Federated Learning of Models Pre-Trained on Different Features with Consensus Graphs
(Supplementary Material)

Tengfei Ma[1] **Trong Nghia Hoang**[2] **Jie Chen**[3,1]

[1]IBM Research
[2]Washington State University
[3]MIT-IBM Watson AI Lab

1 PERMUTATION AMBIGUITY EXAMPLE FOR GRU

In Section 4, we discuss that one can arbitrarily permute the latent representations while keeping a local model fixed. Here, we give another example – the GRU. Let $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ be an input sequence. The embedding function $\mathbf{h} = \mathrm{embedding}(\mathbf{x})$ implemented as a GRU reads:

```
function $\mathbf{h} = \mathrm{GRU}(\{\mathbf{x}_t\}_{t=1}^T)$
    $\mathbf{h}_0 = \mathbf{0}$
    for $t = 1, \dots, T$ do
        $\mathbf{z}_t = \mathrm{sigmoid}(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z)$
        $\mathbf{r}_t = \mathrm{sigmoid}(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r)$
        $\mathbf{n}_t = \tanh(\mathbf{W}_n\mathbf{x}_t + \mathbf{U}_n(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_n)$
        $\mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \mathbf{n}_t$
    end for
    return $\mathbf{h} = \mathbf{h}_T$
end function
```

One can arbitrarily permute the elements of $\mathbf{h}$ through manipulating the GRU parameters properly. To achieve $\mathbf{h}[\mathrm{p}] = \mathrm{embedding}(\mathbf{x}[\mathrm{p}])$,

- the gate outputs and bias vectors ($\mathbf{z}_t$, $\mathbf{r}_t$, $\mathbf{n}_t$, $\mathbf{h}_t$, $\mathbf{b}_z$, $\mathbf{b}_r$, $\mathbf{b}_n$) will need be permuted accordingly ($\mathbf{z}_t[\mathrm{p}]$, $\mathbf{r}_t[\mathrm{p}]$, $\mathbf{n}_t[\mathrm{p}]$, $\mathbf{h}_t[\mathrm{p}]$, $\mathbf{b}_z[\mathrm{p}]$, $\mathbf{b}_r[\mathrm{p}]$, $\mathbf{b}_n[\mathrm{p}]$);
- the weight matrices attached to the input ($\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}_n$) will need to have their rows (i.e., output neurons) permuted ($\mathbf{W}_z[\mathrm{p}, :]$, $\mathbf{W}_r[\mathrm{p}, :]$, $\mathbf{W}_n[\mathrm{p}, :]$); and
- the weight matrices attached to the hidden states ($\mathbf{U}_z$, $\mathbf{U}_r$, $\mathbf{U}_n$) will need to have both their rows and columns permuted ($\mathbf{U}_z[\mathrm{p}, \mathrm{p}]$, $\mathbf{U}_r[\mathrm{p}, \mathrm{p}]$, $\mathbf{U}_n[\mathrm{p}, \mathrm{p}]$).

2 PROOFS AND ADDITIONAL RESULTS OF THEOREM 1

2.1 DISTRIBUTION OF GUMBEL SOFTMAX

The Gumbel softmax reparameterization trick [Jang et al., 2017, Maddison et al., 2017] works in the following manner. Let $\mathrm{Cat}(\boldsymbol{\pi})$ be the categorical distribution with probability vector $\boldsymbol{\pi}$ and let $\mathbf{g}$, of the same shape as $\boldsymbol{\pi}$, be a vector variable whose elements are i.i.d. $\sim \mathrm{Gumbel}(0, 1)$. Then,

$$\mathbf{y} \quad = \quad \mathrm{softmax}\left(\frac{1}{\tau}(\log \boldsymbol{\pi} + \mathbf{g})\right), \quad \tau > 0 \tag{1}$$

admits a distribution converging to $\mathrm{Cat}(\boldsymbol{\pi})$ when $\tau \to 0$. Hence, to sample $\mathrm{Ber}(\theta)$ approximately but differentiably, it suffices to let $\boldsymbol{\pi} = [\theta, 1 - \theta]^\top$ and use y_1 as the sample.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

As preliminary, we consider the first entry y_1 of the random variable $\mathbf{y}$ defined in (1) for the Gumbel softmax parameterization. Note that for any $\tau \neq 0$, y_1 is only approximately binary; the possible values of y_1 in fact span the entire interval $[0, 1]$. We derive the following CDF for y_1. For notational simplicity, θ denotes a scalar rather than a matrix.

Theorem 1. For all $\tau > 0$, $\theta \in (0, 1)$, and $t \in [0, 1]$, we have

$$\Pr(y_1 \leq t) \quad = \quad \frac{t^\tau(1-\theta)}{t^\tau(1-\theta) + (1-t)^\tau \theta}. \tag{2}$$

Proof. We first consider the case $0 < t < 1$. Through simple algebraic manipulation, we obtain that $y_1 \leq t$ is equivalent to

$$g_1 - g_2 \quad \leq \quad \tau \log \frac{t}{1-t} - \log \frac{\theta}{1-\theta}. \tag{3}$$

Let $g_1 = -\log(-\log u)$ and $g_2 = -\log(-\log v)$, where u and v are independent and $\sim \mathcal{U}(0, 1)$. Then, (3) is equivalent to

$$v \;\geq\; u^M \quad \text{where} \quad M \;=\; \frac{t^\tau(1-\theta)}{(1-t)^\tau \theta}.$$

Therefore, by recalling that u and v are uniform in $[0, 1]^2$, we note that the probability that $v \leq u^M$ happens is the double integral

$$\Pr(v \geq u^M) \;=\; \int_0^1 \int_{u^M}^1 1 \, dv du.$$

This integral is nothing but

$$1 - \int_0^1 u^M \, du \;=\; \frac{M}{1+M},$$

which completes the proof of (2). The cases of $t = 0$ or 1 obviously hold by continuity.

2.2 PROOF OF THEOREM 1

We first consider the case when the distribution with CDF F is finitely supported on $[a, b]$. Through simple algebraic manipulation, we obtain that $z \leq t$ is equivalent to $s \geq M$ where $M := F^{-1}(\theta) + \tau \log(t^{-1} - 1)$. If $t < \text{sigmoid}((F^{-1}(\theta) - b)/\tau)$, we see that $M > b$ and thus such s can never occur. Similarly, if $t > \text{sigmoid}((F^{-1}(\theta) - a)/\tau)$, we see that $M < a$, which indicates that $s \geq M$ always happens. Otherwise, when t is within the two extremes, the probability that $s \geq M$ happens is $1 - F(M)$, concluding the proof of Theorem 1.

The statement of the theorem regarding the case when the distribution is not finitely supported is obviously true.

To show that the distribution of z converges to $\text{Ber}(\theta)$, let us first consider the scenario when the distribution with CDF F is finitely supported. The CDF of z is always continuous but it has three segments connected by two joints: $t_1 = \text{sigmoid}((F^{-1}(\theta) - b)/\tau)$ and $t_2 = \text{sigmoid}((F^{-1}(\theta) - a)/\tau)$. When $\tau \to 0$, the joint $t_1 \to 0$ and the joint $t_2 \to 1$ and thus the middle segment has a wider and wider support converging to $[0, 1]$. Hence, it suffices to consider only the middle segment. Further, with an analogous argument for other scenarios, it is also true that it suffices to consider only the third case of Theorem 1.

In this case, for any fixed $t < 1$ and when $\tau \to 0$, we have $\tau \log(t^{-1} - 1) \to 0$ and thus $\Pr(z \leq t) \to 1 - F(F^{-1}(\theta)) = 1 - \theta$. Meanwhile, we cannot push $\tau \to 1$ because then the limit of $\tau \log(t^{-1} - 1)$ is undefined. However, we know by definition that $\Pr(z \leq 1) = 1$. Hence, the continuous distribution of z converges to a degenerate distribution $\Pr(z < 1) = 1 - \theta$ and $\Pr(z = 1) = 1$. This is the CDF of $\text{Ber}(\theta)$.

3 TUNING GUIDANCE FOR TEMPERATURE τ

Our tuning guidance for the temperature τ is motivated from an asymptotic convergence comparison between **ICDF** and Gumbel reparameterization, which is featured in the theorem below.

Theorem 2. When τ is small,

$$\begin{aligned}\mathrm{Bias}(y_1) &= \frac{1}{6}\tau^2\pi^2\theta(1-\theta)(1-2\theta) \;+\; O(\tau^4), && (4)\\ \mathrm{Bias}(z) &= \frac{1}{6}\tau^2\pi^2 F''(F^{-1}(\theta)) \;+\; O(\tau^4). && (5)\end{aligned}$$

Moreover, when F is the CDF of a normal variable $\sim \mathbb{N}(0,\sigma^2)$,

$$\mathrm{Bias}(z) \;=\; -\frac{1}{6\sigma^2}\tau^2\pi^{\frac{3}{2}}\,\mathrm{erf}^{-1}(2\theta-1)\,e^{-(\mathrm{erf}^{-1}(2\theta-1))^2} \;+\; O(\tau^4). \tag{6}$$

Its formal proof is detailed later.

Theorem 2 suggests that the **ICDF** method converges equally fast as does the Gumbel trick – both on the order of $O(\tau^2)$. On the other hand, the biases depend on θ. Thus, one cannot set temperatures τ independently of the desired probability θ to equate the two biases. In practice, τ is a tunable hyper-parameter and a guidance on the tuning range is therefore necessary.

To begin, we use a subscript to distinguish the two temperatures – τ_{g} for the Gumbel trick and τ_{i} for the **ICDF** method – and write, based on (4) and (6) and ignoring the high order terms,

$$\frac{\mathrm{Bias}(y_1)}{\mathrm{Bias}(z)} \;\simeq\; \frac{\tau_{\mathrm{g}}^2\sigma^2}{\tau_{\mathrm{i}}^2}r(\theta) \quad \text{where} \quad r(\theta) \;=\; \frac{\sqrt{\pi}\theta(1-\theta)(2\theta-1)}{\mathrm{erf}^{-1}(2\theta-1)e^{-(\mathrm{erf}^{-1}(2\theta-1))^2}}.$$

Note that $r(\theta)$ is symmetric around $\theta=\frac{1}{2}$, is concave, attains maximum $\frac{1}{2}$ when $\theta=\frac{1}{2}$, and attains minimum 0 when $\theta=0,1$. Hence, if $\tau_{\mathrm{g}}=\tau_{\mathrm{i}}$ and $\sigma=\sqrt{2}$, the bias of the Gumbel trick is (approximately) smaller than that of the **ICDF** method. On the other hand, for a $\sigma>\sqrt{2}$, there exist $\widetilde{\theta}_1<\widetilde{\theta}_2$ such that $\sigma^{-2}=r(\widetilde{\theta}_1)=r(\widetilde{\theta}_2)$ and that $\mathrm{Bias}(y_1)\gtrapprox\mathrm{Bias}(z)$, whenever $\theta\in[\widetilde{\theta}_1,\widetilde{\theta}_2]$. For example, when $\sigma\approx 2.5$, on the interval $\theta\in[0.01,0.99]$, the bias of the Gumbel trick is (approximately) greater than that of the **ICDF** method.

Based on the foregoing, a practical guide is to use the same tuning range of τ for the **ICDF** method as for the Gumbel trick. A small change of σ (e.g., $\sqrt{2}$ versus 2.5) will entirely flip the landscape of the bias comparison between the two methods. Because the tuning range is much wider than the change of σ, for simplicity it suffices to fix $\sigma=1$.

4 PROOF OF THEOREM 2 AND ADDITIONAL RESULTS

By the definition of bias, we have

$$\mathrm{Bias}(x)=\mathbb{E}[x]-\theta \quad \text{where} \quad \mathbb{E}[x]=\int_0^1 t\,d\,\mathrm{Pr}(x\le t)=1-\int_0^1 \mathrm{Pr}(x\le t)\,dt.$$

Therefore, for Gumbel softmax,

$$\mathrm{Bias}(y_1)=1-\theta-\int_0^1 \frac{t^\tau(1-\theta)}{t^\tau(1-\theta)+(1-t)^\tau\theta}\,dt,$$

and for **ICDF** with any F,

$$\mathrm{Bias}(z)=\int_0^1 F(F^{-1}(\theta)+\tau\log(t^{-1}-1))\,dt-\theta.$$

We now prove Theorem 2 in a few parts.

Proof of (5). Let $s=F^{-1}(\theta)$ and perform a change of variable $m=\log(t^{-1}-1)$. Then,

$$\mathrm{Bias}(z)=\int_0^1[F(s+\tau m)-F(s)]\,dt=\int_{-\infty}^{\infty}[F(s+\tau m)-F(s)]\frac{e^m}{(1+e^m)^2}\,dm.$$

We perform Taylor expansion of F around s and obtain

$$F(s+\tau m)-F(s)=\sum_{n=1}^{\infty}\frac{F^{(n)}(s)}{n!}\tau^n m^n.$$

Therefore,

$$\text{Bias}(z) = \sum_{n=1}^{\infty} \frac{F^{(n)}(s)}{n!} \tau^n \int_{-\infty}^{\infty} \frac{m^n e^m}{(1+e^m)^2}\, dm$$

Each integral term is finite and the odd terms vanish because the integrands are odd functions. Thus, for small τ, we are left with

$$\text{Bias}(z) = \frac{F''(s)}{2} \tau^2 \int_{-\infty}^{\infty} \frac{m^2 e^m}{(1+e^m)^2}\, dm + O(\tau^4).$$

The definite integral evaluates to $\frac{\pi^2}{3}$; we therefore conclude the proof.

Proof of (6). Equation (6) is straightforward by substuting

$$F''(s) = -\frac{s}{\sigma^3\sqrt{2\pi}} e^{-\frac{s^2}{2\sigma^2}} = -\frac{\text{erf}^{-1}(2\theta-1)}{\sigma^2\sqrt{\pi}} e^{-(\text{erf}^{-1}(2\theta-1))^2}.$$

into (5).

Proof of (4). To simplify notation, let $\beta = \theta/(1-\theta)$ and perform a change of variable $m = \log(t^{-1}-1)$. Then,

$$\int_0^1 \frac{t^\tau(1-\theta)}{t^\tau(1-\theta)+(1-t)^\tau\theta}\, dt = \int_0^1 \frac{dt}{1+\beta e^{m\tau}} = \int_{-\infty}^{\infty} \frac{1}{1+\beta e^{m\tau}} \frac{e^m}{(1+e^m)^2}\, dm.$$

Denote $h(\tau, m) = [1+\beta e^{m\tau}]^{-1}$. Treating h a function of τ and performing Taylor expansion around zero, we obtain

$$h(\tau, m) = \sum_{n=0}^{\infty} \frac{h^{(n)}(0,m)}{n!} \tau^n.$$

Therefore,

$$\int_0^1 \frac{t^\tau(1-\theta)}{t^\tau(1-\theta)+(1-t)^\tau\theta}\, dt = \sum_{n=0}^{\infty} \frac{\tau^n}{n!} \int_{-\infty}^{\infty} h^{(n)}(0,m) \frac{e^m}{(1+e^m)^2}\, dm.$$

In a moment, we will show that for all n,

$$h^{(n)}(0,m) = C_n m^n \quad \text{where } C_n \text{ is independent of } m. \tag{7}$$

Suppose that (7) holds. Then, each integral term is finite and the odd terms vanish, because the integrands are odd functions. Therefore, for small τ, we are left with

$$\int_0^1 \frac{t^\tau(1-\theta)}{t^\tau(1-\theta)+(1-t)^\tau\theta}\, dt = C_0 \int_{-\infty}^{\infty} \frac{e^m}{(1+e^m)^2}\, dm + C_2 \frac{\tau^2}{2} \int_{-\infty}^{\infty} \frac{m^2 e^m}{(1+e^m)^2}\, dm + O(\tau^4).$$

By calculating

$$C_0 = h(0,m) = [1+\beta]^{-1} = 1-\theta, \qquad C_2 = h''(0,m) = -\theta(1-\theta)(1-2\theta),$$
$$\int_{-\infty}^{\infty} \frac{e^m}{(1+e^m)^2}\, dm = 1, \qquad \int_{-\infty}^{\infty} \frac{m^2 e^m}{(1+e^m)^2}\, dm = \frac{\pi^2}{3},$$

we conclude that

$$\text{Bias}(y_1) = \frac{\tau^2\pi^2\theta(1-\theta)(1-2\theta)}{6} + O(\tau^4).$$

It remains to prove (7). We suppress the argument on m and write $g(\tau) = 1 + \beta e^{m\tau}$ and $h(\tau) = g(\tau)^{-1}$. By Faà di Bruno's formula,

$$h^{(n)}(0) = \left(\frac{1}{g(\tau)}\right)^{(n)}\Bigg|_{\tau=0} = \sum_{k=1}^{n} \frac{(-1)^k k!}{g(0)^{k+1}} \cdot B_{n,k}\Big(g'(0), g''(0), \dots, g^{(n-k+1)}(0)\Big),$$

where $B_{n,k}$ is the Bell polynomial. Clearly, $g(0) = 1+\beta$ and $g^{(r)}(0) = \beta m^r$ for all $r > 0$. Hence, $B_{n,k}$ is a multiple of m^n. Therefore, $h^{(n)}(0)$ is a multiple of m^n.

4.1 ADDITIONAL RESULT REGARDING THE BIAS

Theorem 2 states results for a small temperature τ. The purpose is to understand the limiting behavior of the bias. Here, we give an additional result for any $\tau > 0$. It states that the biases of the two sampling approaches have the same sign. This result is a nontrivial extension of Theorem 2 and requires a different proof technique.

Theorem 3. For any $\tau > 0$,

$$\mathrm{Bias}(y_1) > 0 \text{ when } \theta < \tfrac{1}{2}, \quad \mathrm{Bias}(y_1) = 0 \text{ when } \theta = \tfrac{1}{2}, \quad \mathrm{Bias}(y_1) < 0 \text{ when } \theta > \tfrac{1}{2}. \tag{8}$$

Moreover, if $F'(x)$ (that is, the pdf) is even and is increasing when $x < 0$, then

$$\mathrm{Bias}(z) > 0 \text{ when } \theta < \tfrac{1}{2}, \quad \mathrm{Bias}(z) = 0 \text{ when } \theta = \tfrac{1}{2}, \quad \mathrm{Bias}(z) < 0 \text{ when } \theta > \tfrac{1}{2}. \tag{9}$$

We prove Theorem 3 in two parts.

Proof of (8). Consider

$$\mathrm{Bias}(y_1) = \int_0^1 g(t,\theta)\,dt \quad \text{where} \quad g(t,\theta) = 1 - \theta - \frac{t^\tau(1-\theta)}{t^\tau(1-\theta) + (1-t)^\tau\theta}.$$

With a brute-force calculation, we have

$$g(t,\theta) + g(1-t,\theta) = \frac{[(1-t)^\tau - t^\tau]^2\theta(1-\theta)(1-2\theta)}{[t^\tau(1-\theta)+(1-t)^\tau\theta][(1-t)^\tau(1-\theta)+t^\tau\theta]}.$$

All terms on the right-hand side are positive, except $1 - 2\theta$. Therefore, when $\theta < \frac{1}{2}$, $g(t,\theta) + g(1-t,\theta) > 0$ and hence

$$\mathrm{Bias}(y_1) = \int_0^1 \frac{g(t,\theta) + g(1-t,\theta)}{2}\,dt > 0.$$

The other cases ($\theta > \frac{1}{2}$ and $\theta = \frac{1}{2}$) are similarly proved.

Proof of (9). Consider

$$\mathrm{Bias}(z) = \int_0^1 h(t,\theta)\,dt - \theta \quad \text{where} \quad h(t,\theta) = F(F^{-1}(\theta) + \tau\log(t^{-1}-1)).$$

We have

$$h(1-t,\theta) = F(F^{-1}(\theta) - \tau\log(t^{-1}-1)).$$

To simplify notation, let $F^{-1}(\theta) = s$ and $\tau\log(t^{-1}-1) = a$. Then, $h(t,\theta) = F(s+a)$ and $h(1-t,\theta) = F(s-a)$. Let us first consider the case $s < 0$ and $a > 0$. We see that

$$F(s+a) - F(s) = \int_s^{s+a} F'(m)\,dm \quad \text{and} \quad F(s) - F(s-a) = \int_{s-a}^{s} F'(m)\,dm.$$

For any $b > 0$, if $s + b < 0$, then by monotonicity, $F'(s+b) > F'(s-b)$. On the other hand, if $s + b \geq 0$, then $F'(s+b) = F'(-s-b) > F'(s-b)$. In both cases, the right integral is always smaller than the left integral. In other words,

$$F(s+a) + F(s-a) > 2F(s).$$

In fact, the above inequality is also established when $s < 0$ and $a < 0$. Therefore, whenever $s < 0$,

$$\int_0^1 h(t,\theta)\,dt = \int_0^1 \frac{h(t,\theta) + h(1-t,\theta)}{2}\,dt > \int_0^1 F(F^{-1}(\theta))\,dt = \theta.$$

That is, $\mathrm{Bias}(z) > 0$. Other cases ($s = F^{-1}(\theta) > 0$ and $s = F^{-1}(\theta) = 0$) are similarly proved.

4.2 EMPIRICAL COMPARISON BETWEEN GUMBEL AND ICDF REPARAMETERIZATION

Extending the last experiment in Section 7, Table 1 summarizes the time and memory consumption during the training of global models on the four data sets. The results indicate that our developed **ICDF** reparameterization is more economic than the Gumbel-Softmax approach.

Table 1: Time and memory consumption of F^3 (five epoches) with respect to **ICDF** and Gumbel-Softmax reparameterization. Time is in seconds and memory is in MB.

	METR-LA		PEMS-BAY		PMU-B		PMU-C	
	Time	Memory	Time	Memory	Time	Memory	Time	Memory
Gumbel-Softmax	87.89	832.38	270.52	1896.11	42.40	348.39	84.89	1119.13
ICDF	79.69	568.24	157.93	1167.19	30.16	322.59	54.07	894.63

5 DATA SET DESCRIPTION AND PREPROCESSING

METR-LA and **PEMS-BAY.** These are traffic data sets (MIT licensed) used by Li et al. [2018]. The former was collected from loop detectors in the highway of Los Angles, CA [Jagadish et al., 2014] and the latter was collected by the California Transportation Agencies Performance Measure System. Both data sets recorded several months of data at the resolution of five minutes. The network graphs are available, which were constructed by imposing a radial basis function on the pairwise distance of sensors at a certain cutoff.

The data sets were originally prepared for forecasting tasks and hence no labeling information exists. We adapt the data for classification. Specifically, we split the time series on the hour, forming hourly windows. We label each window as whether or not it corresponds to rush hour. For proof of concept, we specify 07:00–10:00 and 16:00–19:00 as rush hour and the others non-rush hour. We note that in the original data sets, one of the attributes is time. We remove this attribute to avoid triviality and retain only the speed attribute.

The specification of rush hours may not be highly accurate, but it is a sensible practice to cope with the nonexistence of labeling information. Intuitively, the signal of rush hour comes from reduced traffic speed, but not every location of the network experiences traffic jam. Hence, the diverse traffic patterns inside the same time window under a single label causes nontrivial challenges for local models to discern. Therefore, the need of a global consensus model is justified and it fits well the federated feature fusion scenario.

PMU-B and **PMU-C.** These are proprietary data sets coordinately provided by multiple data owners of the U.S. power grid. No personally identifiable information is present. The suffixes B and C indicate the interconnects of the grid. The data sets come with thousands of annotated grid events spanning a period of two years; they form the classification labels. Many variables (attributes) of the grid condition are recorded; we select only the voltage magnitude and the current magnitude, because they appear to be the strongest signals for event detection based on domain knowledge, and also because more data are available for these two variables. The grid topology is not available.

For each event, we select a one-second window from the three-minute window that covers the approximate annotated event time, based on the largest z-score. We retain a sampling frequency of 30Hz, even though some data are 60Hz. Furthermore, a large amount of data are missing in the raw data. We impute the series by using `pandas.DataFrame.interpolate(method = 'linear', limit_direction = 'both')` from the Python `pandas` package. This way, a windowed series is complete if it ever has raw data. Even so, many series are entirely empty, which corresponds to the scenario illustrated by Figure 1 of the main text. Classes in these two data sets are rather skewed. For PMU-B, we remove a class that consists of only one data point and for PMU-C, we combine classes that contain fewer than 24 data points into a single class.

6 EXPERIMENT DETAILS

The experiments are conducted on one x86 node of a computing cluster with one A100 NVIDIA GPU. The compute node has eight Intel cores and 128GB memory. For each data set, we perform a 70/10/20 random split for training, validation, and testing, respectively.

For local models, we use LSTM with the same hyperparameters: one hidden layer whose hidden dimension is 16 and the maximum number of epochs $= 200$. We pre-train the local models and freeze their parameters afterward. We train each global model for a maximum of 500 epochs and use early stopping according to the validation loss, with a patience of 50 epochs. For the GNN global model, we use a 2-layer GCN with skip connections. The hidden dimension is set at 8 and we select the learning rate from $\{0.01, 0.001\}$. For missing data, we impute the node features by using zero.

Table 2: Different approaches to feature alignment.

	METR-LA		PEMS-BAY		PMU-B		PMU-C	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
soft alignment	.835	.975	.860	.980	.390	.734	.451	.725
hard alignment	.839	.973	.855	.976	.390	.737	.429	.721

7 SOFT AND HARD FEATURE ALIGNMENT

Feature alignment can be achieved in two manners. The first approach is a soft alignment, which treats each $\mathbf{P}_i$ a free parameter matrix to optimize. Such an alignment softens the one-to-one correspondence in the permutation constraint; i.e., each feature in the source can have a weighted correspondence to each of the features in the target. That is the way we used in the main paper.

An alternative approach is a hard alignment, which treats each $\mathbf{P}_i$ as a permutation matrix. Learning permutation matrices is challenging, however, because they correspond to combinatorial structures and are unsuitable for gradient-based training. We follow Mena et al. [2018], Emami and Ranka [2018] and relax $\mathbf{P}_i$ by a doubly stochastic matrix, which can be differentiably parameterized by the Sinkhorn–Knopp algorithm [Sinkhorn and Knopp, 1967]. Specifically, starting from a nonnegative square matrix $\mathbf{K}_0$ and column vectors $\mathbf{r}_0 = \mathbf{c}_0 = \mathbf{1}$ of matching lengths, define the sequence

$$\mathbf{c}_{j+1} = \mathbf{1} \oslash (\mathbf{K}_0^T \mathbf{r}_j) \text{ and } \mathbf{r}_{j+1} = \mathbf{1} \oslash (\mathbf{K}_0 \mathbf{c}_j), \quad \text{for } j = 0, 1, \ldots \tag{10}$$

Then, under a mild condition, $\mathbf{K}_j := \operatorname{diag}(\mathbf{r}_j)\mathbf{K}_0 \operatorname{diag}(\mathbf{c}_j)$ converges to a doubly stochastic matrix. We truncate the sequence at the Tth step and treat $\mathbf{K}_T$ as an approximation of $\mathbf{P}_i$.

Despite the advocation by Mena et al. [2018], Emami and Ranka [2018], we obtain the following convergence result of Sinkhorn–Knopp, which reveals no free lunch.

Theorem 4 (informal)**.** Under a condition of $\mathbf{K}_0$, there exists a positive integer J and a constant C_J such that for all $j \geq J$,

$$\left\| \begin{bmatrix} \mathbf{K}_j^T \mathbf{1} \\ \mathbf{K}_j \mathbf{1} \end{bmatrix} - \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \right\| \leq C_J (1 + \sigma_2^2) \sigma_2^{2(j-J)},$$

where $\sigma_2 \leq 1$ is the second largest singular value of the limit of $\mathbf{K}_j$.

Since this is not the focus of this paper, we omit the rigorous analysis of this theorem. The result suggests that for a desirable limit being a permutation matrix, whose $\sigma_2 = 1$, the error $O(\sigma_2^{2j})$ does not drop. In practice, to expect for an approximate permutation matrix, $\sigma_2 \approx 1$ and the convergence is exceedingly slow. The practical usefulness of (10) depends on the learned quality of $\mathbf{K}_0$.

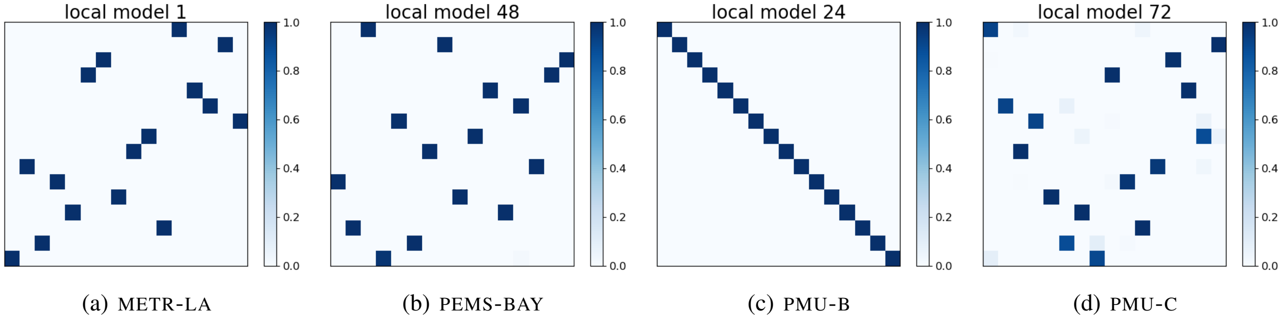
The soft and hard alignment approaches have pros and cons. The hard approach maintains the correspondence of each feature dimension of the latent vectors while the soft approach does not. Maintaining the dimension correspondence is an advantage, especially for local models that produce disentangled latent representations [Higgins et al., 2018], because each feature dimension is equipped with a semantic meaning that controls a certain aspect of the data. On the other hand, the soft approach is more straightforward and the hard approach is based on an algorithm that barely converges. In practice, we observe that two approaches deliver similar performance. We list the results for both approaches in Table 2, and we visualize the hard alignment matrix learned on each dataset Figure 1, to help readers understand the feature alignment.

References

Patrick Emami and Sanjay Ranka. Learning permutations with Sinkhorn policy gradient. Preprint arXiv:1805.07010, 2018.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. Preprint arXiv:1812.02230, 2018.

H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, 2014.



(a) METR-LA (b) PEMS-BAY (c) PMU-B (d) PMU-C

Figure 1: Examples of learned permutation matrices ($\mathbf{K}_T$). The plots clearly show patterns of a permutation matrix: there is one and only one significant value per row and per column. Because of the slow convergence, we attribute the desirable results of $\mathbf{K}_T$ (at a small T) to the success of the learning of $\mathbf{K}_0$. Note also interestingly that a learned permutation may be the identity mapping.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *ICLR*, 2018.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with Gumbel-Sinkhorn networks. Preprint arXiv:1802.08665, 2018.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21 (2):343–348, 1967.