Fast and Provably Convergent Algorithms for Gromov-Wasserstein in Graph Learning

Jiajin Li\textsuperscript{1} Jianheng Tang\textsuperscript{2} Lemin Kong\textsuperscript{3} Huikang Liu\textsuperscript{4} Jia Li\textsuperscript{2} Anthony Man-Cho So\textsuperscript{3} Jose Blanchet\textsuperscript{1}

Abstract

In this paper, we study the design and analysis of a class of efficient algorithms for computing the Gromov-Wasserstein (GW) distance tailored to large-scale graph learning tasks. Armed with the Luo-Tseng error bound condition (Luo & Tseng, 1992), two proposed algorithms, called Bregman Alternating Projected Gradient (BAPG) and hybrid Bregman Proximal Gradient (hBPG) are proven to be (linearly) convergent. Upon task-specific properties, our analysis further provides novel theoretical insights to guide how to select the best fit method. As a result, we are able to provide comprehensive experiments to validate the effectiveness of our methods on a host of tasks, including graph alignment, graph partition, and shape matching. In terms of both wall-clock time and modeling performance, the proposed methods achieve state-of-the-art results.

1. Introduction

The Gromov-Wasserstein (GW) distance aims at defining ways to compare probability distributions on unaligned metric spaces which only requires modeling the topological or relational aspects of the distributions within each domain. As such, we have witnessed a fast-growing body of literature that applies the GW distance to various structural data analysis tasks, e.g., 2D/3D shape matching (Peyré et al., 2016; Mémoli & Sapiro, 2004; Mémoli, 2009), molecule analysis (Vayer et al., 2018; Titouan et al., 2019), graph alignment and partition (Chowdhury & Mémoli, 2019; Vincent-Cuaz et al., 2021b; Xu et al., 2019b;a; Chowdhury & Needham, 2021; Gao et al., 2021), generative modeling (Bunne et al., 2019; Xu et al., 2021), to name a few.

Although the GW distance has received much attention for graph learning problems, there are still very few results that address the design of efficient algorithms with provable convergence guarantees. In fact, only recently researchers have proposed an entropy-regularized iterative sinkhorn projection algorithm (Solomon et al., 2016), i.e., eBPG, which is proven to converge under the Kurdyka-Łojasiewicz framework (Attouch et al., 2010; 2013). However, eBPG has several crucial drawbacks that prevent its widespread adoption. First, instead of tackling the GW problem directly, eBPG addresses an entropic regularized GW objective, whose regularization parameter affects the model performance dramatically. Second, eBPG relies on a subroutine (i.e., Sinkhorn (Cuturi, 2013)) to solve the entropic optimal transport problem (Peyré et al., 2019; Benamou et al., 2015) and is thus rather computationally expensive for large-scale graph learning tasks. On another front, Titouan et al. (2019) have introduced the Frank-Wolfe method (see Jaggi (2013); Lacoste-Julien (2016) for recent treatments) to solve the GW problem. Nevertheless, in their implementation, they still need to invoke off-the-shelf linear programming solvers and line-search schemes, which are not well-suited for large-scale applications. Another closely related effort appears in Xu et al. (2019b), which proposes the Bregman projected gradient (BPG) to exactly deal with the GW problem. Unfortunately, BPG encounters the numerical instability issue. To overcome this drawback, Xu et al. (2019a) has developed a simple heuristic modification and presented an attractive empirical performance on the node correspondence task. However, such heuristic still lacks theoretical support and, thus, it is not necessarily guaranteed to perform well (or even converge) under naturally noisy observations.

To bridge the above theoretical-computational gap, we develop provably efficient iterative methods for the GW problem. Specifically, we propose two theoretically solid algorithms tailored to different graph learning tasks. Our first algorithm is the Bregman Alternating Projected Gradient (BAPG), which is also the first provable single-loop algorithm in the GW literature. The core idea behind BAPG is to handle the row and column constraints of the coupling matrix separately. As such, BAPG only involves the matrix-vector/matrix-matrix multiplications and element-wise matrix operations. Thus, BAPG enjoys a variety of nice properties — GPU-friendly implementation, low memory cost, and robustness concerning the step size (i.e., the only

\textsuperscript{1}Stanford University \textsuperscript{2}Hong Kong University of Science and Technology \textsuperscript{3}The Chinese University of Hong Kong \textsuperscript{4}Shanghai University of Finance and Economics.

\textit{Preliminary work.}
hyper-parameter). All these benefits make BAPG amenable to large-scale graph learning tasks. Nonetheless, the iterates generated by BAPG do not necessarily satisfy the Birkhoff polytope constraint and can only reach a critical point of the original GW problem asymptotically. To complement BAPG and avoid such a drawback, we revisit BPG from a fresh perspective and introduce our second algorithm— hybrid Bregman Projected Gradient (hBPG). In particular, we apply eBPG to get a good initial point and then use BPG to get to a critical point. Taking advantage of both eBPG and BPG, the resulting hBPG achieves a great balance between accuracy and efficiency.

Next, we investigate the convergence behavior of the proposed BAPG and hBPG. A key fact here is that the GW problem is a symmetric nonconvex quadratic program with Birkhoff polytope constraints. By fully exploiting this structure, it is interesting to note that the GW problem satisfies the Luo-Tseng error bound condition (Luo & Tseng, 1992), which plays a vital role in our analysis. We first quantify the approximation bound for the fixed-point set of BAPG explicitly and the subsequent convergence result follows. Moreover, we prove the linear convergence result of BPG and hBPG under the established local error bound property. We also explicitly analyze the factor affecting the shrinking rate and rigorously explain why BPG has the aforementioned numerical issue.

As a result of the developed theoretical results and algorithm characteristics, we are able to provide novel insights to help users to select the most suitable algorithm. In fact, BAPG is the best fit for the graph alignment and partition tasks (see Fig.1 (a) and (b) for details), which can sacrifice some feasibility to gain both matching accuracy (i.e., performance measure) and computational efficiency. By contrast, hBPG is more suitable for the shape matching task (i.e., Fig.1 (c)), where one of the quality metrics is the sharpness of the matching coupling. Hence, BAPG would be the suboptimal choice due to its infeasibility issue. In terms of wall-clock time and modeling performance, extensive experiment results have shown that our methods consistently achieve superior performance on all graph learning tasks. It is also well worth noting that all theoretical insights have been well-supported by our experiment results. To sum up, this paper opens up an exciting avenue for realizing the benefits of GW in large-scale graph learning analysis.

2. Proposed Algorithms

In this section, we first formulate the GW distance as a nonconvex quadratic program with polytope constraints. Then, we introduce two proposed algorithms— BAPG and hBPG, and further analyze their computational properties and applicable scenarios.

2.1. Problem Setup

The Gromov-Wasserstein (GW) distance was originally proposed in (Mémoli, 2011; 2014; Peyré et al., 2019) for quantifying the distance between two probability measures supported on heterogeneous metric spaces. More precisely:

**Definition 1 (GW distance).** Suppose that we are given two unregistered compact metric spaces $(\mathcal{X}, d_{\mathcal{X}})$, $(\mathcal{Y}, d_{\mathcal{Y}})$ accompanied with Borel probability measures $\mu, \nu$ respectively. The GW distance between $\mu$ and $\nu$ is defined as

$$\inf_{\pi \in \Pi(\mu, \nu)} \int \int \left| d_{\mathcal{X}}(x, x') - d_{\mathcal{Y}}(y, y') \right|^2 d\pi(x, y) \, d\pi(x', y') ,$$

where $\Pi(\mu, \nu)$ is the set of all probability measures on $\mathcal{X} \times \mathcal{Y}$ with $\mu$ and $\nu$ as marginals.

Intuitively, the GW distance is trying to preserve the isometric structure between two probability measures under the optimal derivation. If a map pairs $x \rightarrow y$ and $x' \rightarrow y'$, then the distance between $x$ and $x'$ is supposed to be close to the distance between $y$ and $y'$. In view of these nice properties, the GW distance acts as a powerful modeling tool in structural data analysis, especially in graph learning; see, e.g., (Vayer et al., 2019; Xu et al., 2019b;a; Solomon et al., 2016; Peyré et al., 2016) and the references therein.

To start with our algorithmic developments, we consider the discrete case for simplicity and practicality, where $\mu$ and $\nu$ are two empirical distributions, i.e., $\mu = \sum_{i=1}^{n} \mu_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} \nu_j \delta_{y_j}$. Then, the GW-distance admits the
following reformulation:

$$\min_{\pi \in \mathbb{R}^{nm}} - \text{Tr}(D_X \pi D_Y \pi^T)$$

s.t.  
\[ \pi 1_m = \mu, \]
\[ \pi^T 1_n = \nu, \]
\[ \pi \geq 0, \]

where \( D_X \) and \( D_Y \) are two symmetric distance matrices. Problem (1) is a nonconvex quadratic program with polytope constraints. Without exploiting the problem-specific structures in our algorithm design, it is not straightforward to realize the benefits of GW distance in large-scale graph learning tasks.

### 2.2. Bregman Alternating Projected Gradient (BAPG)

In this subsection, we present the proposed Bregman alternating projected gradient (BAPG) method, which is the first provable single-loop algorithm tailored to the GW distance computation. To begin with, we consider the compact form for simplicity:

$$\min_{\pi} f(\pi) + g_1(\pi) + g_2(\pi). \quad (2)$$

Here, \( f(\pi) = -\text{Tr}(D_X \pi D_Y \pi^T) \) is a nonconvex quadratic function; \( g_1(\pi) = \mathbb{1}_{(\pi \in C_1)} \) and \( g_2(\pi) = \mathbb{1}_{(\pi \in C_2)} \) are two indicator functions over closed convex polyhedral sets, where \( C_1 = \{ \pi \geq 0 : \pi 1_m = \mu \} \) and \( C_2 = \{ \pi \geq 0 : \pi^T 1_n = \nu \} \).

Arguably, one of the main drawbacks of the double-loop scheme in all existing methods is the computational burden and not GPU friendly. To circumvent this drawback, a starting point to handle \( C_1 \) and \( C_2 \) separately. Otherwise, even if we linearize the objective, (2) will reduce to the vanilla regularized optimal transport problem and hence still rely on some iterative schemes (such as Sinkhorn (Cuturi, 2013)) to tackle it. This motivates us to take the projected gradient descent step in an alternating fashion. Also, by fully detecting the hidden structures, we are able to further benefit from the fact that the Bregman projection for the simplex constraint (e.g., \( C_1 \) and \( C_2 \)) can be extremely efficient computed (Krithche et al., 2015). The above considerations lead to the closed-form updates in each iteration of BAPG:

**BAPG**

\[
\begin{align*}
\pi &\leftarrow \pi \odot \exp(D_X \pi D_Y / \rho), \\
\pi &\leftarrow \text{diag}(\mu / \pi 1_m) \odot \pi, \\
\pi &\leftarrow \pi \odot \exp(D_X \pi D_Y / \rho), \\
\pi &\leftarrow \pi \odot \text{diag}(\nu / \pi^T 1_n),
\end{align*}
\]

where \( \rho \) is the step size and \( \odot \) denotes element-wise (Hadamard) matrix multiplication. BAPG enjoys several nice properties that are extremely attractive for large-scale graph learning tasks. First, BAPG is a single-loop algorithm that only involves matrix-vector/matrix-matrix multiplications and element-wise matrix operations. All these operations are GPU-friendly. Second, different from the entropic regularization parameter in eBPG, BAPG is more robust to the step size \( \rho \) in terms of solution performance. Third, BAPG only involves one memory operation of a large matrix with size \( nm \). Notably, even for large-scale optimal transport problems, the main bottleneck is not floating-point computations, but rather time-consuming memory operations (Mai et al., 2021).

**Theoretical Insights for BAPG**

Now, we start to give the theoretical intuition why BAPG will work well in practice. To better understand the alternating projected scheme developed in (3), we adopt the operator splitting strategy to reformulate (2) as

$$\min_{\pi, w} f(\pi, w) + g_1(\pi) + g_2(\pi)$$

s.t.  
\[ \pi = w, \]

where \( f(\pi, w) = -\text{Tr}(D_X \pi D_Y w^T) \). The BAPG update (3) can be interpreted as processing the alternating minimization scheme on the constructed penalized function, i.e.,

$$F_{\rho}(\pi, w) = f(\pi, w) + g_1(\pi) + g_2(\pi) + \rho D_h(\pi, w).$$

Here, \( D_h(\cdot, \cdot) \) is the so-called Bregman divergence, i.e.,

$$D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle,$$

where \( h(\cdot) \) is the Legendre function, e.g., \( \frac{1}{2} |x|^2 \), relative entropy \( x \log x \), etc. For \( k \)-th iteration, the BAPG update takes the form

$$\begin{align*}
\pi^{k+1} &= \arg\min_{\pi \in C_1} \left\{ -\langle \pi, D_X w^k D_Y \rangle + \rho D_h(\pi, w^k) \right\}, \\
w^{k+1} &= \arg\min_{w \in C_2} \left\{ -\langle w, D_X \pi^{k+1} D_Y \rangle + \rho D_h(w, \pi^{k+1}) \right\}.
\end{align*}\]

When \( h \) is the relative entropy, (3) can be derived from (5).

Similar to the quadratic penalty method (Nocedal & Wright, 2006), BAPG is an infeasible method that can only converge to a critical point of (1) in an asymptotic sense. In other words, if we choose the parameter \( \rho \) as a constant, there is always an infeasibility gap. Fortunately, BAPG is already able to achieve the desirable performance for some large-scale graph learning tasks, especially those that care more about the implementation efficiency and matching accuracy. Recently, Vincent-Cuazer et al. (2021a) has proposed a relaxed version of GW distance for the graph partition task, which further corroborates our investigation that it is acceptable and promising to sacrifice some feasibility to gain other benefits. In Sections 4.2 and 4.3, extensive experiments have been conducted to demonstrate that BAPG has superior performance when compared with other existing baselines on large-scale graph alignment and partition tasks.
At last, we summarize all the aforementioned algorithms in Table 1, which aims to provide a theoretically-supported guideline for readers on how to select the best-fit method based on their task properties.

2.3. Hybrid Bregman Projected Gradient (hBPG)

To remedy the infeasible issue of BAPG, we revisit the Bregman proximal gradient descent (BPG) method, which is a feasible method for addressing the original problem (1) exactly. Such an approach has already been well-explored in some early works (Xu et al., 2019b). For the \( k \)-th iteration, BPG takes the form

\[
P_k = \arg \min_{\pi \in C_1 \cap C_2} \left\{ \nabla f(\pi_k)^T \pi + \frac{1}{t_k} D_h(\pi, \pi_k) \right\},
\]

where \( t_k \) is the chosen step size. The core difficulty here is the need for an inner solver to tackle (6) efficiently. As it turns out, without the entropic regularizer, BPG suffers from the numerical instability issue. It is difficult for the inner solver to achieve the desired accuracy so as to guarantee convergence. Although Xu et al. (2019b) has provided a subsequent convergence result for BPG, it is too weak to guide the user in which scenarios BPG will potentially enjoy notable advantages. Such a state of affairs greatly limits its applicability. Different from graph alignment and partition tasks studied in Xu et al. (2019b:a), BPG-type methods are more attractive for applications that require a sharp matching map (such as shape correspondence), since the approximation (infeasibility) gap will dramatically affect the modeling performance.

To make BPG applicable to large-scale problems, we further exploit its local linear convergence property, see Section 3 for details. Specifically, such a property guides us to take full advantage of both methods — BPG and eBPG. A natural idea is to apply eBPG to get a good initial point and then use BPG to reach critical points. It is reasonable to infer that the resulting hybrid method (denoted by hBPG) will achieve a trade-off between accuracy and efficiency. Comprehensive experiments conducted in Section 4.4 corroborate our theoretical insights.

At last, we summarize the first half of Table 1, which aims to provide a theoretically-supported guideline for readers on how to select the best-fit method based on their task properties.

3. Theoretical Results

In this section, we give all theoretical results developed in this paper, which justify our delivered theoretical intuition and insights in a rigorous manner. To begin with, we introduce the canonical assumption used in the analysis of Bregman proximal-type algorithms (Bauschke & Lewis, 2000; Bauschke et al., 2017).

**Assumption 2** (Bregman divergence function \( h \)).

(a) \( h \) is \( \sigma \)-strongly convex on \( \text{dom}(h) \).

(b) For any compact set \( K \subseteq \text{int} \text{dom}(h) \), we have

\[ \| \nabla h(x) - \nabla h(y) \| \leq L_h(K) \| x - y \| \]

for some constant \( L_h(K) > 0 \).

Our main tool is the following regularity condition:

**Proposition 3** (Luo-Tseng Error Bound Condition for (1)).

There exist scalars \( \epsilon > 0 \) and \( \tau > 0 \) such that

\[ \text{dist}(\pi, \mathcal{X}) \leq \tau \| \pi - \text{proj}_{C_1 \cap C_2}(\pi + D_X \pi D_Y) \|, \]

whenever \( \| \pi - \text{proj}_{C_1 \cap C_2}(\pi + D_X \pi D_Y) \| \leq \epsilon \), where \( \mathcal{X} \) is the critical point set of (2) defined by

\[ \mathcal{X} = \{ \pi \in C_1 \cap C_2 : 0 \in \nabla f(\pi) + N_{C_1}(\pi) + N_{C_2}(\pi) \} \]

and \( N_C(\pi) \) is the normal cone to \( C \) at \( \pi \).

By invoking Theorem 2.3 in (Luo & Tseng, 1992), the error bound condition (7) holds on the feasible set \( C_1 \cap C_2 \). Proposition 3 extends (7) to the whole space \( \mathbb{R}^{n \times m} \). As we shall see later, this error bound condition not only plays an important role in quantifying the approximation bound for the fixed points set of BAPG but also for establishing the local linear convergence rate of BPG.

3.1. Properties of Fixed Points of BAPG

To begin with, we present one crucial lemma that shall be used in studying the approximation bound of BAPG.

**Lemma 4.** Let \( C_1 \) and \( C_2 \) be convex polyhedral sets. There exists a constant \( M > 0 \) such that

\[ \| \text{proj}_{C_1}(x) + \text{proj}_{C_2}(y) - 2 \text{proj}_{C_1 \cap C_2}\left(\frac{x + y}{2}\right) \| \leq M \| \text{proj}_{C_1}(x) - \text{proj}_{C_2}(y) \|. \]

The proof idea follows essentially from the observation that the inequality can be regarded as the stability of the optimal solution for a linear-quadratic problem. Together with Theorem 4.1 in (Zhang & Luo, 2020), the desired result is obtained. Equipped with Lemma 4 and Proposition 3, it is not hard to obtain the following approximation result.

**Proposition 5.** The point \( (\pi^*, w^*) \) belongs to the fixed-point set \( \mathcal{X}_{\text{BAPG}} \) of BAPG if it satisfies

\[ \begin{aligned}
\nabla f(w^*) + \rho (\nabla h(\pi^*) - \nabla h(w^*)) + p &= 0, \\
\nabla f(\pi^*) + \rho (\nabla h(w^*) - \nabla h(\pi^*)) + q &= 0,
\end{aligned} \]

where \( \rho \) is the optimal solution for a linear-quadratic problem. Together with Theorem 4.1 in (Zhang & Luo, 2020), the desired result is obtained. Equipped with Lemma 4 and Proposition 3, it is not hard to obtain the following approximation result.

**Proposition 5.** The point \((\pi^*, w^*)\) belongs to the fixed-point set \( \mathcal{X}_{\text{BAPG}} \) of BAPG if it satisfies

\[ \begin{aligned}
\nabla f(w^*) + \rho (\nabla h(\pi^*) - \nabla h(w^*)) + p &= 0, \\
\nabla f(\pi^*) + \rho (\nabla h(w^*) - \nabla h(\pi^*)) + q &= 0,
\end{aligned} \]
where \( p \in X_{C_1}(\pi^*) \) and \( q \in X_{C_2}(w^*) \). Then, the infeasibility error satisfies \( \| \pi^* - w^* \| \leq \Theta \left( \frac{1}{\rho} \right) \) and the gap between \( h_{\text{BAPG}} \) and \( h \) satisfies
\[
\text{dist} \left( \frac{\pi^* + w^*}{2}, h \right) \leq \Theta \left( \frac{1}{\rho} \right).
\]

If \( \pi^* = w^* \), then (9) is identical to \( h \) and BAPG can reach a critical point of (1). Proposition 5 indicates that as \( \rho \to +\infty \), the infeasibility error term \( \| \pi^* - w^* \| \) shrinks to zero and thus BAPG converges to a critical point of (1) in an asymptotic way. Furthermore, it explicitly quantifies the approximation gap when we select the parameter \( \rho \) as a constant. The proof can be found in Appendix.

### 3.2. Convergence Analysis of BAPG

A further natural question is whether BAPG will converge or not. We answer the question in the affirmative. Specifically, we show that under several canonical assumptions, any limit point of BAPG belongs to \( h_{\text{BAPG}} \). Towards that end, let us first establish the sufficient decrease property of BAPG based on the potential function \( F_p(.) \).

**Proposition 6.** Let \( \{(\pi^k, w^k)\}_{k \geq 0} \) be the sequence generated by BAPG. Suppose that Assumption 2 is satisfied and \( D_h(\cdot, \cdot) \) is symmetric on the whole sequence. Then, we have
\[
F_p(\pi^{k+1}, w^{k+1}) - F_p(\pi^k, w^k) 
\leq -\rho D_h(\pi^k, \pi^{k+1}) - \rho D_h(w^k, w^{k+1}).
\]

As \( F_p(.) \) is coercive, we have
\[
\sum_{k=0}^{\infty} D_h(\pi^k, \pi^{k+1}) + D_h(w^k, w^{k+1}) < +\infty,
\]
and both \( \{D_h(\pi^k, \pi^{k+1})\}_{k \geq 0} \) and \( \{D_h(w^k, w^{k+1})\}_{k \geq 0} \) converge to zero. Thus, the following convergence result holds.

**Theorem 7 (Subsequent Convergence of BAPG).** Any limit point of the sequence \( \{(\pi^k, w^k)\}_{k \geq 0} \) generated by BAPG belongs to \( h_{\text{BAPG}} \).

### 3.3. Linear Convergence of BPG and hBPG

Next, we investigate the convergence behavior of BPG when applied to problem (1). At the heart of our convergence rate analysis is the Luo-Tseng local error bound condition (cf. Proposition 3), which has been demonstrated as a crucial tool to establish the linear convergence rate of first-order algorithms (Zhou & So, 2017). Recall that a sequence \( \{x^k\}_{k \geq 0} \) is said to converge R-linearly (resp. Q-linearly) to a point \( x^\infty \) if there exist constants \( \gamma > 0 \) and \( \eta \in (0, 1) \) such that \( \|x^k - x^\infty\| \leq \gamma \eta^k \) for all \( k \geq 0 \) (resp. if there exists an index \( K \geq 0 \) and a constant \( \eta \in (0, 1) \) such that \( \|x^{k+1} - x^\infty\| / \|x^k - x^\infty\| \leq \eta \) for all \( k \geq K \)).

**Theorem 8 (Local Linear Convergence of BPG).** Suppose that in Problem (1), the step size \( t_k \) in (6) satisfies \( 0 < t \leq t_k < \frac{1}{\sigma} \leq \sigma \|L_f \| \) for \( k \geq 0 \) where \( L_f \) and \( \sigma \) are given constants and \( L_f \) is the gradient Lipschitz constant of \( f \). Moreover, suppose that the sequence \( \{\pi^k\}_{k \geq 0} \) has an element-wise lower bound, i.e., \( \pi_k \geq \epsilon > 0 \) and Assumption 2 holds. Then, the sequence of solutions \( \{\pi^k\}_{k \geq 0} \) generated by BPG converges R-linearly to an element in the critical point set \( X \).

To analyze factors affecting the speed of the linear convergence rate, we provide the retraction inequality of the function value sequence
\[
F(\pi^{k+1}) - F(\pi^*) \leq \frac{\kappa_2(\tau \kappa_3 + 1)}{\kappa_1 + \kappa_2(\tau \kappa_3 + 1)} (F(\pi^k) - F(\pi^*)),
\]
where \( \pi^* \) is the limit point of the sequence \( \{\pi^k\}_{k \geq 0} \) generated by BPG,
\[
\kappa_1 = \frac{1}{\frac{L_f}{\sigma}},
\]
\[
\kappa_2 = \max \left( \left( \frac{L_f}{\frac{L_f}{\tau} + 3L_f}, \frac{7L_f}{2} \right) \right),
\]
\[
\kappa_3 = \left( \frac{2L_f + \frac{L_f(K)}{\frac{L_f}{\tau} + 2}}{\frac{L_f}{\tau}} \right).
\]
Here, \( L = \max \left( L_{\text{F}}(K) \cdot \frac{4M}{\epsilon L_f} \right) \), where \( K = C_1 \cap C_2 \cap \{ \pi : \pi \geq \epsilon I_{nxm} \} \) and \( M \) is a given constant.

**Remark 9.** We consider the most relevant case — using the relative entropy as the Legendre function \( h \). Then, we can compute the Lipschitz constant as \( L_{\text{h}}(K) \propto \frac{1}{\tau^2} \). The shrinking rate in Theorem 8 will go to 1 when \( \epsilon \) goes to 0. It further explains why BPG is still time-consuming if \( \epsilon \) goes to zero. However, due to the lack of the entropic regularization term, the GW problem enforces the solution to be sparse, which results in numerical instability issue.

For the sake of brevity, we omit the proof. We refer the reader to Appendix for further details.

### 4. Experiment Results

In this section, we provide extensive experiment results to validate the effectiveness of the proposed BAPG and hBPG algorithms on various representative graph learning tasks, including graph alignment, graph partition, and shape matching. All simulations are implemented using Python 3.8 on a high-performance computing server running Ubuntu 20.04 with an Intel(R) Xeon(R) Gold 6226R CPU and an NVIDIA GeForce RTX 3090 GPU.

#### 4.1. Toy 2D Matching Problem

We investigate a toy matching problem in 2D to corroborate our theoretical insights and results in Section 2 and 3. Fig. 2 shows an example of mapping a two-dimensional shape
Graph alignment aims to identify the node correspondence between two graphs possibly with different topology structures (Zhang et al., 2021; Chen et al., 2020). Instead of solving the restricted quadratic assignment problem (Lawler, 1963; Lacoste-Julien et al., 2006), the GW distance provides the optimal probabilistic correspondence relationship via preservation of the isometric property. Here, we compare the proposed BAPG with all existing baselines: BPG, GWL (Xu et al., 2019b), ScalaGWL (Xu et al., 2019a), SpecGWL (Chowdhury & Needham, 2021), and eBPG (Solomon et al., 2016). Except for BPG and eBPG, others are pure heuristic methods without any theoretical guarantee. **Parameters Setup** We utilize the unweighted symmetric adjacent matrices as our input distance matrices, i.e., $D_X$ and $D_Y$. Alternatively, SpecGWL uses the heat kernel $\exp(-L)$ where $L$ is the normalized graph Laplacian matrix. We set both $\mu$ and $\nu$ to be the uniform distribution. For three heuristic methods — GWL, ScalaGWL, and SpecGWL, we follow the same setup reported in their papers. As mentioned, eBPG is very sensitive to the entropic regularization parameter. To get comparable results, we report the best result among the set \{0.1, 0.01, 0.001\} of the regularization parameter. For BPG and BAPG, we use the fixed step size $\lambda = 5$ and $\rho = 0.1$ respectively.

**Database Statistics** We test all methods on both synthetic and real-world databases. Our setup for the synthetic database is the same as in Xu et al. (2019b). The source graph $G_s = \{V_s, E_s\}$ is generated by two ideal random models, Gaussian random partition and Barabasi-Albert models, with different scales, i.e., $|V_s|\in\{500, 1000, 1500, 2000, 2500\}$. Then, we generate the target graph by adding more than $q\%$ nodes and edges on top of the source graph, where $q\in\{0, 10, 20, 30, 40, 50\}$. For each setup, we generate 5 synthetic graph pairs over random seeds. To sum up, the synthetic database contains 300 different graph pairs. We also validate our proposed methods on other three real-world databases from (Chowdhury & Needham, 2021), including two biological graph databases Proteins and Enzymes, and a social network database Reddit. Furthermore, to demonstrate the robustness of our method regarding the noise level, we follow the noise generating process (i.e., $q = 10\%$) conducted for the synthesis case to create new databases on top of the three real-world databases. Towards that end, the statistics of all databases used for the graph alignment task have been summarized in Table 4.

Table 2 shows the comparison of matching accuracy and
Table 2. Comparison of the matching accuracy (%) and wall-clock time (seconds) on graph alignment; '*' denotes the computational time accelerated by GPU. For the CPU time, BAPG costs 9026s on Synthetic and 780s on Reddit.

| Method   | Synthetic Acc | Time | Proteins Raw | Noisy | Time | Enzymes Raw | Noisy | Time | Reddit Raw | Noisy | Time |
|----------|---------------|------|--------------|-------|------|-------------|-------|------|-------------|-------|------|
| GWL      | 61.48         | 22600| 71.74        | 52.74 | 40.4 | 79.25       | 62.21 | 13.4 | 39.04       | 36.68 | 1431 |
| ScalaGWL | 17.93         | 12002| 16.37        | 16.05 | 372.2| 12.72       | 11.46 | 213.0| 0.54        | 0.70  | 1109 |
| SpecGWL  | 13.27         | 1462 | 78.11        | 19.31 | 30.7 | 79.07       | 21.14 | 6.7  | 50.71       | 19.66 | 1074 |
| eBPG     | 12.42         | 9502 | 67.48        | 45.85 | 208.2| 78.25       | 60.46 | 499.7| 3.76        | 3.34  | 1234 |
| BPG      | 88.29         | *704*| 78.18        | 57.16 | 59.1 | 79.66       | 62.85 | 14.8 | 50.83       | 49.45 | 115* |

Table 3. Comparison of AMI scores on graph partition datasets; Spectral denotes the heat kernel matrices.

| Matrices | Method | Wikipedia Raw | Noisy | EU-email Raw | Noisy | Amazon Raw | Noisy | Village Raw | Noisy |
|----------|--------|---------------|-------|--------------|-------|------------|-------|-------------|-------|
| Adjacent | GWL    | 0.312         | 0.285 | 0.451        | 0.443 | 0.352      | 0.606 | 0.560       |
|          | eBPG   | 0.461         | **0.413** | **0.517** | 0.422 | 0.429      | 0.387 | 0.703       | 0.658 |
|          | BPG    | 0.367         | 0.333 | 0.478        | 0.414 | 0.412      | 0.368 | 0.642       | 0.575 |
|          | BAPG   | **0.468**     | 0.385 | 0.508        | **0.428** | 0.436      | **0.426** | **0.709** | **0.681** |
| Spectral | SpecGWL| 0.442         | 0.395 | 0.487        | 0.425 | 0.565      | 0.487 | 0.758       | 0.707 |
|          | eBPG   | 0.000         | 0.000 | 0.000        | 0.000 | 0.000      | 0.000 | 0.000       | 0.000 |
|          | BPG    | 0.405         | 0.373 | 0.473        | 0.253 | 0.492      | 0.436 | 0.705       | 0.619 |
|          | BAPG   | **0.529**     | **0.397** | **0.533** | **0.436** | **0.609** | **0.505** | **0.797** | **0.711** |

Table 4. Statistics of databases for graph alignment

| Dataset   | # Samples | Ave. Nodes | Ave. Edges |
|-----------|-----------|------------|------------|
| Synthetic | 300       | 1500       | 56579      |
| Proteins  | 1113      | 39.06      | 72.82      |
| Enzymes   | 600       | 32.63      | 62.14      |
| Reddit    | 500       | 375.9      | 449.3      |

wall-clock time on four databases. We observe that BAPG works exceptionally well both in terms of computational time and accuracy, especially for two large-scale noisy graph databases Synthetic and Reddit. As we mentioned in Section 2, the effectiveness of GPU acceleration for BAPG is also well corroborated. We remark that the performance of eBPG and ScalaGWL are influenced by the entropic regularization parameter and approximation error respectively, which accounts for their poor performance. Moreover, it is easy to observe that SpecGWL works pretty well on the small dataset but the performance degrades dramatically on the large one, e.g., synthetic. The reason is that SpecGWL relies on a linear programming solver as its subroutine, which is not well-suited for large-scale settings. Besides, although ScalaGWL has the lowest per-iteration computational complexity, the recursive K-partition mechanism developed in (Xu et al., 2019a) is not friendly to parallel computing. Therefore, ScalaGWL does not demonstrate attractive performance on multi-core processors.

We also report the sensitivity analysis of BAPG regarding the noise level q% and the step size ρ in Fig.3 on the Synthetic Database. Surprisingly, the solution performance of BAPG is robust to the noise level but instead, the accuracy of other methods degrades dramatically as the noise level increases. Beside, different convergence curves of ρ in Fig.3 (b) corroborate Proposition 5 empirically and further guide us on how to choose the step size.

4.3. Graph Partition

The GW distance can also be potentially applied on the graph partition task. That is, we are trying to match the source graph with a disconnected target graph having K isolated and self-connected super nodes, where K is the
Table 5. Comparison of the infeasibility error (i.e., $\frac{\log \tau_{m,n} + \log \tau_{n,m}^*}{m}$) and CPU wall-clock time. For BAPG, we also report the time accelerated by GPU implementation.

| Method    | Hand Error | Hand Time | Octopus Error | Octopus Time | Mug Error | Mug Time | Chair Error | Chair Time | Human Error | Human Time |
|-----------|------------|-----------|---------------|--------------|-----------|---------|-------------|------------|-------------|------------|
| eBPG      | 2.95e-10   | 9.37      | 7.21e-10      | 12.08        | 6.41e-10  | 27.86   | 1.77e-10    | 12.87      | 5.52e-10    | 11.46      |
| BPG       | 3.00e-07   | 389.72    | 2.00e-07      | 32.55        | 3.50e-07  | 196.93  | 4.00e-07    | 304.98     | 2.53e-07    | 93.27      |
| hBPG      | 3.00e-07   | 193.85    | 2.00e-07      | 8.98         | 3.50e-07  | 90.59   | 4.00e-07    | 189.41     | 2.53e-07    | 53.29      |
| BAPG      | 4.54e-06   | 61.77     | 2.14e-05      | 6.19         | 6.62e-05  | 30.83   | 2.13e-05    | 127.78     | 5.62e-05    | 8.26       |
| BAPG-GPU  | -          | 3.10      | -             | 1.28         | -         | 1.39    | -           | 3.22       | -           | 0.78       |

Figure 4. Visualization of the matching results: colored points (source surface) are mapped to colored distributions on the target object.

4.4. Shape Correspondence

Finally, we evaluate the matching performance and computational cost of our proposed hBPG and BAPG on five 3D triangle mesh datasets used in (Solomon et al., 2016).

Parameters Setup By making use of gptoolbox (Jacobson et al., 2018), $D_X$ and $D_Y$ are constructed by computing the geodesic distances over the triangle mesh. Here, $\mu$ and $\nu$ are discrete uniform distributions. For BAPG, we attempt $\rho \in \{0.05, 0.1, 0.2, 0.8\}$ for the fixed step size and report the best result. For BPG and hBPG, we choose the fixed step size $t_k \in \{30, 50, 100, 1000\}$ and report the best result. For eBPG, we use the same setup in (Solomon et al., 2016).

Different from graph alignment and partition, the shape matching task cares more about the sharpness of the correspondence relationship. To visualize this soft matching result, we label several color points on the source surface and then quantify the sharpness of each mapping distribution via the size of the colored area on the target surface. In fact, the smaller area indicates a sharper mapping. Regarding the accuracy and sharpness, the superiority of hBPG and BPG over other methods are obviously observed from Fig.4.

If we further take the computational burden into account, hBPG is the best-fit for this task, see Table 5 for details. hBPG can achieve a great trade-off between efficiency and accuracy. Although BAPG shows attractive advantages in terms of computational cost, it suffers from the infeasibility issue, which is also corroborated by the experiment results in Table 5. The visualization results of the other four 3D mesh objects are given in Appendix.

5. Conclusion

In this paper, we conduct a systematic investigation on developing provably efficient iterative methods for the Gromov-Wasserstein distance computation. Two theoretically solid algorithms, called BAPG and hBPG, have been proposed to tackle different graph learning tasks. Our theoretical results provide novel insights to help users to select the well-suited algorithm for their tasks. A natural future direction is to consider sparse and low-rank structures of the matching matrix to decrease the per-iteration cost and further speed up our methods (Scetbon et al., 2021).
References

Abrishami, T., Guillen, N., Rule, P., Schutzman, Z., Solomon, J., Weighill, T., and Wu, S. Geometry of graph partitions via optimal transport. *SIAM Journal on Scientific Computing*, 42(5):A3340–A3366, 2020.

Attouch, H., Bolte, J., Redont, P., and Soubeyran, A. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-lojasiewicz inequality. *Mathematics of operations research*, 35(2):438–457, 2010.

Attouch, H., Bolte, J., and Svaister, B. F. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss-seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

Bauschke, H. H. *Projection Algorithms and Monotone Operators*. PhD thesis, Simon Fraser University, 1996.

Bauschke, H. H. and Lewis, A. S. Dykstras algorithm with bregman projections: A convergence proof. *Optimization*, 48(4):409–427, 2000.

Bauschke, H. H., Bolte, J., and Teboulle, M. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Bunne, C., Alvarez-Melis, D., Krause, A., and Jegelka, S. Learning generative models across incomparable spaces. In *International Conference on Machine Learning*. pp. 851–861. PMLR, 2019.

Chen, X., Heimann, M., Vahedian, F., and Koutra, D. Consistent network alignment via proximity-preserving node embedding. arXiv preprint arXiv:2005.04725, 2020.

Chowdhury, S. and Méromi, F. The gromov–wasserstein distance between networks and stable network invariants. *Information and Inference: A Journal of the IMA*, 8(4):757–787, 2019.

Chowdhury, S. and Needham, T. Generalized spectral clustering via gromov-wasserstein learning. In *International Conference on Artificial Intelligence and Statistics*. pp. 712–720. PMLR, 2021.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Gao, J., Huang, X., and Li, J. Unsupervised graph alignment with wasserstein distance discriminator. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. pp. 426–435, 2021.

Hou, K., Zhou, Z., So, A. M.-C., and Luo, Z.-Q. On the linear convergence of the proximal gradient method for trace norm regularization. In *NIPS*. pp. 710–718, 2013.

Jacobson, A. et al. gptoolbox: Geometry processing toolbox. ONLINE: http://github.com/alecjacobson/gptoolbox, 2018.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*. pp. 427–435. PMLR, 2013.

Krichene, W., Krichene, S., and Bayen, A. Efficient bregman projections onto the simplex. In 2015 54th IEEE Conference on Decision and Control (CDC), pp. 3291–3298. IEEE, 2015.

Lacoste-Julien, S. Convergence rate of frank-wolfe for non-convex objectives. arXiv preprint arXiv:1607.00345, 2016.

Lacoste-Julien, S., Taskar, B., Klein, D., and Jordan, M. Word alignment via quadratic assignment. 2006.

Lawler, E. L. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.

Li, G. and Pong, T. K. Calculus of the exponent of Kurdyka-Lojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of Computational Mathematics*, 18(5):1199–1232, 2018.

Luo, Z.-Q. and Tseng, P. Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem. *SIAM Journal on Optimization*, 2(1):43–54, 1992.

Mai, V. V., Lindbäck, J., and Johansson, M. A fast and accurate splitting method for optimal transport: Analysis and implementation. arXiv preprint arXiv:2110.11738, 2021.

Méromi, F. Spectral gromov–wasserstein distances for shape matching. In 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pp. 256–263. IEEE, 2009.

Méromi, F. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

Méromi, F. The gromov–wasserstein distance: A brief overview. *Axioms*, 3(3):335–341, 2014.

Gao, J., Huang, X., and Li, J. Unsupervised graph alignment with wasserstein distance discriminator. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. pp. 426–435, 2021.

Hou, K., Zhou, Z., So, A. M.-C., and Luo, Z.-Q. On the linear convergence of the proximal gradient method for trace norm regularization. In *NIPS*. pp. 710–718, 2013.
Mémoli, F. and Sapiro, G. Comparing point clouds. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 32–40, 2004.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Peyré, G., Cuturi, M., and Solomon, J. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pp. 2664–2672. PMLR, 2016.

Peyré, G., Cuturi, M., et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Scetbon, M., Peyré, G., and Cuturi, M. Linear-time gromov wasserstein distances using low rank couplings and costs. *arXiv preprint arXiv:2106.01128*, 2021.

Solomon, J., Peyré, G., Kim, V. G., and Sra, S. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)*, 35(4):1–13, 2016.

Titouan, V., Courty, N., Tavenard, R., and Flamary, R. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pp. 6275–6284. PMLR, 2019.

Vayer, T., Chapel, L., Flamary, R., Tavenard, R., and Courty, N. Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties. *arXiv preprint arXiv:1811.02834*, 2018.

Vayer, T., Flamary, R., Tavenard, R., Chapel, L., and Courty, N. Sliced gromov-wasserstein. In *NeurIPS 2019-Thirty-third Conference on Neural Information Processing Systems*, volume 32, 2019.

Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. Semi-relaxed gromov wasserstein divergence with applications on graphs. *arXiv preprint arXiv:2110.02753*, 2021a.

Vincent-Cuaz, C., Vayer, T., Flamary, R., Corneli, M., and Courty, N. Online graph dictionary learning. *arXiv preprint arXiv:2102.06555*, 2021b.

Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.

Xu, H., Luo, D., and Carin, L. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32:3052–3062, 2019a.

Xu, H., Luo, D., Zha, H., and Duke, L. C. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pp. 6932–6941. PMLR, 2019b.

Xu, H., Luo, D., Carin, L., and Zha, H. Learning graphons via structured gromov-wasserstein barycenters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10505–10513, 2021.

Zhang, J. and Luo, Z. A global dual error bound and its application to the analysis of linearly constrained non-convex optimization. *arXiv preprint arXiv:2006.16440*, 2020.

Zhang, S., Tong, H., Jin, L., Xia, Y., and Guo, Y. Balancing consistency and disparity in network alignment. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2212–2222, 2021.

Zhou, Z. and So, A. M.-C. A unified approach to error bounds for structured convex optimization problems. *Mathematical Programming*, 165(2):689–728, 2017.
Supplementary of “Fast and Provably Convergent Algorithms for Gromov-Wasserstein in Graph Learning”

Problem Setup (1)

In this paper, we consider the discrete case, where only $m, n$ samples are drawn from real distributions, i.e., $\mu = \sum_{i=1}^{n} \mu_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} \nu_j \delta_{y_j}$. The Gromov-Wasserstein distance between $\mu$ and $\nu$ can be recast into the following minimization problem:

$$\min_{\pi \in \Pi(\mu, \nu)} \sum_{i=1}^{n} \sum_{j=1}^{m} |d_X(x_i, x_j) - d_Y(y_k, y_l)|^2 \pi_{ik} \pi_{jl} \tag{1}$$

where $\Pi(\mu, \nu)$ is a generalized Birkhoff polytope, i.e., $\Pi(\mu, \nu) = \{ \pi \in \mathbb{R}^{n \times m} | \pi_1 \mu = \mu, \pi^T 1 = \nu \}$. To further simplify the problem and detect the hidden structures, we reformulate it as a compact form,

$$\min_{\pi \in \Pi(\mu, \nu)} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} (D_X(i, j) - D_Y(k, l))^2 \pi_{ik} \pi_{jl}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} (D_X^2(i, j) + D_Y^2(k, l)) \pi_{ik} \pi_{jl} - 2 \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} D_X(i, j) D_Y(k, l) \pi_{ik} \pi_{jl}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} D_X^2(i, j) \mu_i \mu_j + \sum_{k=1}^{m} \sum_{l=1}^{m} D_Y^2(k, l) \nu_k \nu_l - 2 \text{Tr}(D_X^T D_Y^T \pi^T).$$

where $D_X(i, j) = d_X(x_i, x_j)$ and $D_Y(k, l) = d_Y(y_k, y_l)$ for $i, j \in [n]$ and $k, l \in [m]$. Thus, (1) is obtained.

Proof of Proposition 3 — Luo-Tseng Error Bound Condition for the GW Problem (1)

Proof. $D_x$ and $D_Y$ are two symmetric matrices and $C_1 \cap C_2$ is a convex polyhedral set. By invoking Theorem 2.3 in (Luo & Tseng, 1992), the Luo-Tseng local error bound condition (7) just holds for the feasible set $C_1 \cap C_2$. That is,

$$\text{dist}(\pi, \mathcal{X}) \leq \tau \left\| \pi - \text{proj}_{C_1 \cap C_2}(\pi + D_X \pi D_Y) \right\|,$$

where $\pi \in C_1 \cap C_2$. Then, we aim at extending (11) to the whole space. Define $\hat{\pi} = \text{proj}_{C_1 \cap C_2}(\pi)$, we have

$$\text{dist}(\pi, \mathcal{X}) \leq \left\| \pi - \hat{\pi} \right\| + d(\hat{\pi}, \mathcal{X})$$

$$\leq \left\| \pi - \hat{\pi} \right\| + \tau \left\| \hat{\pi} - \text{proj}_{C_1 \cap C_2}(\pi + D_X \hat{\pi} D_Y) \right\|$$

$$\leq \left\| \pi - \hat{\pi} \right\| + \tau \left\| \text{proj}_{C_1 \cap C_2}(\pi + D_X \hat{\pi} D_Y) - \pi \right\|$$

$$\leq \left\| \pi - \hat{\pi} \right\| + \tau \left\| \hat{\pi} - \pi \right\|$$

By letting $\tau = \tau_{\sigma_{\max}(D_X \sigma_{\max}(D_Y))}$, we get the desired result. \qed
Convergence Analysis of BAPG

Assumption 10. The critical point set $X$ is non-empty.

Definition 11 (Bregman Divergence). We define the proximity measure $D_h : \text{dom}(h) \times \text{int}(\text{dom}(h)) \rightarrow \mathbb{R}_+$

$$D_h(x, y) := h(x) - h(y) - \langle \nabla h(y), x - y \rangle$$

The proximity measure $D_h$ is the so-called Bregman Distance. It measures the proximity between $x$ and $y$. Indeed, thanks to the gradient inequality, one has $h$ is convex if and only if $D_h(x, y) \geq 0, \forall x \in \text{dom} h, y \in \text{int} \text{dom} h$.

Concrete Examples

- Relative entropy, or Kullback–Leibler divergence, $h(x) = \sum_i x_i \log x_i$ and the Bregman divergence is given as

$$D_h(x, y) = \sum_i x_i \log \frac{x_i}{y_i}.$$

- Quadratic function: $h(x) = \frac{1}{2} \|x\|^2$. As such, the Bregman divergence $D_h(x, y)$ will reduce to the Euclidean distance.

For the completeness of our algorithmic development, we also give the details how the general BAPG (5) update implies (3) if we choose the Legendre function as relative entropy.

$$\pi^{k+1} = \arg \min_{\pi \in C_1} \left\{ -\langle \pi, D_Y w^k D_Y \rangle + \rho D_h(\pi, w^k) \right\}$$

$$= \arg \min_{\pi \in C_1} D_h(\pi, w^k \exp \left( \frac{D_X w^k D_Y}{\rho} \right)),

= \text{diag}(\mu / P^k m) \odot P^k,$$

where $P^k = w^k \exp \left( \frac{D_X w^k D_Y}{\rho} \right)$.

$$w^{k+1} = \arg \min_{w \in C_2} \left\{ -\langle w, D_X \pi^{k+1} D_Y \rangle + \rho D_h(w, \pi^{k+1}) \right\}$$

$$= \arg \min_{w \in C_2} D_h(w, \pi^{k+1} \exp \left( \frac{D_X \pi^{k+1} D_Y}{\rho} \right)),

= Z^k \odot \text{diag}(\nu / Z^{kT} 1_n),$$

where $Z^k = \pi^{k+1} \exp \left( \frac{D_X \pi^{k+1} D_Y}{\rho} \right)$.

The optimality conditions for each iteration follows,

$$\begin{cases}
0 \in -D_X w^k D_Y + \rho(\nabla h(\pi^{k+1}) - \nabla h(w^k)) + N_{C_1}(\pi^{k+1}), \\
0 \in -D_X \pi^{k+1} D_Y + \rho(\nabla h(w^{k+1}) - \nabla h(\pi^{k+1})) + N_{C_2}(w^{k+1}).
\end{cases} \quad (12)$$

Proof of Lemma 4

Proof. To prove

$$\|\text{proj}_{C_1}(x) + \text{proj}_{C_2}(y) - 2 \text{proj}_{C_1 \cap C_2} \left( \frac{x + y}{2} \right)\| \leq M \|\text{proj}_{C_1}(x) - \text{proj}_{C_2}(y)\|, \quad (13)$$

we first convert the left-hand side of (13) to

$$\|\text{proj}_{C_1}(x) - \text{proj}_{C_1 \cap C_2} \left( \frac{x + y}{2} \right) + \text{proj}_{C_2}(y) - \text{proj}_{C_1 \cap C_2} \left( \frac{x + y}{2} \right)\|.$$
When we can bound the distance between two optimal solutions by the perturbation quantity, which is indeed the right hand side of (13). By invoking Theorem 4.1 in (Zhang & Luo, 2020), we can bound the distance between two optimal solutions by the perturbation quantity r, i.e.,

$$
\| (p(0), q(0)) - (p(r), q(r)) \| \leq M \| r \|.
$$

**Proof of Proposition 5 — Approximation bound of fixed points of BAPG**

**Definition 12 (Bounded Linear Regularity (cf. Definition 5.6 in (Bauschke, 1996))).** Let $C_1, \ldots, C_N$ be closed convex subsets of $\mathbb{R}^n$ with a non-empty intersection $C$. We say that the collection $\{C_1, \ldots, C_N\}$ is bounded linear regular (BLR) if for every bounded subset $\mathcal{B}$ of $\mathbb{R}^n$, there exists a constant $\kappa > 0$ such that

$$
d(x, C) \leq \kappa \max_{i \in \{1, \ldots, N\}} d(x, C_i), \text{ for all } x \in \mathcal{B}.
$$

If all of $C_i$ are polyhedral sets, BLR condition will be automatically hold.

**Proof.** Recall the fixed-point set of BAPG:

$$
\mathcal{X}_{BAPG} = \left\{ (\pi^*, w^*) : \nabla f(w^*) + \rho(\nabla h(\pi^*) - \nabla h(w^*)) + p = 0, p \in NC_1(\pi^*) \right\}.
$$

Define $\hat{\pi} = \text{proj}_{C_1 \cap C_2}(\pi^*)$, we first want to argue the following inequality holds,

$$
\| \hat{\pi} - \pi^* \| + \| \pi^* - w^* \| \leq (2\kappa + 1)\| \pi^* - w^* \|.
$$

As the BLR condition is satisfied for the polyhedral constraint, see Definition 12 for details, we have

$$
\| \hat{\pi} - \pi^* \| + \| \hat{\pi} - w^* \| \leq 2\| \hat{\pi} - \pi^* \| + \| \pi^* - w^* \|
$$

$$
= 2 \text{dist}(\pi^*, C_1 \cap C_2) + \| \pi^* - w^* \|
$$

$$
\leq (2\kappa + 1)\| \pi^* - w^* \|.
$$

Based on the stationary points defined in (9), we have,

$$
\nabla f(w^*)^T(\hat{\pi} - \pi^*) + \rho(\nabla h(\pi^*) - \nabla h(w^*))^T(\hat{\pi} - \pi^*) + p^T(\hat{\pi} - \pi^*) = 0, p \in NC_1(\pi^*),
$$

$$
\nabla f(\pi^*)^T(\hat{\pi} - w^*) + \rho(\nabla h(w^*) - \nabla h(\pi^*))^T(\hat{\pi} - w^*) + q^T(\hat{\pi} - w^*) = 0, q \in NC_2(\pi^*).
$$

Summing up the above two equations,

$$
\nabla f(w^*)^T(\hat{\pi} - \pi^*) + \nabla f(\pi^*)^T(\hat{\pi} - w^*) + \rho(\nabla h(\pi^*) - \nabla h(w^*))^T(w^* - \pi^*) + p^T(\hat{\pi} - \pi^*) + q^T(\hat{\pi} - w^*) = 0
$$

$$
\Rightarrow \nabla f(w^*)^T(\hat{\pi} - \pi^*) + \nabla f(\pi^*)^T(\hat{\pi} - w^*) - \rho(D_h(\pi^*, w^*) + D_h(w^*, \pi^*)) + p^T(\hat{\pi} - \pi^*) + q^T(\hat{\pi} - w^*) = 0
$$

$$
\Rightarrow \nabla f(w^*)^T(\hat{\pi} - \pi^*) + \nabla f(\pi^*)^T(\hat{\pi} - w^*) - \rho(D_h(\pi^*, w^*) + D_h(w^*, \pi^*)) \geq 0,
$$
where (●) holds as \((\nabla h(x) - \nabla h(y))^T(x - y) = D_h(x, y) + D_h(y, x)\) and (●) follows from the property of the normal cone. For instance, since \(q \in \mathcal{N}_{C_1}(\pi)\), we have \(q^T(\hat{\pi} - \pi^*) \leq 0\) (i.e., \(\hat{\pi} \in C_1\)). Therefore,
\[
\rho(D_h(\pi^*, w^*) + D_h(w^*, \pi^*)) \leq \nabla f(w^*)^T(\hat{\pi} - \pi^*) + \nabla f(\pi^*)^T(\hat{\pi} - w^*) \\
\leq \|\nabla f(w^*)\|\|\hat{\pi} - \pi^*\| + \|\nabla f(\pi^*)\|\|\hat{\pi} - w^*\| \\
\leq L_f(\|\hat{\pi} - \pi^*\| + \|\hat{\pi} - w^*\|) \\
\leq (2\kappa + 1)L_f\|\pi^* - w^*\|.
\]

The next to last inequality holds as \(f(\cdot)\) is a quadratic function and the effective domain \(C_1 \cap C_2\) is bounded. Thus, the norm of its gradient will naturally have a constant upper bound, i.e., \(L_f = \sigma_{\max}(D_X)\sigma_{\max}(D_Y)\). As \(h\) is a \(\sigma\)-strongly convex, we have
\[
D_h(\pi^*, w^*) \geq \frac{\sigma}{2}\|\pi^* - w^*\|^2.
\]
Together with this property, we can quantify the infeasibility error \(\|\pi^* - w^*\|\),
\[
\|\pi^* - w^*\| \leq \frac{(2\kappa + 1)L_f}{\delta\rho}.
\] (17)

When \(\rho \rightarrow +\infty\), it is easy to observe that the infeasibility error term \(\|\pi^* - w^*\|\) will shrink to zero. More importantly, if \(\pi^* = w^*\), then \(X_{BAPG}\) will be identical to \(\mathcal{X}\).

Next, we target at quantifying the approximation gap between the fixed-point set of BAPG and the critical point set of the original problem (1). Upon (15), we have
\[
\nabla f\left(\frac{\pi^* + w^*}{2}\right) + \frac{p + q}{2} = 0,
\]
where \(\nabla f(\cdot)\) is a linear operator. By applying the Luo-Tseng local error bound condition of (1), i.e., Proposition 3, we have
\[
\text{dist}\left(\frac{\pi^* + w^*}{2}, \mathcal{X}\right) \leq \frac{\tau}{2}\left\|\frac{\pi^* + w^*}{2} - \text{proj}_{C_1 \cap C_2}\left(\frac{\pi^* + w^*}{2} - \nabla f\left(\frac{\pi^* + w^*}{2}\right)\right)\right\| \\
\leq \frac{\tau}{2}\left\|\frac{\pi^* + w^*}{2} - \text{proj}_{C_1 \cap C_2}\left(\frac{\pi^* + p + w^* + q}{2}\right)\right\| \\
\leq \frac{\tau}{2}\left\|\text{proj}_{C_1}(\pi^* + p) + \text{proj}_{C_2}(w^* + q) - 2\text{proj}_{C_1 \cap C_2}\left(\frac{\pi^* + p + w^* + q}{2}\right)\right\| \\
\leq M\frac{\|\pi^* - w^*\|}{2},
\]
where (●) holds due to the normal cone property, that is, \(\text{proj}_{C_1}(x + z) = x\) for any \(x \in C\) and \(z \in \mathcal{N}_C(x)\), and (●) follows from Lemma 4. Incorporating with (17), the approximation bound for BAPG has been characterized quantitatively, i.e.,
\[
\text{dist}\left(\frac{\pi^* + w^*}{2}, \mathcal{X}\right) \leq \frac{(2\kappa + 1)L_fM}{2\delta\rho}.
\]

\[\square\]

**Proof of Proposition 6 — Sufficient decrease property of BAPG**

**Proof.** We first observe from the optimality conditions of the the main updates, i.e.,
\[
0 \in \nabla_x f(\pi^k, w^k) + \rho(\nabla h(\pi^{k+1}) - \nabla h(w^k)) + \partial g_1(\pi^{k+1}) \\
0 \in \nabla_w f(\pi^{k+1}, w^k) + \rho(\nabla h(w^{k+1}) - \nabla h(\pi^{k+1})) + \partial g_2(w^{k+1})
\] (18) (19)
where \(g_1(\pi) = I_{\pi \in C_1}\) and \(g_2(w) = I_{w \in C_2}\). Due to the convexity of \(g_1(\cdot)\), it is natural to obtain,
\[
g_1(\pi^k) - g_1(\pi^{k+1}) \geq -\langle \nabla f(\pi^k, w^k) + \rho(\nabla h(\pi^{k+1}) - \nabla h(w^k)), \pi^k - \pi^{k+1}\rangle \\
= -\langle \nabla f(\pi^k, w^k), \pi^k - \pi^{k+1}\rangle + \langle \rho(\nabla h(\pi^{k+1}) - \nabla h(w^k)), \pi^{k+1} - \pi^k\rangle
\]
As \( f(\pi, w) = -\text{tr}(D_X^T \pi D_Y w^T) \) is a bilinear function, we have,
\[
f(\pi^k, w^k) - f(\pi^{k+1}, w^k) - \langle \nabla_\pi f(\pi^k, w^k), \pi^k - \pi^{k+1} \rangle = 0.
\]
Consequently, we get
\[
f(\pi^k, w^k) + g_1(\pi^k) - f(\pi^{k+1}, w^k) - g_1(\pi^{k+1}) \geq \rho(\nabla h(\pi^{k+1}) - \nabla h(w^k), \pi^{k+1} - \pi^k).
\] (20)
Similarly, based on the \( w \)-update, we obtain
\[
f(\pi^{k+1}, w^k) + g_2(w^k) - f(\pi^{k+1}, w^{k+1}) - g_2(w^{k+1}) \geq \rho(\nabla h(w^{k+1}) - \nabla h(\pi^{k+1}), w^{k+1} - w^k).
\] (21)
Combine with (20) and (21), we obtain
\[
f(\pi^k, w^k) + g_1(\pi^k) + g_2(w^k) - f(\pi^{k+1}, w^k) - g_1(\pi^{k+1}) - g_2(w^{k+1})
\geq \rho(\nabla h(\pi^{k+1}) - \nabla h(w^k), \pi^{k+1} - \pi^k) + \rho(\nabla h(w^{k+1}) - \nabla h(\pi^{k+1}), w^{k+1} - w^k).
\] (22)
The right-hand side can be further simplified,
\[
\rho(\nabla h(\pi^{k+1}) - \nabla h(w^k), \pi^{k+1} - \pi^k) + \rho(\nabla h(w^{k+1}) - \nabla h(\pi^{k+1}), w^{k+1} - w^k)
\geq \rho(D_h(\pi^k, \pi^{k+1}) + D(h(\pi^{k+1}, w^k) - D_h(\pi^k, w^k)) + \rho(D_h(\pi^k, w^{k+1}) + D(h(\pi^{k+1}, \pi^{k+1}) - D_h(\pi^k, \pi^{k+1}))
\geq \rho(D_h(\pi^k, \pi^{k+1}) + \rho D_h(w^k, w^{k+1}) + \rho D_h(\pi^k, w^k) + \rho D_h(\pi^{k+1}, w^{k+1}).
\]
Here, \( (\cdot) \) uses the fact that the three point property of Bregman divergence holds, i.e., For any \( y, z \in \text{int dom} h \) and \( x \in \text{dom} h, \)
\[
D_h(x, z) - D_h(x, y) - D_h(y, z) = (\nabla h(y) - \nabla h(z), x - y).
\]
Moreover, \( (\cdot) \) holds as \( D_h(\cdot, \cdot) \) is symmetric on the whole sequence. To proceed, this together with (22) implies
\[
F_\rho(\pi^{k+1}, w^{k+1}) - F_\rho(\pi^k, w^k) \leq -\rho D_h(\pi^k, \pi^{k+1}) - \rho D_h(w^k, w^{k+1}).
\] (23)
Summing up (23) from \( k = 0 \) to \( +\infty \), we obtain
\[
F_\rho(\pi^\infty, w^\infty) - F_\rho(\pi^0, w^0) \leq -\kappa_1 \sum_{k=0}^{\infty} (D_h(\pi^k, \pi^{k+1}) + D_h(w^k, w^{k+1})).
\]
As the potential function \( F_\rho(\cdot, \cdot) \) is coercive and \( \{(\pi^k, w^k)\}_{k=0}^\infty \) is a bounded sequence, it means the left-hand side is bounded, which implies
\[
\sum_{k=0}^{\infty} (D_h(\pi^k, \pi^{k+1}) + D_h(w^k, w^{k+1})) < +\infty.
\]
\[
\square
\]
**Proof of Theorem 8 — Subsequence convergence result of BAPG**

**Proof.** Let \( (\pi^\infty, w^\infty) \) be a limit point of the sequence \( \{(\pi^k, w^k)\}_{k=0}^\infty \). Then, there exists a sequence \( \{n_k\}_{k=0}^\infty \) such that \( \{(\pi^{n_k}, w^{n_k})\}_{k=0}^\infty \) converges to \( (\pi^\infty, w^\infty) \). Replacing \( k \) by \( n_k \) in (18) and (19), taking limits on both sides as \( k \to +\infty \)
\[
0 \in \nabla_\pi f(\pi^\infty, w^\infty) + \rho(\nabla h(\pi^\infty) - \nabla h(w^\infty)) + \partial g_1(\pi^\infty)
0 \in \nabla_w f(\pi^\infty, w^\infty) + \rho(\nabla h(w^\infty) - \nabla h(\pi^\infty)) + \partial g_2(w^\infty).
\]
Based on the fact that \( \nabla_\pi f(\pi^\infty, w^\infty) = \nabla f(w^\infty) \) and \( \nabla_w f(\pi^\infty, w^\infty) = \partial f(\pi^\infty) \), it can be easily concluded that \( (\pi^\infty, w^\infty) \in \mathcal{X}_{\text{BAPG}} \).
\[
\square
\]
More importantly, the above subsequence convergence result can be extended to the global convergence if the Legendre function \( h(\cdot) \) is quadratic.
We revisit the BPG iteration update rule:

\[ t \]

where \( \sigma \) Again, together with the updates (18) and (19), this implies

\[(\text{Global Convergence of BAPG — Quadratic Case})\]

Suppose further that Problem \( \{ \pi^k, w^k \} \) to be non-empty. Suppose that the generated sequence \( \{ \pi^k \} \), whose the critical point set \( \mathcal{X} \) is assumed to be non-empty. Suppose that the generated sequence \( \{ \pi^k \} \) satisfying the following properties:

(A) (Sufficient Descent) There exist a constant \( \kappa_1 > 0 \) and an index \( k_1 \geq 0 \) such that for \( k \geq k_1 \),

\[ F \left( \pi^{k+1} \right) - F \left( \pi^k \right) \leq -\kappa_1 \| \pi^{k+1} - \pi^k \|^2. \]

(B) (Cost-to-Go Estimate) There exist a constant \( \kappa_2 > 0 \) and an index \( k_2 \geq 0 \) such that for \( k \geq k_2 \),

\[ F \left( \pi^{k+1} \right) - F(\pi^*) \leq \kappa_2 \left( \operatorname{dist}^2 \left( \pi^*, \mathcal{X} \right) + \| \pi^{k+1} - \pi^k \|^2 \right), \]

where \( \pi^* \) is the limit point of the sequence \( \{ \pi^k \} \).

(C) (Safeguard) There exist a constant \( \kappa_3 > 0 \) and an index \( k_3 \geq 0 \) such that for \( k \geq k_3 \),

\[ \| R(\pi^k) \|_2 \leq \kappa_3 \| \pi^{k+1} - \pi^k \|_2, \]

where \( R(\pi^k) \) is the proximal residual function, defined as

\[ R(\pi^k) = \| \pi^k - \operatorname{prox}_g(\pi^k - \nabla f(\pi^k)) \|_2. \]

Suppose further that Problem (1) possesses the Luo-Tseng error bound condition. Then, the sequence \( \{ F(\pi^k) \} \) converges Q-linearly to \( F(\pi^*) \) and the sequence \( \{ \pi^k \} \) converges R-linearly to some \( \pi^* \in \mathcal{X} \).
Proof of Theorem 8 — Linear convergence of BPG

Proof. As Problem (1) satisfies the Luo-Tseng local error bound condition, i.e., Proposition 3, we can invoke Fact 14 to conduct the convergence analysis. Under the given setting, it is sufficient to prove the above mentioned three key properties.

Step 1: Sufficient Decrease Property
It is worthwhile noting that \( f(\pi) \) is a quadratic function, i.e., \( f(\pi) = -\text{Tr}(D_X \pi D_Y \pi^T) \), then \( f(\pi) \) is gradient Lipschitz continuous with the modulus constant \( \sigma_{\text{max}}(D_X)\sigma_{\text{max}}(D_Y) \), where \( \sigma_{\text{max}}(\cdot) \) denotes the largest singular value. To simplify the notation, let \( L_f = \sigma_{\text{max}}(D_X)\sigma_{\text{max}}(D_Y) \).

\[
F(\pi^{k+1}) - F(\pi^k) \leq f(\pi^k) + \nabla f(\pi^k)^T(\pi^{k+1} - \pi^k) + \frac{L_f}{2}\|\pi^{k+1} - \pi^k\|^2 + g(\pi^{k+1}) - f(\pi^k) - g(\pi^k)
\]

\[
\leq \nabla f(\pi^k)^T(\pi^{k+1} - \pi^k) + \frac{L_f}{\sigma}D_h(\pi^{k+1}, \pi^k) + g(\pi^{k+1}) - g(\pi^k)
\]

\[
= \nabla f(\pi^k)^T(\pi^{k+1} - \pi^k) + \frac{1}{t_k}D_h(\pi^{k+1}, \pi^k) + g(\pi^{k+1}) - g(\pi^k) + \left(\frac{L_f}{\sigma} - \frac{1}{t_k}\right)D_h(\pi^{k+1}, \pi^k)
\]

\[
\leq \left(\frac{L_f}{\sigma} - \frac{1}{t_k}\right)D_h(\pi^{k+1}, \pi^k)
\]

\[
\leq -\left(\frac{1}{t_k} - \frac{L_f}{\sigma}\right)\|\pi^{k+1} - \pi^k\|^2,
\]

where \( \bullet \) holds due to the fact \( D_h(\pi^{k+1} - \pi^k) \geq \frac{\epsilon}{2}\|\pi^{k+1} - \pi^k\|^2 \) and \( \bullet \) follows as the optimal solution of \( \hat{F}(\pi; \pi^k) \) is \( \pi^{k+1} \), i.e., (6). By letting \( t_k = \frac{1}{L_f} - \frac{L_f}{\sigma} > 0 \), we get the desired result.

Step 2: Cost-to-Go Estimate Property
Let \( \tilde{\pi}^k \) be the projection of \( \pi^k \) onto \( \mathcal{X} \). Again, we aim at exploiting the structure information from the BPG update, i.e., (6). Similarly, as the optimal solution of \( \hat{F}(\pi; \pi^k) \) is \( \pi^{k+1} \), we have,

\[
\hat{F}(\pi^{k+1}; \pi^k) \leq \hat{F}(\pi^k; \pi^k)
\]

\[
\Rightarrow \nabla f(\pi^k)^T\pi^{k+1} + \frac{1}{t_k}D_h(\pi^{k+1}, \pi^k) + g(\pi^{k+1}) \leq \nabla f(\pi^k)^T\pi^k + \frac{1}{t_k}D_h(\pi^k, \pi^k) + g(\pi^k)
\]

\[
\Rightarrow \nabla f(\pi^k)^T(\pi^{k+1} - \pi^k) + g(\pi^{k+1}) - g(\pi^k) \leq \frac{1}{t_k}D_h(\pi^k, \pi^k) - \frac{1}{t_k}D_h(\pi^{k+1}, \pi^k).
\]

It implies

\[
\nabla f(\pi^k)^T(\pi^{k+1} - \pi^k) + g(\pi^{k+1}) - g(\pi^k) \leq \frac{1}{t_k}D_h(\pi^k, \pi^k) \leq \frac{L}{t_k}\|\pi^k - \pi^k\|^2 \leq \frac{L}{t_k}\text{dist}^2(\pi^k, \mathcal{X}),
\]

where \( L = \max\{L_h(K), \frac{4M_c}{\sigma}\} \) is a constant.

If we assume the sequence \(\{\pi^k\}_{k \geq 0}\) is lower bounded by a small constant \(\epsilon \), i.e., \(\pi^k \geq \epsilon I\), then we can conclude

\[
D_h(\pi^k, \pi^k) \leq L_h\|\pi^k - \pi^k\|^2,
\]

If \(\|\tilde{\pi}^k - \pi^k\| < \frac{\epsilon}{2}\), then \(\pi^k \geq \frac{\epsilon}{2}I_{nm} \). Therefore, \( L = \frac{L_h(K)}{2} \) where \( K \) is the given compact set, i.e., \( K = C_1 \cap C_2 \cap \{\pi: \pi \geq \epsilon I_{nm}\} \). Otherwise, due to the coercive property of \( D_h(\cdot, \cdot) \) and boundedness of \(\{\pi^k\}_{k \geq 0}\), there exists a constant \( M > 0 \) such that

\[
D_h(\tilde{\pi}^k, \pi^k) \leq M \leq \frac{4M_c^2}{\epsilon^2}\|\pi^k - \pi^k\|^2.
\]

By the mean value theorem and \( f \) is continuous differentiable, there exists a \( \tilde{\pi}^k \in [\pi^k, \pi^{k+1}] \) such that,

\[
f(\pi^{k+1}) - f(\pi^k) = \nabla f(\tilde{\pi}^k)^T(\pi^{k+1} - \pi^k).
\]
Finally, we compute,
\[
F(\pi^{k+1}) - F(\pi^k) \\
= \nabla f(\hat{\pi}^k)^T (\pi^{k+1} - \pi^k) + g(\pi^{k+1}) - g(\pi^k) \\
\leq \frac{L}{t} \text{dist}^2(\pi^k, \mathcal{X}) + (\nabla f(\hat{\pi}^k) - \nabla f(\pi^k))^T (\pi^{k+1} - \pi^k) \\
\leq \frac{L}{t} \text{dist}^2(\pi^k, \mathcal{X}) + L_f \| \hat{\pi}^k - \pi^k \| \| \pi^{k+1} - \pi^k \| \\
\leq \frac{L}{t} \text{dist}^2(\pi^k, \mathcal{X}) + L_f \left( \| \hat{\pi}^k - \pi^k \| + \| \pi^{k+1} - \pi^k \| \right) \| \pi^{k+1} - \pi^k \| \\
\leq \frac{L}{t} \text{dist}^2(\pi^k, \mathcal{X}) + L_f \left( \| \pi^k \| - \| \pi^{k+1} \| + \| \pi^{k+1} - \pi^k \| \right) \| \pi^{k+1} - \pi^k \| \\
\leq \frac{L}{t} \text{dist}^2(\pi^k, \mathcal{X}) + \frac{3L_f}{2} \| \pi^k - \pi^{k+1} \|^2 + \frac{L_f}{2} \| \pi^{k+1} - \pi^k \|^2 \\
\leq \frac{L}{t} \text{dist}^2(\pi^k, \mathcal{X}) + 3L_f \| \pi^k - \pi^{k+1} \|^2 + \frac{7L_f}{2} \| \pi^{k+1} - \pi^k \|^2 \\
= \left( \frac{L}{t} + 3L_f \right) \text{dist}^2(\pi^k, \mathcal{X}) + \frac{7L_f}{2} \| \pi^{k+1} - \pi^k \|^2 \\
\leq \max \left( \left( \frac{L}{t} + 3L_f \right), \frac{7L_f}{2} \right) \left( \text{dist}^2(\pi^k, \mathcal{X}) + \| \pi^{k+1} - \pi^k \|^2 \right) .
\]

At last, by applying Lemma 3.1 in (Luo & Tseng, 1992), we know \( F(\cdot) \) enjoys the isolation property around the local region. Therefore, by letting \( \kappa_2 = \max \left( \left( \frac{L}{t} + 3L_f \right), \frac{7L_f}{2} \right) \), we obtain the desired result.

**Step 3: Safeguard Property**

Again, invoking the optimality condition of the main BPG update (6), we have
\[
0 \in \nabla f(\pi^k) + \partial g(\pi^{k+1}) + \frac{1}{t_k} \nabla h(\pi^{k+1}, \pi^k) \\
\Rightarrow 0 \in \nabla f(\pi^{k+1}) + \partial g(\pi^{k+1}) + \nabla f(\pi^k) - \nabla f(\pi^{k+1}) + \frac{1}{t_k} (\nabla h(\pi^{k+1}) - \nabla h(\pi^k)).
\]

Upon this, we have
\[
\text{dist}(0, \partial F(\pi^{k+1})) \leq L_f \| \pi^{k+1} - \pi^k \| + \frac{1}{t_k} \| \nabla h(\pi^{k+1}) - \nabla h(\pi^k) \| \\
\leq \left( L_f + \frac{L_h(K)}{t_k} \right) \| \pi^{k+1} - \pi^k \| .
\]

Moreover, due to Lemma 4.1 in (Li & Pong, 2018), it is so nice that we can connect the proximal residual function \( R(\pi^k) \) and \( \text{dist}(0, \partial F(\pi^{k+1})) \). That is,
\[
\| R(\pi^{k+1}) \| \leq \text{dist}(0, \partial F(\pi^{k+1})).
\]

Then,
\[
\| R(\pi^k) \| = \| R(\pi^k) - R(\pi^{k+1}) + R(\pi^{k+1}) \| \\
\leq \| R(\pi^k) - R(\pi^{k+1}) \| + \| R(\pi^{k+1}) \| \\
\leq \| R(\pi^k) - R(\pi^{k+1}) \| + \text{dist}(0, \partial F(\pi^{k+1})) \\
= \| \pi^k - \pi^{k+1} \| + \text{prox}_g(\pi^{k+1} - \nabla f(\pi^{k+1})) - \text{prox}_g(\pi^k - \nabla f(\pi^k)) \| + \text{dist}(0, \partial F(\pi^{k+1})) \\
\leq \| \pi^k - \pi^{k+1} \| + \| \text{prox}_g(\pi^{k+1} - \nabla f(\pi^{k+1})) - \text{prox}_g(\pi^k - \nabla f(\pi^k)) \| + \text{dist}(0, \partial F(\pi^{k+1})) \\
\leq \text{dist}(0, \partial F(\pi^{k+1})).
\]
\[
\begin{align*}
\leq (L_f + 2) \| \pi^k - \pi^{k+1} \| + \text{dist}(0, \partial F(\pi^{k+1})) \\
\leq \left( 2L_f + \frac{L_h(K)}{t} + 2 \right) \| \pi^{k+1} - \pi^k \|.
\end{align*}
\]

This, together with (25), implies the safeguard property with \( \kappa_3 = \left( 2L_f + \frac{L_h(K)}{t} + 2 \right) \).

**Additional Experiment Results**

![Image of additional experiment results]

Source codes of all baselines used in this paper

- GWL (Xu et al., 2019b): [https://github.com/HongtengXu/gwl](https://github.com/HongtengXu/gwl)
- ScalaGWL (Xu et al., 2019a): [https://github.com/HongtengXu/s-gwl](https://github.com/HongtengXu/s-gwl)

- SpecGW (Chowdhury & Needham, 2021):
  [https://github.com/trneedham/Spectral-Gromov-Wasserstein](https://github.com/trneedham/Spectral-Gromov-Wasserstein)

- (Python) eBPG (Solomon et al., 2016): [https://github.com/PythonOT/POT](https://github.com/PythonOT/POT)

- (Matlab) eBPG (Solomon et al., 2016): [https://github.com/justsol/EntropicMetricAlignment](https://github.com/justsol/EntropicMetricAlignment)