSPEECH*, 2011.

J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted random walks. In *CVPR 2011*, pages 1633–1640, 2011. doi: 10.1109/CVPR.2011.5995387.

M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1482–1489 Vol. 2, 2005. doi: 10.1109/ICCV.2005.20.

M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *International Journal of Computer Vision*, 96:28–45, 04 2012. doi: 10.1007/s11263-011-0442-2.

E. Maset, F. Arrigoni, and A. Fusiello. Practical and efficient multi-view matching. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4578–4586, 2017.

A. Nowak, S. Villar, A. S. Bandeira, and J. Bruna. Revised note on learning quadratic assignment with graph neural networks. *2018 IEEE Data Science Workshop (DSW)*, pages 1–5, 2018.

D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

S. Phillips and K. Daniilidis. All graphs lead to rome: Learning geometric and cycle-consistent representations with graph convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop: Image Matching: Local Features and Beyond*, 2019.

M. V. Pogančić, A. Paulus, V. Musil, G. Martius, and M. Rolinek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2020.

M. Rol'inek, P. Swoboda, D. Zietlow, A. Paulus, V. Musil, and G. Martius. Deep graph matching via blackbox differentiation of combinatorial solvers. In *ECCV*, 2020.

R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35(2):876–879, 1964. doi: 10.1214/aoms/1177703591.

Y. Song, A. G. Schwing, R. S. Zemel, and R. Urtasun. Training deep neural networks via direct loss minimization. In *ICML*, 2016.

P. Swoboda, C. Rother, H. A. Alhaija, D. Kainmüller, and B. Savchynskyy. A study of lagrangean decompositions and dual ascent solvers for graph matching. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7062–7071, 2017.

P. Swoboda, D. Kainm"uller, A. Mokarian, C. Theobalt, and F. Bernard. A convex relaxation for multi-graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

L. Torresani, V. Kolmogorov, and C. Rother. A dual decomposition approach to feature correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (2):259–271, 2013. doi: 10.1109/TPAMI.2012.105.

R. Wang, J. Yan, and X. Yang. Learning combinatorial embedding networks for deep graph matching. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3056–3065, 2019a.

R. Wang, J. Yan, and X. Yang. Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching, 2019b.

R. Wang, J. Yan, and X. Yang. Graduated assignment for joint multi-graph matching and clustering with application to unsupervised graph matching network learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020a.

R. Wang, J. Yan, and X. Yang. Combinatorial learning of robust deep graph matching: an embedding based approach. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2020b.

R. Wang, J. Yan, and X. Yang. Neural graph matching network: Learning lawlerś quadratic assignment problem with extension to hypergraph and multiple-graph matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2021. doi: 10.1109/TPAMI.2021.3078053.

T. Wang, H. Liu, Y. Li, Y. Jin, X. Hou, and H. Ling. Learning combinatorial solver for graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020c.

T. Yu, R. Wang, J. Yan, and B. Li. Learning deep graph matching with channel-independent embedding and hungarian attention. In *ICLR*, 2020.

A. Zanfir and C. Sminchisescu. Deep learning of graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. doi: 10.1109/CVPR.2008.4587500.

Z. Zhang and W. S. Lee. Deep graphical feature learning for the feature matching problem. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Z. Zhang, Q. Shi, J. McAuley, W. Wei, Y. Zhang, and A. van den Hengel. Pairwise matching through max-weight bipartite belief propagation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1202–1210, 2016. doi: 10.1109/CVPR.2016.135.

F. Zhou and F. D. la Torre. Factorized graph matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38(09), 2016. doi: 10.1109/TPAMI.2015.2501802.

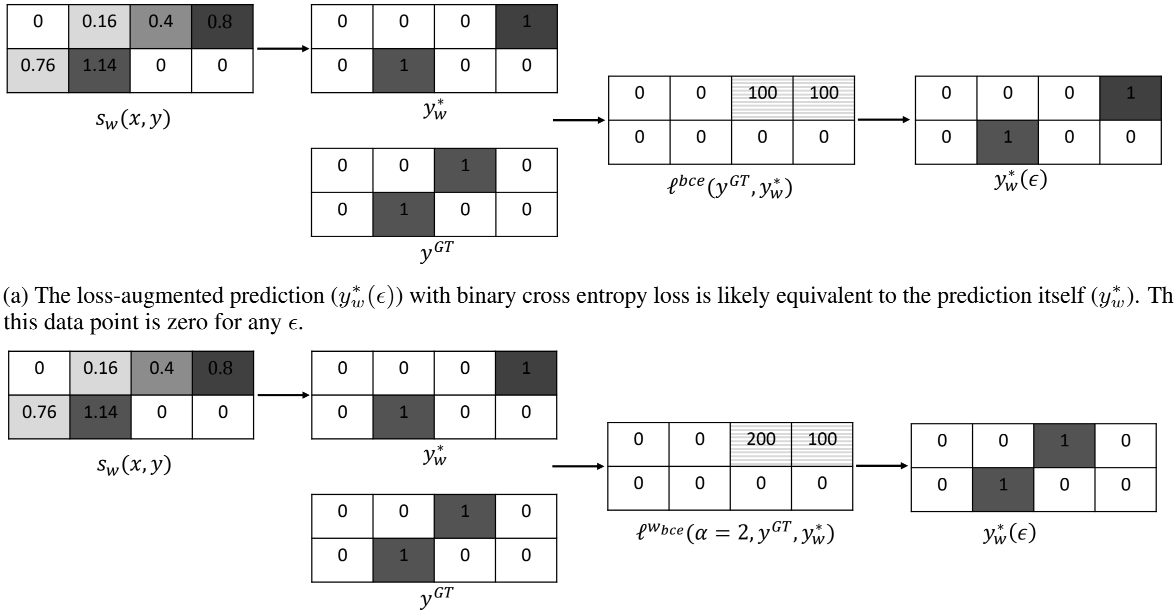
X. Zhou, M. Zhu, and K. Daniilidis. Multi-image matching via fast alternating minimization. *IEEE International Conference on Computer Vision (ICCV)*, pages 4032–4040, 2015.

A APPENDIX

A.1 BINARY CROSS-ENTROPY LOSS WITH DISCRETE PREDICTIONS

Recall that our predictions y^* are discrete. Then, BCE loss would penalize false negatives and false positives equally. We further highlight that pytorch clamps its BCE loss to -100 to achieve stability in backpropogation. As a consequence, the loss-augmented prediction $y^*(\epsilon)$ would likely be identical to y^* (Figure 7a). To generate a ”moving-towards” low-loss gradient step we adopt a weighted BCE loss ($\alpha > 1$) which penalizes false negatives more heavily.

We list the hyperparameters setting of α in the experiments section.



0	0.16	0.4	0.8
0.76	1.14	0	0

0	0	0	1
0	1	0	0

0	0	1	0
0	1	0	0

0	0	100	100
0	0	0	0

0	0	0	1
0	1	0	0

(a) The loss-augmented prediction ($y_w^*(\epsilon)$) with binary cross entropy loss is likely equivalent to the prediction itself (y_w^*). The gradient for this data point is zero for any ϵ.

0	0.16	0.4	0.8
0.76	1.14	0	0

0	0	0	1
0	1	0	0

0	0	1	0
0	1	0	0

0	0	200	100
0	0	0	0

0	0	1	0
0	1	0	0

(b) A weighted binary cross entropy loss penalizes false negatives more heavily than false positives. The corresponding loss-augmented prediction ($y_w^*(\epsilon)$) is likely to choose a lower loss structure than the prediction (y_w^*). Such a dynamics generates a "towards-better" gradient step.

Figure 7: A toy example illustrating the gradient step dynamics of direct loss minimization with binary cross entropy loss versus weighted binary cross entropy loss.

A.2 CONSIDERATIONS OF SETTING THE LOSS-AUGMENTATION PARAMETER

The loss-augmented prediction $y^*(\epsilon)$ (Equation 4) chooses a structure with a lower loss than the prediction’s y^* (Equation 3). The direct loss minimization gradient (Equation 5) encourages moves toward better predictions as it increases the score function for the low-loss structure $s_w(x, y^*(\epsilon))$. This gradient holds for $\epsilon \to 0$. However, in practice, such a small loss augmentation would likely result in zero gradients since the prediction would equal the loss-augmented prediction. To escape zero gradients, we set an initial small ϵ and increase it by a certain amount whenever the loss is positive and the gradients are zero.

A.3 EXPERIMENTS

We use unchanged peer methods implementations in the unified ThinkMatch project as much as possible. Our code was written in adherence with the setting of the ThinkMatch project to allow fair comparison.

Datasets should be downloaded and organized as instructed in the ThinkMatch project.

Unbalanced matching is formed by setting both problem configurations $TGT\_OUTLIER$ and $SRC\_OUTLIER$ to $TRUE$. In both experiments, each image is cropped to its bounding box and scaled to 256×256 px.

UNBALANCED MATCHING

Average imbalance in sampled pairs of images from the Pascal VOC dataset. We measure the imbalance (max/min) of the number of keypoints in 6385 pairs of images sampled from the Pascal VOC dataset. Results in Table 5 show that the average imbalance over all sampled pairs is 1.14. The class with the lowest imbalance is 'bottle' (1.14) and the class with the highest imbalance is 'sheep' (1.96).

Table 5: The average overall and per-class imbalance (max/min) of the number of keypoints in 6385 pairs of images sampled from the Pascal VOC dataset.

Class	✈	🚲	🐦	🚢	🍾	🚌	🚗	🐈	🪑	🐄	🍽	🐕	🐎	🏍	🚶	🪴	🐑	🛋	🚆	🖥	average
imbalance	1.32	1.19	1.464	1.32	1.14	1.22	1.38	1.458	1.4	1.364	1.708	1.4	1.42	1.42	1.45	1.267	1.96	1.52	1.42	1.26	1.40

We run experiments on the Pascal VOC on an Nvidia Tesla K80 24GB GPU.

General hyper-parameters and settings. We experiment with a small range of parameters related to our gradient step. Specifically, $\epsilon \in \{3e-5, 5e-5\}$ and noise scaling factor $\sigma \in \{24, 38\}$.

Vertex classifier and loss We do not perform Sinkhorn normalization prior to the matching prediction in all unbalanced matching experiments. To accomplish this we change the vertex classification of the NGM-v2 architecture such a matching is predicted (y^*_w) on the unnormalized randomly perturbed score function ($s_w(x, y)$). A per-element weighted binary cross entropy is calculated given ground-truth matching. Our method allows for estimating the gradient of the expected loss.

Normalizations We experimented with performing Sinkhorn normalization in gnn embedding layers, and find it gave slightly better results in the two-graph matching experiment only.

We find it beneficial in the two-graph and multi-matching matching experiments to normalize by channels the output of the gnn embedding layer. This normalization in a way replaces the Sinkhorn normalization, though it does not generate doubly-stochastic representations. We also find it beneficial in all experiments to add channel-wise normalization in the 'SiameseSConvOnNodes' class.

Configurations To recreate this setup, set in the config file: $MATCHING\_TYPE =$ 'Unbalanced' and $filter\_type =$ 'NoFilter'.

A.3.1 Multi-Graph Unbalanced Matching

Hyper-parameters. We set $\epsilon = 7e-5$, sigma noise scaling parameter $\sigma = 38$. The sigma noise decays by 1.002 with every epoch. The cycle-inconsistency penalty parameter n is set initially to 25 and increases by 75 with every epoch. The batch size is equal to 20. Sets of 5 image pairs are sampled.

Sinkhorn normalization is not performed in the gnn embedding layers (We set $SK\_ITER\_NUM = 0$).

A.3.2 Two-Graph Unbalanced Matching

Hyper-parameters. We set $\epsilon = 3e-5$, sigma noise scaling parameter $\sigma = 38$. The sigma noise decays by 1.002 with every epoch. Sinkhorn normalization is performed in the gnn embedding layers. We set $SK\_ITER\_NUM = 20$ such that 20 iterations of Sinkhorn normalization are performed.

The batch size is equal to 26.

A.3.3 Hypergraph Unbalanced Matching

Hyper-parameters. We set $\epsilon = 3e-4$, sigma noise scaling parameter $\sigma = 24$. The sigma noise decays by 1.02 with every epoch.

We change the NGM-v2 architecture such that Sinkhorn normalization is not performed in the matching-aware gnn embedding layers (We set $SK\_ITER\_NUM = 0$). The batch size is equal to 26.

BALANCED MATCHING

The two-graph matching and the multi-matching experiment were performed on an Nvidia Tesla K80 12GB GPU. The hypergraph matching experiment was performed on an Nvidia Tesla K80 24GB GPU.

Peer results. All peer methods were evaluated using the ThinkMatch project or quoted from peer papers.

General Hyper-parameters and settings. We set $SK\_ITER\_NUM = 20$ such that 20 iterations of Sinkhorn normalization are performed. We experiment with a small range of parameters related to our gradient step. specifically, $\epsilon \in \{3e-5, 5e-5\}$ and noise scaling factor $\sigma \in \{24, 38\}$.

We do not perform Sinkhorn normalization prior to the matching prediction neither in the two-graph matching experiment nor in the multi-matching experiment. Sinkhorn normalization is performed in the gnn embedding layers.

We find it beneficial in the two-graph and multi-matching matching experiments to normalize by channels the output of the gnn embedding layer. This normalization in a way replaces the Sinkhorn normalization, though it does not generate doubly-stochastic representations. We do not perform this channel-wise normalization in the hypergraph matching experiment (and do perform Sinkhorn normalization prior to prediction)

We find it beneficial in all experiments to add channel-wise normalization in the 'SiameseSConvOnNodes' class.

Configurations To recreate this setup, set in the config file: $MATCHING\_TYPE =' Balanced'$.

A.3.4 Multi-Graph Matching

Hyper-parameters. We set $\epsilon = 5e-5$, sigma noise scaling parameter $\sigma = 38$. The sigma noise decays by 1.002 with every epoch. The cycle-inconsistency penalty parameter $\mathcal{C}$ is set initially to 0 and increases by 1300 with every epoch. The batch size is equal to 6. We experimented with sampling sets of 5 and 12 and image pairs. Finally, sets of 5 image pairs gave the best results.

A.3.5 Two-Graph Matching

Hyper-parameters. We set $\epsilon = 3e-5$, sigma noise scaling parameter $\sigma = 38$. The sigma noise decays by 1.002 with every epoch. The batch size is equal to 26.

A.3.6 Hypergraph matching

Hyper-parameters. Models are pretrained on the corresponding two-graph matching problem.

We set $\epsilon = 3e-5$, sigma noise scaling parameter $\sigma = 24$. The sigma noise decays by 1.002 with every epoch. We perform Sinkhorn normalization prior to matching prediction. Results for both ours and NHGM-v2 are obtained with batch size =26.

Ablation Study Details

We keep the two-graph matching architecture and hyper-parameters unchanged, However, the prediction $y*$ is performed by applying argmax function on each row, such that a valid matching is no longer enforced. The weighted binary cross entropy loss is augmented with an additional loss function $\ell_{lm}(y^*(x^{ij}))$ which softly enforces matching constraints. In balanced matching, $\ell_{lm}(y^*(x^{ij}))$ translates to

$$\ell_{lm}(y^*(x^{ij})) = \frac{\sum_{t=1}^{n_i} y^*_{tm}(x^{ij}) - 1}{\sum_{t=1}^{n_i} y^*_{tm}(x^{ij})} \; \forall l = 1, .., n_i. \tag{12}$$

.