Supplementary materials for the paper
“Meta-Learning with a Geometry-Adaptive Preconditioner”

A. Toy example
To build an intuition for the effect of Riemannian metric, we construct a 2-D toy example over the parameter space. A learner minimizes an objective function of the form below.

\[ f(x_1, x_2) = x_1^2 + x_2^2 + x_1 x_2 + \frac{1}{2}(\sin^2 5x_1 + \sin^2 5x_2) \]
\[ - \frac{1}{2}(\cos^2 3x_1 + \cos^2 3x_2) \]  
(13)

We set the initial point to \((x_1, x_2) = (-4, -2)\) and the learning rate to 0.1. In Figure 2 (a), we train the learner for 50 iterations. In Figure 2 (b), we define a preconditioner \(P_1\) as follows:

\[ P_1 = \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & -0.3 \end{bmatrix} \]  
(14)

and train the learner with \(P_1\) for 13 iterations. In Figure 2 (c), we derive a preconditioner \(P_2\), which is the Riemannian metric corresponding to the parameter space (Eq. 13) as follows [32]:

\[ P_2 = \begin{bmatrix} 1 + u^2 & uv \\ uv & 1 + v^2 \end{bmatrix} \]  
(15)

where

\[ u = 2x_1 + x_2 + 3\sin(3x_1)\cos(3x_1) + 5\cos(5x_1)\sin(5x_1) \]
\[ v = 2x_2 + x_1 + 3\sin(3x_2)\cos(3x_2) + 5\cos(5x_2)\sin(5x_2) \]

We train the learner with \(P_2\) for 50 iterations.

B. Proofs of Theorems
Definition 1. Two \(n \times n\) matrices \(A\) and \(B\) are similar if there exists an invertible \(n \times n\) matrix \(P\) such that

\[ B = P^{-1}AP \]  
(16)

Lemma 1. Let \(A = \text{blkdiag}(A_1, \cdots, A_n)\) be a block diagonal matrix such that the main-diagonal blocks \(A_i\) are \(k \times k\) positive definite matrices. Then \(A\) is a positive definite matrix.

Proof. First, we show that \(A\) is a positive definite matrix. For all non-zero \(x = (x_1, \cdots, x_n) \in \mathbb{R}^n\) where \(x_i \in \mathbb{R}^k\), we can derive the following.

\[ x^T Ax = x^T \text{blkdiag}(A_1, \cdots, A_n) x \]
\[ = x_1^T A_1 x_1 + \cdots + x_n^T A_n x_n \]
\[ > 0 \quad (\because A_i \text{ is a positive definite}) \]  
(17)

Next, we show that \(A\) is a symmetric matrix. Since \(A_i\) is a symmetric matrix (i.e., \(A_i = A_i^T\)), we find that the following is satisfied.

\[ A^T = \text{blkdiag}(A_1, \cdots, A_n)^T \]
\[ = \text{blkdiag}(A_1^T, \cdots, A_n^T) \]
\[ = \text{blkdiag}(A_1, \cdots, A_n) \]  
(18)

Hence, \(A\) is a symmetric matrix. Therefore, \(A\) is a positive definite matrix.

\[ \Box \]

Theorem 1. Let \(\tilde{G}_{\tau, k}^l \in \mathbb{R}^{m \times n}\) be the \('l\)-layer \(k\)-th inner-step' gradient matrix transformed by meta parameter \(M^l\) for task \(\tau\). Then preconditioner \(P_{\text{GAP}}\) induced by \(\tilde{G}_{\tau, k}^l\) is a Riemannian metric and depends on the task-specific parameters \(\theta_{\tau, k}\).

Proof. We can rewrite the \(\tilde{G}_{\tau, k}^l\) as follows:

\[ \tilde{G}_{\tau, k}^l = U_{\tau, k}^l(M^l \cdot \Sigma_{\tau, k}^l) V_{\tau, k}^T \]
\[ = (U_{\tau, k}^l M^l U_{\tau, k}^T)^T U_{\tau, k}^l \Sigma_{\tau, k}^l V_{\tau, k}^T \]
\[ = D_{\tau, k}^l G_{\tau, k}^l, \]

where \(D_{\tau, k}^l = U_{\tau, k}^l M^l U_{\tau, k}^T\). To induce preconditioner in Eq. (19), we reformulate Eq. (19) as the general gradient descent form (i.e., matrix-vector product):

\[ \text{vec}(G_{\tau, k}^l) = \text{blkdiag}(D_{\tau, k}^l, \cdots, D_{\tau, k}^l) \cdot \text{vec}(G_{\tau, k}^l) \]  
(20)

where \(P_{\text{GAP}}\) is a block diagonal matrix such that the main-diagonal blocks are \(D_{\tau, k}^l\)'s. Now, we prove that block \(D_{\tau, k}^l\) is a positive definite matrix. Since \(D_{\tau, k}^l\) is similar to \(M^l\) by Definition 1, they have the same eigenvalues. In addition, all eigenvalues of \(D_{\tau, k}^l\) are positive because all eigenvalues of \(M^l\) are positive. Next, we show that \(D_{\tau, k}^l\) is a symmetric matrix as below.

\[ (D_{\tau, k}^l)^T = U_{\tau, k}^l M^l U_{\tau, k}^T \]
\[ = U_{\tau, k}^l M^l U_{\tau, k}^T \]  
(21)

Therefore, \(D_{\tau, k}^l\) is a positive definite matrix. By Lemma 1, \(P_{\text{GAP}}\) is a positive definite matrix.

Since the unitary matrix \(U_{\tau, k}^l\) depends on the gradient matrix \(\tilde{G}_{\tau, k}^l\), it depends on the task-wise parameters \(\theta_{\tau, k}\).
Following Eq. (27), we show that its expectation is equal to:

\[
\mathbb{E}[\langle x, y \rangle] = \mathbb{E}[X_1],
\]

and its variance is equal to:

\[
\mathbb{V}(X_1) = \frac{1}{n} \quad \text{(by Lemma 2)}.
\]

By applying Chebyshev’s inequality [12] on \( \langle x, y \rangle \), we have

\[
P(|\langle x, y \rangle| \geq \epsilon) \leq \frac{1}{n\epsilon^2},
\]

for any real number \( k > 0 \). Let \( \frac{k}{\sqrt{n}} \) be a \( \epsilon \). Then we rewrite the in Eq. (30) as follows:

\[
P(|\langle x, y \rangle| \geq \epsilon) \leq \frac{1}{n\epsilon^2}.
\]

This result indicates that the two vectors \( x \) and \( y \) become asymptotically orthogonal as \( n \) increases. □

**Assumption 2.** The elements of gradient matrix follows an i.i.d. normal distribution with zero mean.

**Theorem 2.** Let \( G \in \mathbb{R}^{m \times n} \) be a gradient matrix and \( \tilde{G} \) be the gradient matrix transformed by meta parameter \( M \). Under the Assumption 2, as \( n \) becomes large, \( \tilde{G} \) asymptotically becomes equivalent to \( MG \) as follows:

\[
\tilde{G} \cong MG
\]

**Proof.** Let \( g_1, g_2, \ldots, g_m \) be the row vectors of \( G \). Then,

\[
G = \begin{bmatrix} \|g_1\| & \cdots & \|g_m\| \\ \vdots & \ddots & \vdots \\ \|g_1\| & \cdots & \|g_m\| \end{bmatrix}.
\]

Under the Assumption 2, \( g_1, g_2, \ldots, g_m \) follow an i.i.d multivariate normal distribution. Then, we have

\[
\frac{g_i}{\|g_i\|} \perp \frac{g_j}{\|g_j\|} \quad (\forall i \neq j),
\]

and \( \frac{g_i}{\|g_i\|}, \frac{g_j}{\|g_j\|} \) are located on the \((n-1)\)-dimensional unit sphere [41]. Since independent vectors \( \frac{g_i}{\|g_i\|}, \frac{g_j}{\|g_j\|} \) are located on the \((n-1)\)-dimensional unit sphere, the vectors are asymptotically orthogonal as \( n \) increases by Lemma 2. Now, we rewrite \( G \) as follows.

\[
G = I \begin{bmatrix} \|g_1\| & \cdots & \|g_m\| \\ \vdots & \ddots & \vdots \\ \|g_1\| & \cdots & \|g_m\| \end{bmatrix}.
\]

Since \( I \) is a unitary matrix and \( \frac{g_1}{\|g_1\|}, \cdots, \frac{g_m}{\|g_m\|} \) approximately becomes semi-unitary matrices as \( n \) increases, the singular values of \( G \) asymptotically become \( \|g_1\|, \cdots, \|g_m\| \). By Eq. (35), the following holds under the Assumption 2 as \( n \) becomes sufficiently large.

\[
\tilde{G} \cong MG
\]
C. Implementation Details

For the reproducibility, we provide the details of implementation. Our implementations are based on Torchmeta [15] library. Our implementation code is available at: https://github.com/Suhyun777/CVPR23-GAP.

C.1. Hyper-parameters

For all the experiments, we use the hyper-parameters in Table 9.

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>mini-ImageNet</th>
<th>tiered-ImageNet</th>
<th>Cross-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch size</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Total training iterations</td>
<td>70000</td>
<td>100000</td>
<td>130000</td>
</tr>
<tr>
<td>inner learning rate</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>outer learning rate</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>The number of training inner steps</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Data augmentation</td>
<td>None</td>
<td>random flip</td>
<td>random flip</td>
</tr>
</tbody>
</table>

Table 9. Hyper-parameters used for training GAP on various few-shot learning experiments.

C.2. Backbone Architecture

C.2.1 2-layer MLP network.

For the few-shot regression experiment, we use a simple Multi-Layer Perceptron (MLP) with 1-dimensional input/output and 40-dimensional hidden layers as in [20].

C.2.2 4-Conv network.

For the few-shot classification and cross-domain few-shot classification experiments, we use the standard Conv-4 backbone used in [56], comprising 4 modules with $3 \times 3$ convolutions, with 128 filters followed by batch normalization [26], ReLU non-linearity, and $2 \times 2$ max-pooling.

C.3. Optimization

We use ADAM optimizer [28]. For tiered-ImageNet experiment, the learning rate (LR) is scheduled by the cosine learning rate decay [38] for every 500 iterations. In all the experiments except for the tiered-ImageNet, the learning rate is unscheduled.

C.4. Preconditioning

In the few-shot regression experiment, we apply preconditioner only to the hidden layer. In both few-shot classification and cross-domain few-shot classification, we only apply preconditioner to 4 convolutional layers.