Supplementary Material of "Efficient Multitask Feature and Relationship Learning"

In this supplementary material we provide all the missing proofs to the propositions, lemmas and theorems in our main paper "Efficient Multitask Feature and Relationship Learning".

A PROOFS

A.1 Proof of Proposition 4.1

Proposition 4.1. (5) can be solved in closed form in $O(m^3d^3 + mnd^2)$ time; the optimal solution W^* is: $\text{vec}(W^*) = \left(I_m \otimes (X^TX) + \eta\, \Sigma_2 \otimes \Sigma_1\right)^{-1} \text{vec}(X^TY)$.

To prove this claim, we need the following facts about tensor product:

Fact A.1. Let A be a matrix. Then $||A||_F = ||\text{vec}(A)||_2$.

Fact A.2. Let $A \in \mathbb{R}^{m_1 \times n_1}$, $B \in \mathbb{R}^{n_1 \times n_2}$ and $C \in \mathbb{R}^{n_2 \times m_2}$. Then $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$.

Fact A.3. Let $S_1 \in \mathbb{R}^{m_1 \times n_1}, S_2 \in \mathbb{R}^{n_1 \times p_1}$ and $T_1 \in \mathbb{R}^{m_2 \times n_2}, T_2 \in \mathbb{R}^{n_2 \times p_2}$. Then $(S_1 \otimes S_2)(T_1 \otimes T_2) = (S_1S_2) \otimes (T_1T_2)$.

Fact A.4. Let $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{m \times m}$. Let $\{\mu_1, \ldots, \mu_n\}$ be the spectrum of A and $\{\nu_1, \ldots, \nu_m\}$ be the spectrum of B. Then the spectrum of $A \otimes B$ is $\{\mu_i\nu_j : 1 \le i \le n, 1 \le j \le m\}$.

We can show the following result by transforming W into its isomorphic counterpart:

Proof.

$$
\begin{aligned}
& ||Y - XW||_F^2 + \eta\, ||\Sigma_1^{1/2} W \Sigma_2^{1/2}||_F^2 \\
&= ||\text{vec}(Y - XW)||_2^2 + \eta\, ||\text{vec}(\Sigma_1^{1/2} W \Sigma_2^{1/2})||_2^2 && \text{(By Fact A.1)} \\
&= ||\text{vec}(Y) - (I_m \otimes X)\text{vec}(W)||_2^2 + \eta\, ||(\Sigma_2^{1/2} \otimes \Sigma_1^{1/2})\text{vec}(W)||_2^2 && \text{(By Fact A.2)} \\
&= \text{vec}(W)^T \left((I_m \otimes X)^T (I_m \otimes X) + \eta (\Sigma_2^{1/2} \otimes \Sigma_1^{1/2})^T (\Sigma_2^{1/2} \otimes \Sigma_1^{1/2}) \right) \text{vec}(W) \\
&\quad - 2\text{vec}(W)^T (I_m \otimes X^T)\text{vec}(Y) + \text{vec}(Y)^T \text{vec}(Y) \\
&= \text{vec}(W)^T \left((I_m \otimes X^TX) + \eta(\Sigma_2 \otimes \Sigma_1) \right) \text{vec}(W) \\
&\quad - 2\text{vec}(W)^T (I_m \otimes X^T)\text{vec}(Y) + \text{vec}(Y)^T \text{vec}(Y) && \text{(By Fact A.3)}
\end{aligned}
$$

The last equation above is a quadratic function of $\text{vec}(W)$, from which we can read off that the optimal solution W^* should satisfy:

$$\text{vec}(W^*) = \left(I_m \otimes (X^TX) + \eta\Sigma_2 \otimes \Sigma_1\right)^{-1} \text{vec}(X^TY) \tag{12}$$

W^* can then be obtained simply by reformatting $\text{vec}(W^*)$ into a $d \times m$ matrix. The computational bottleneck in the above procedure is in solving an $md \times md$ system of equations, which scales as $O(m^3d^3)$ if no further structure is available. The overall computational complexity is $O(m^3d^3 + mnd^2)$. ■

A.2 Proof of Proposition 4.2

To analyze the convergence rate of gradient descent in this case, we start by bounding the smallest and largest eigenvalue of the quadratic system.

Lemma A.1 (Weyl's inequality)**.** Let A, B and C be n-by-n Hermitian matrices, and $C = A + B$. Let $a_1 \ge \cdots \ge a_n$, $b_1 \ge \cdots \ge b_n$ and $c_1 \ge \cdots \ge c_n$ be the eigenvalues of A, B and C respectively. Then the following inequalities hold for $r + s - 1 \le i \le j + k - n, \forall i = 1, \ldots, n$:

$$a_j + b_k \le c_i \le a_r + b_s$$

Let $\lambda_k(A)$ be the k-th largest eigenvalue of matrix A.

Lemma A.2. If Σ_1 and Σ_2 are feasible in (4), then

$$\lambda_1(I_m \otimes (X^T X) + \eta I_{md} + \rho \Sigma_2 \otimes \Sigma_1) \leq \lambda_1(X^T X) + \eta + \rho u^2$$
$$\lambda_{md}(I_m \otimes (X^T X) + \eta I_{md} + \rho \Sigma_2 \otimes \Sigma_1) \geq \lambda_d(X^T X) + \eta + \rho l^2$$

Proof. By Weyl's inequality, setting $r = s = i = 1$, we have $c_1 \leq a_1 + b_1$. Set $j = k = i = n$, we have $c_n \geq a_n + b_n$. We can bound the largest and smallest eigenvalues of $I_m \otimes (X^T X) + \eta I_{md} + \rho \Sigma_2 \otimes \Sigma_1$ as follows:

$$\begin{aligned}
&\lambda_1(I_m \otimes (X^T X) + \eta I_{md} + \rho \Sigma_2 \otimes \Sigma_1) \\
&\leq \lambda_1(I_m \otimes (X^T X)) + \lambda_1(\eta I_{md}) + \lambda_1(\rho \Sigma_2 \otimes \Sigma_1) && \text{(By Weyl's inequality)} \\
&= \lambda_1(I_m)\lambda_1(X^T X) + \eta + \rho \lambda_1(\Sigma_1)\lambda_1(\Sigma_2) && \text{(By Fact A.4)} \\
&\leq \lambda_1(X^T X) + \eta + \rho u^2 && \text{(By the feasibility assumption)}
\end{aligned}$$

and

$$\begin{aligned}
&\lambda_{md}(I_m \otimes (X^T X) + \eta I_{md} + \rho \Sigma_2 \otimes \Sigma_1) \\
&\geq \lambda_{md}(I_m \otimes (X^T X)) + \lambda_{md}(\eta I_{md}) + \lambda_{md}(\rho \Sigma_2 \otimes \Sigma_1) && \text{(By Weyl's inequality)} \\
&= \lambda_m(I_m)\lambda_d(X^T X) + \eta + \rho \lambda_m(\Sigma_1)\lambda_d(\Sigma_2) && \text{(By Fact A.4)} \\
&\geq \lambda_d(X^T X) + \eta + \rho l^2 && \text{(By the feasibility assumption)}
\end{aligned}$$

■

We will first introduce the following two lemmas adapted from (Nesterov, 2013), using the fact that the spectral norm of the Hessian matrix $\nabla^2 h(W)$ is bounded.

Lemma A.3. Let $f(W) : \mathbb{R}^{d \times m} \mapsto \mathbb{R}$ be a twice differentiable function with $\lambda_1(\nabla^2 f(W)) \leq L$. $L > 0$ is a constant. The minimum value of $f(W)$ can be achieved. Let $W^* = \arg\min_W f(W)$, then

$$f(W^*) \leq f(W) - \frac{1}{2L} ||\nabla f(W)||_F^2$$

Lemma A.4. Let $f(W) : \mathbb{R}^{d \times m} \mapsto \mathbb{R}$ be a convex, twice differentiable function with $\lambda_1(\nabla^2 f(W)) \leq L$. $L > 0$ is a constant, then $\forall W_1, W_2$:

$$\text{tr}\left((\nabla f(W_1) - \nabla f(W_2))^T (W_1 - W_2)\right) \geq \frac{1}{L} ||\nabla f(W_1) - \nabla f(W_2)||_F^2$$

We can now proceed to show Proposition 4.2.

Proposition 4.2. Let $\lambda_l = \lambda_d(X^T X) + \eta l^2$ and $\lambda_u = \lambda_1(X^T X) + \eta u^2$. Choose $0 < t \leq \frac{2}{\lambda_u + \lambda_l}$. For all $\varepsilon > 0$, gradient descent with step size t converges to the optima within $O(\log(1/\varepsilon))$ steps.

Proof. Define function $g(W)$ as follows:

$$g(W) = h(W) - \frac{\lambda_l}{2} ||W||_F^2$$

Since we have already bounded that $\lambda_{md}(\nabla^2 h(W)) \geq \lambda_l$, it follows that $g(W)$ is a convex function and furthermore $\lambda_1(\nabla^2 g(W)) \leq \lambda_u - \lambda_l$. Applying Lemma A.4 to g, $\forall W_1, W_2 \in \mathbb{R}^{d \times m}$, we have:

$$\text{tr}\left((\nabla g(W_1) - \nabla g(W_2))^T (W_1 - W_2)\right) \geq \frac{1}{\lambda_u - \lambda_l} ||\nabla g(W_1) - \nabla g(W_2)||_F^2$$

Plug in $\nabla g(W) = \nabla h(W) - \lambda_l W$ into the above inequality and after some algebraic manipulations, we have:

$$\text{tr}\left((\nabla h(W_1) - \nabla h(W_2))^T (W_1 - W_2)\right) \geq \frac{1}{\lambda_u + \lambda_l} ||\nabla h(W_1) - \nabla h(W_2)||_F^2 + \frac{\lambda_u \lambda_l}{\lambda_u + \lambda_l} ||W_1 - W_2||_F^2 \qquad (13)$$

Let $W^* = \arg\min_W h(W)$. Within each iteration of the algorithm, we have the update formula as $W^+ = W - t\nabla h(W)$, we can bound $||W^+ - W^*||_F^2$ as follows

$$\begin{aligned}
||W^+ - W^*||_F^2 &= ||W - W^* - t\nabla h(W)||_F^2 \\
&= ||W - W^*||_F^2 + t^2||\nabla h(W)||_F^2 - 2t\text{tr}\left((W - W^*)^T\nabla h(W)\right) \\
&\leq (1 - 2t\frac{\lambda_u\lambda_l}{\lambda_u + \lambda_l})||W - W^*||_F^2 + t(t - \frac{2}{\lambda_u + \lambda_l})||\nabla h(W)||_F^2 \quad \text{(By inequality 13)} \\
&\leq (1 - 2t\frac{\lambda_u\lambda_l}{\lambda_u + \lambda_l})||W - W^*||_F^2 \quad \text{(For } 0 < t \leq 2/(\lambda_u + \lambda_l))
\end{aligned}$$

Apply the above inequality recursively for T times, we have

$$||W^{(T)} - W^*||_F^2 \leq \gamma^T||W^{(0)} - W^*||_F^2$$

where $\gamma = 1 - 2t\frac{\lambda_u\lambda_l}{\lambda_u+\lambda_l}$. For $t = 2/(\lambda_u + \lambda_l)$, we have

$$\gamma = 1 - 4\lambda_u\lambda_u/(\lambda_l + \lambda_u)^2 = \left(\frac{\lambda_u - \lambda_l}{\lambda_u + \lambda_l}\right)^2$$

Now pick $\forall\varepsilon > 0$, setting the upper bound $\gamma^T||W^{(0)} - W^*||_F^2 \leq \varepsilon$ and solve for T, we have

$$T \geq \log_{1/\gamma}(C/\varepsilon) = O(\log_{1/\gamma}(1/\varepsilon)) = O(\kappa\log(1/\varepsilon))$$

where $C = ||W^{(0)} - W^*||_F^2$ is a constant, and $\kappa = \lambda_u/\lambda_l$ is the condition number. ■

A.3 Detailed Proof on Optimization of Σ_1 and Σ_2

In this section we show the detailed derivation on how to solve (7) efficiently.

As mentioned in the main paper, since $\Sigma_2 \in \mathbb{S}_{++}^m$, it follows that $W\Sigma_2W^T \in \mathbb{S}_+^d$. Without loss of generality, using spectral decomposition, we can reparametrize $\Sigma_1 = U\Lambda U^T$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ with $u \geq \lambda_1 \geq \lambda_2 \cdots \geq \lambda_d \geq l$ and $U \in \mathbb{R}^{d\times d}$ with $U^TU = UU^T = I_d$. Similarly, we can represent $W\Sigma_2W^T = VNV^T$ where $V \in \mathbb{R}^{d\times d}$, $V^TV = VV^T = I_d$ and $N = \text{diag}(\nu_1, \ldots, \nu_d)$ with $0 \leq \nu_1 \leq \cdots \leq \nu_d$. Note that the eigenvectors in N corresponds to eigenvalues in increasing order rather than decreasing order, for reasons that will become clear below. Realizing that U is an orthonormal matrix, we have:

$$\log|\Sigma_1| = \log|U\Lambda U^T| = \log|\Lambda|, \quad \text{tr}(\Sigma_1W\Sigma_2W^T) = \text{tr}(\Lambda U^TVNV^TU) \tag{14}$$

Set $K = U^TV$. Since both U and V are orthonormal matrices, K is also an orthonormal matrix. We can further transform (14) to be $\text{tr}(\Lambda U^TVNV^TU) = \text{tr}((\Lambda K)(KN)^T)$. Note that the mapping between U and K is bijective since V is a fixed orthonormal matrix. Using K and Λ, we can equivalently transform the optimization problem (7) into the following new form:

$$\min_{K,\Lambda} \text{tr}((\Lambda K)(KN)^T) - m\log|\Lambda|, \quad \text{s.t. } l\,\text{diag}(\mathbf{1}_d) \leq \Lambda \leq u\,\text{diag}(\mathbf{1}_d), K^TK = KK^T = I_d \tag{15}$$

where $\mathbf{1}_d$ is a d-dimensional vector of all ones. At first glance it seems that the new form of optimization is more complicated to solve since it is even not a convex problem due to the quadratic equality constraint. However, as we will see shortly, the new form helps to decouple the interaction between K and Λ in that K does not influence the second term $-m\log|\Lambda|$. This implies that we can first partially optimize over K, finding the optimal solution as a function of Λ, and then optimize over Λ. Mathematically, it means:

$$\min_{K,\Lambda} -m\log|\Lambda| + \text{tr}((\Lambda K)(KN)^T) \quad \Leftrightarrow \quad \min_{\Lambda} -m\log|\Lambda| + \min_{K} \text{tr}((\Lambda K)(KN)^T) \tag{16}$$

So that we can first consider the minimization over K: $\text{tr}((\Lambda K)(KN)^T) = \sum_{i=1}^d\sum_{j=1}^d \lambda_iK_{ij}^2\nu_j = \lambda^TP\nu$, where we define $P = K \circ K$, $\lambda = (\lambda_1, \cdots, \lambda_d)^T$ and $\nu = (\nu_1, \cdots, \nu_d)^T$. Since K is an orthonormal matrix, we have the

following two equations: $\sum_{j=1}^{d} P_{ij} = \sum_{j=1}^{d} K_{ij}^2 = 1, \quad \forall i \in [d]$, $\sum_{i=1}^{d} P_{ij} = \sum_{i=1}^{d} K_{ij}^2 = 1, \quad \forall j \in [d]$, which implies that P is a doubly stochastic matrix

$$\min_{K} \operatorname{tr}((\Lambda K)(KN)^T) = \min_{P} \lambda^T P \nu \tag{17}$$

In order to solve the minimization over the doubly stochastic matrix P, we need to introduce the following theorems.

Lemma A.5 ((Bertsimas and Tsitsiklis, 1997)). Consider the minimization of a linear program over a polyhedron P. Suppose that P has at least one extreme point and that there exists an optimal solution. Then there exists an optimal solution that is an extreme point of P.

Definition A.1 (Birkhoff polytope). The *Birkhoff polytope* B_d is the set of $d \times d$ doubly stochastic matrices. B_d is a convex polytope.

Lemma A.6 (Birkhoff-von Neumann theorem). Let B_d be the Birkhoff polytope. B_d is the convex hull of the set of $d \times d$ permutation matrices. Furthermore, the vertices (extreme points) of B_d are the permutation matrices.

Combine the above two theorems, it is clear to see that there exists an optimal solution P to the optimization problem (17) that is a $d \times d$ permutation matrix. This implies that we can reduce (17) to a minimum-weight perfect matching problem on a weighted complete bipartite graph.

Definition A.2 (Minimum-weight perfect matching). Let $G = (V, E)$ be an undirected graph with edge weight $w : E \to \mathbb{R}_+$. A *perfect matching* in G is a set $M \subseteq E$ such that no two edges in M have a vertex in common and every vertex from V occurs as the endpoint of some edge in M. A matching M is called a minimum-weight perfect matching if it is a perfect matching that has the minimum weight among all the perfect matchings of G.

For any $\lambda, \nu \in \mathbb{R}_+^d$, we can construct a weighted d by d bipartite graph $G = (V_\lambda, V_\nu, E; w)$ as follows:

- For each λ_i, construct a vertex $v_{\lambda_i} \in V_\lambda, \forall i$.
- For each ν_j, construct a vertex $v_{\nu_j} \in V_\nu, \forall j$.
- For each pair $(v_{\lambda_i}, v_{\nu_j})$, construct an edge $e(v_{\lambda_i}, v_{\nu_j}) \in E$ with weight $w(e(v_{\lambda_i}, v_{\nu_j})) = \lambda_i \nu_j$.

The following lemma relates the solution of the minimum weight matching to the solution of (17):

Lemma A.7. The minimum value of (17) is equal to the minimum weight of a perfect matching on $G = (V_\lambda, V_\nu, E, w)$.

Surprisingly, we do not even need to run standard graph matching algorithms to solve our matching problem. Instead, Thm. A.1 gives a closed form solution.

Theorem A.1. Let $\lambda = (\lambda_1, \ldots, \lambda_d)$ and $\nu = (\nu_1, \ldots, \nu_d)$ with $\lambda_1 \geq \cdots \geq \lambda_d$ and $\nu_1 \leq \cdots \leq \nu_d$. The minimum-weight perfect matching on G is $\pi^* = \{(v_{\lambda_i}, v_{\nu_i}) : 1 \leq i \leq d\}$ with the minimum weight $w(\pi^*) = \sum_{i=1}^{d} \lambda_i \nu_i$.

The permutation matrix that achieves the minimum weight is $P^* = I_d$ since $\pi^*(\lambda_i) = \nu_i$. Note that $P = K \circ K$, it follows that the optimal K^* is also I_d. Hence we can solve for the optimal U^* matrix by solving the equation $U^{*T} V = I_d$, which leads to $U^* = V$. Now plug in the optimal $K^* = I_d$ into (16). The optimization w.r.t. Λ decomposes into d independent problems, each of which being a simple scalar optimization problem:

$$\begin{aligned} &\underset{\lambda}{\text{minimize}} && \sum_{i=1}^{d} \lambda_i \nu_i - m \log \lambda_i \\ &\text{subject to} && l \leq \lambda_i \leq u, \quad \forall i = 1, \ldots, d \end{aligned} \tag{18}$$

Depending on whether the value m/ν_i is within the range $[l, u]$, the optimal solution λ_i^* for each scalar minimization problem may take different forms. Define a hard-thresholding operator $\mathbb{T}_{[l,u]}(z)$ as follows:

$$\mathbb{T}_{[l,u]}(z) = \begin{cases} l, & z < l \\ z, & l \leq z \leq u \\ u, & z > u \end{cases} \tag{19}$$

Using this hard-thresholding operator, we can express the optimal solution λ_i^* as $\lambda_i^* = \mathbb{T}_{[l,u]}(m/\nu_i)$, which finishes the proof.

A.3.1 Proof of Lemma A.7

Lemma A.7. The minimum value of (17) is equal to the minimum weight of a perfect matching on $G = (V_\lambda, V_\nu, E, w)$.

Proof. By Lemma A.5 and Lemma A.6, the optimal value is achieved when P is a permutation matrix. Given a permutation matrix P, we can understand P as a bijective mapping from the index of rows to the index of columns. Specifically, construct a permutation $\pi_P : [d] \to [d]$ from P as follows. For each row index $i \in [d]$, $\pi_P(i) = j$ iff $P_{ij} = 1$. It follows that π_P is a permutation of $[d]$ since P is assumed to be a permutation matrix. The objective function in (17) can be written in terms of π_P as

$$\lambda^T P \nu = \sum_{i=1}^{d} \lambda_i \nu_{\pi_P(i)}$$

which is exactly the weight of the perfect matching on $G(V_\lambda, V_\nu, E, w)$ given by π_P:

$$w(\pi_P) = w(\{(i, \pi_P(i)) : 1 \le i \le d\}) = \sum_{i=1}^{d} \lambda_i \nu_{\pi_P(i)}$$

Similarly, in the other direction, given any perfect matching $\pi : [d] \to [d]$ on the bipartite graph $G(V_\lambda, V_\nu, E, w)$, we can construct a corresponding permutation matrix P_π: $P_{\pi,ij} = 1$ iff $\pi(i) = j$, otherwise 0. Since π is a perfect matching, the constructed P_π is guaranteed to be a permutation matrix.

Hence the problem of finding the optimal value of (17) is equivalent to finding the minimum weight perfect matching on the constructed bipartite graph $G(V_\lambda, V_\nu, E, w)$. Note that the above constructive process also shows how to recover the optimal permutation matrix P_{π^*} from the minimum weight perfect matching π^*. ■

A.3.2 Proof of Theorem A.1

Note that $\lambda = (\lambda_1, \ldots, \lambda_d)$ and $\nu = (\nu_1, \ldots, \nu_d)$ are assumed to satisfy $\lambda_1 \ge \cdots \ge \lambda_d$ and $\nu_1 \le \cdots \le \nu_d$. To make the discussion more clear, we first make the following definition of an *inverse pair*.

Definition A.3 (Inverse pair)**.** Given a perfect match π of $G(V_\lambda, V_\nu, E, w)$, $(\lambda_i, \lambda_j, \nu_k, \nu_l)$ is called an *inverse pair* if $i \le j$, $k \le l$ and $(v_{\lambda_i}, v_{\nu_l}) \in \pi$, $(v_{\lambda_j}, v_{\nu_k}) \in \pi$.

Lemma A.8. Given a perfect match π of $G(V_\lambda, V_\nu, E, w)$ and assuming π contains an inverse pair $(\lambda_i, \lambda_j, \nu_k, \nu_l)$. Construct $\pi' = \pi \backslash \{(v_{\lambda_i}, v_{\nu_l}), (v_{\lambda_j}, v_{\nu_k})\} \cup \{((v_{\lambda_i}, v_{\nu_k}), (v_{\lambda_j}, v_{\nu_l})\}$. Then $w(\pi') \le w(\pi)$.

Proof. Let us compare the weights of π and π'. Note that since $i \le j, k \le l$, we have $\lambda_i \ge \lambda_j$ and $\nu_k \le \nu_l$.

$$\begin{aligned} w(\pi') - w(\pi) &= (\lambda_i \nu_k + \lambda_j \nu_l) - (\lambda_i \nu_l + \lambda_j \nu_k) \\ &= (\lambda_i - \lambda_j)(\nu_k - \nu_l) \\ &\le 0 \end{aligned}$$

Intuitively, this lemma says that we can always decrease the weight of a perfect matching by re-matching an inverse

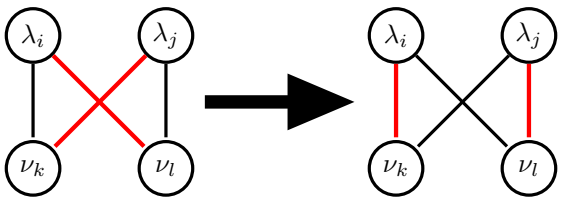

Figure 5: Re-matching an inverse pair $(\lambda_i, \lambda_j, \nu_k, \nu_l) = \{(v_{\lambda_i}, v_{\nu_l}), (v_{\lambda_j}, v_{\nu_k})\}$ on the left side to a match with smaller weight $\{(v_{\lambda_i}, v_{\nu_k}), (v_{\lambda_j}, v_{\nu_l})\}$. Red color is used to highlight edges in the perfect matching.

pair. Fig. 5 illustrates this process. It is worth emphasizing here that the above re-matching process only involves four nodes, i.e., $v_{\lambda_i}, v_{\nu_l}, v_{\lambda_j}$ and v_{ν_k}. In other words, the other parts of the matching stay unaffected. ■

Using Lemma A.8, we are now ready to prove Thm. A.1:

Theorem A.1. Let $\lambda = (\lambda_1, \ldots, \lambda_d)$ and $\nu = (\nu_1, \ldots, \nu_d)$ with $\lambda_1 \geq \cdots \geq \lambda_d$ and $\nu_1 \leq \cdots \leq \nu_d$. The minimum-weight perfect matching on G is $\pi^* = \{(v_{\lambda_i}, v_{\nu_i}) : 1 \leq i \leq d\}$ with the minimum weight $w(\pi^*) = \sum_{i=1}^{d} \lambda_i \nu_i$.

Proof. We will prove by induction.

- **Base case**. The base case is $d = 2$. In this case there are only two valid perfect matchings, i.e., $\{(v_{\lambda_1}, v_{\nu_1}), (v_{\lambda_2}, v_{\nu_2})\}$ or $\{(v_{\lambda_1}, v_{\nu_2}), (v_{\lambda_2}, v_{\nu_1})\}$. Note that the second perfect matching $\{(v_{\lambda_1}, v_{\nu_2}), (v_{\lambda_2}, v_{\nu_1})\}$ is an inverse pair. Hence by Lemma A.8, $w(\{(v_{\lambda_1}, v_{\nu_1}), (v_{\lambda_2}, v_{\nu_2})\}) = \lambda_1\nu_1 + \lambda_2\nu_2 \leq \lambda_1\nu_2 + \lambda_2\nu_1 = w(\{(v_{\lambda_1}, v_{\nu_2}), (v_{\lambda_2}, v_{\nu_1})\})$.
- **Induction step**. Assume Thm. A.1 holds for $d = n$. Consider the case when $d = n + 1$. Start from any perfect matching π. Check the matches of node $v_{\lambda_{n+1}}$ and $v_{\nu_{n+1}}$. Here we have two subcases to discuss:
  - If $v_{\lambda_{n+1}}$ is matched to $v_{\nu_{n+1}}$ in π. Then we can remove nodes $v_{\lambda_{n+1}}$ and $v_{\nu_{n+1}}$ from current graph, and this reduces to the case when $n = d$. By induction assumption, the minimum weight perfect matching on the new graph is given by $\sum_{i=1}^{n} \lambda_i \nu_i$, so the minimum weight on the original graph is $\sum_{i=1}^{n} \lambda_i \nu_i + \lambda_{n+1}\nu_{n+1} = \sum_{i=1}^{n+1} \lambda_i \nu_i$.
  - If $v_{\lambda_{n+1}}$ is not matched to $v_{\nu_{n+1}}$ in π. Let v_{ν_j} be the match of $v_{\lambda_{n+1}}$ and v_{λ_i} be the match of $v_{\nu_{n+1}}$, where $i \neq n + 1$ and $j \neq n + 1$. In this case we have $i < n + 1$ and $j < n + 1$, so $(\lambda_i, \lambda_{n+1}, \nu_j, \nu_{n+1})$ forms an inverse pair by definition. By Lemma A.8, we can first re-match $v_{\lambda_{n+1}}$ to $v_{\nu_{n+1}}$ and v_{λ_i} to v_{ν_j} to construct a new match π' with $w(\pi') \leq w(\pi)$. In the new matching π' we have the property that $v_{\lambda_{n+1}}$ is matched to $v_{\nu_{n+1}}$, and this becomes the above case that we have already analyzed, so we still have the minimum weight perfect matching to be $\sum_{i=1}^{n+1} \lambda_i \nu_i$.

Intuitively, as shown in Fig. 5, an inverse pair corresponds to a cross in the matching graph. The above inductive proof basically works from right to left to recursively remove inverse pairs (crosses) from the matching graph. Each re-matching step in the proof will decrease the number of inverse pairs at least by one. The whole process stops until there is no inverse pair in the current perfect matching. Since the total number of possible inverse pairs, the above process can stop in finite steps. We illustrate the process of removing inverse pairs in Fig. 6. ■

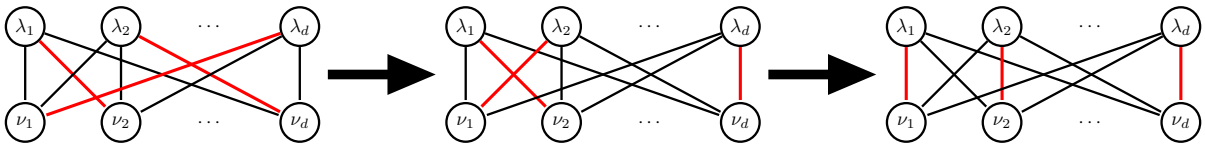

Figure 6: The inductive proof works by recursively removing inverse pairs from the right side of the graph to the left side of the graph. The process stops until there is no inverse pair in the matching. Red color is used to highlight edges in the perfect matching.