Sliced-Wasserstein Importance Weighting for Robust Brain-Computer Interface Speech Decoding

Noah Cowan
Department of Statistics
Stanford University
Stanford, CA 94305
ncowan@stanford.edu

Scott Linderman
Department of Statistics
Stanford University
Stanford, CA 94305
scott.linderman@stanford.edu

Abstract

Brain–computer interfaces (BCIs) hold transformative potential, but their performance often degrades across sessions due to signal drift and calibration challenges. In this paper, we propose a method to improve cross-session robustness by reweighting training data according to their similarity to the target session, as measured with the Sliced-Wasserstein distance. We provide theoretical justification for this approach in a simplified statistical model, and we evaluate it on real BCI data. Our results show that Sliced-Wasserstein weighting improves BCI performance by reducing phoneme error rate from 0.296 to 0.169 (a 42.9% reduction) on the first post-training session, and it maintains nearly the same level of performance over the following three sessions. Our results suggest that distributionally informed reweighting offers a principled and fully unsupervised way to mitigate session-to-session variability in BCIs, paving the way toward more reliable long-term neural decoding without the need for costly recalibration.

1 Introduction

Brain-computer interfaces hold great promise to return the power of language to those who have lost it by decoding their neural activity into text and speech Herff et al. [2015], Anumanchipalli et al. [2019], Moses et al. [2021], Willett et al. [2023]. Unfortunately,current BCIs exhibit performance degradation over time. This instability can arise due to array movement Santhanam et al. [2007], behavioral change Degenhart et al. [2020], or changes in neuron connectivity Churchland and Shenoy [2007]. These phenomena force BCIs to be re-calibrated very frequently, sometimes even daily, to ensure acceptable error rates Willett et al. [2023].

A variety of methods have been used to circumvent BCI instability, including learning a low dimensional neural-representation of the data Nuyujukian et al. [2014], Degenhart et al. [2020], fitting low-degree polynomials to estimate and correct for covariate shift Satti et al. [2010], and applying exponential time weighting mechanism Orsborn et al. [2012]. While these methods capture the idea that alignment across sessions is key to stabilization, they generally require strong parametric assumptions, remain fragile in the low-sample, high-dimensional regime of BCIs, and still demand frequent recalibration.

In this work, we take a different approach: instead of imposing parametric models, we align sessions by reweighting training data based on empirical distributional similarity. Specifically, we extend the exponential weighting model of Orsborn et al. [2012] by introducing weights derived from the Sliced-Wasserstein distance (SWD) Bonneel et al. [2015] between sessions.

Although applications of SWD in neuroscience are still emerging, prior work has shown its promise. Bonet et al. [2023] define a matrix-valued SWD and show that it serves as an efficient surrogate

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Data on the Brain & Mind.

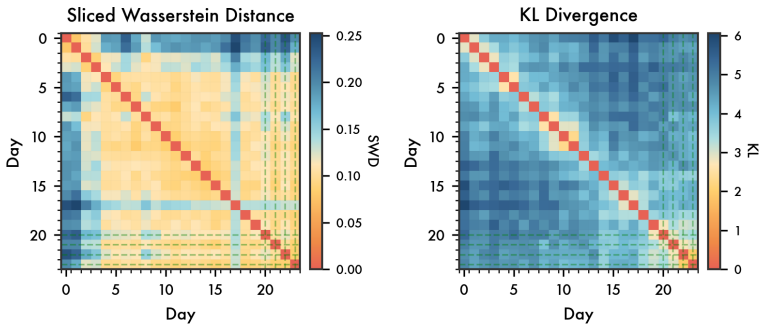

Figure 1: While both KL and SWD see distance increase as sessions get farther apart temporally, SWD shows more bands of entire far away days. This can help ignore days that produced data completely different from the current day's data. We also see significantly more variation in the SWD matrix, highlighting that it can differentiation useful data more effectively.

to Wasserstein for BCI domain adaptation in M/EEG data. Beyond BCIs, Sliced-Wasserstein has been applied to mitigate distribution shift more broadly: Lee et al. [2019] propose a SWD based discrepancy that aligns source and target domains by matching classifier outputs. Our work differs by using SWD as a similarity-driven importance-weighting scheme to pool cross-day data in intracortical speech BCI, rather than learning an explicit alignment map.

Because our method relies solely on unlabeled neural data for alignment, it can directly reduce recalibration demands and extend the useful lifespan of a trained decoder. As shown in Pun et al. [2024], decoder accuracy is strongly correlated with the statistical distance between training and deployment data. This suggests that alignment based on a principled distance metric such as SWD can improve cross-session performance. Bischoff et al. [2024] provide an extended review of various sample-based statistical distances, including SWD.

2 Problem Setup

We consider the problem of decoding intended speech from intra-corticular neural activity data found in Willett et al. [2023]. Letting the neural data for a timestep j on day i be written as $X_{i,j}$ where $X_{i,j} \in \mathcal{X} = \mathbb{N}^p$, $p = 256$, and $Y_{i,j} \in \mathcal{Y} = \{1, 2, \ldots, K\}$ be the label (in this case, phoneme) corresponding to timestep $j \in \{0, \ldots, n_i - 1\}$, we set up the problem as a distribution shift problem where the distribution of neural recordings change on each session day. Our dataset contains neural recordings from multiple training sessions from a single participant. For each training day $i \in \{0, 1, \ldots, N-1\}$ we get neural data of length n_i and corresponding phonemes of length m_i, where $m_i < n_i$.

We make the standard assumption that both the joint $P_i(x, y)$ and the marginal of the neural data $P_i(x)$ are day specific, they vary across recording sessions. Because domain shift causes significant performance degradation across days, it is necessary to correct for it. We denote the empirical distribution of the neural data for day i as $\widehat{P}_d(x) = \frac{1}{T_d} \sum_{i=1}^{T_d} \mathbf{1}(X_i = x)$, where T_d is the number of samples collected that day. In practice, we have access to empirical distributions $\widehat{P}_i(x)$ for the past training days (in our example days zero through 19), along with a small number of samples from the test day distributions $P_t(x)$ (days 20 through 23).

Our goal is to learn a decoder $f_\theta : \mathcal{X}^{n_i} \to \mathcal{Y}^{n_i}$ that outputs predicted labels for each timestep and performs well on the target days $t \in \mathcal{T}$ using training data and the limited target day data. Under standard empirical risk minimization, all training days are weighted equally, setting θ such that $\theta^* = \arg\min_\theta \sum_{i=0}^{N_{\text{train}}-1} \sum_{j=1}^{n_i} \ell(f_\theta(x_{i,j}), y_{i,j})$ where $\ell(\cdot, \cdot)$ is the CTC (Connectionist Temporal Classification) loss function. CTC is appropriate because phoneme sequences are shorter than the observed timesteps, and their alignments are unknown. While the model is trained using the CTC loss, we evaluate performance using phoneme error rate (PER), which directly measures decoding accuracy.

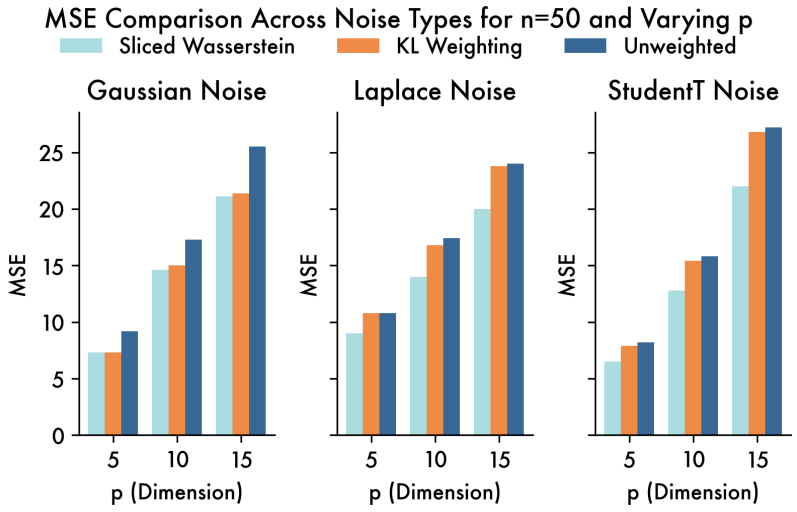

Figure 2: In the synthetic example outlined in section 4.1 KL performs worse than SWD under mis-specification of the model.

However, this approach fails to account for the varying degrees of domain shift between source days i and the target day t. Intuitively, training samples from days whose neural distributions are more similar to the target day should receive higher importance during training.

3 Method

We propose a principled adaptation method that weighs training samples based on distributional similarity between source and target days, measured using the Sliced-Wasserstein Distance (SWD, defined in Appendix C). SWD provides a low-variance proxy for the Wasserstein distance that remains informative in high dimensions. It captures the geometric information of how far one must move probability mass in order to match another density, while still retaining the notion of closeness that typically disappears in high dimensions. We see in Figure 1 that the SWD and KL divergence capture different information with more bands of outlier days in SWD.

To use the information given by the SWD, for each target day t, we compute a weight $w_{i,t}$ for each source day i that quantifies the similarity between their neural distributions. The weighted training goal is to find θ satisfying

$$\theta^* = \arg\min_{\theta} \sum_{i=0}^{N_{\text{train}}-1} w_{i,t} \sum_{j=1}^{n_i} \ell(f_\theta(x_{i,j}), y_{i,j}) \tag{1}$$

This formulation directs learning toward neural patterns whose distributions are closest to the target day, thereby improving generalization under domain shift.

To find the optimal weighting scheme for this problem, we noted that in a linear model with covariates and parameters drifting as Gaussian random walks, the Bayesian motivated weights with the Jeffreys prior in a weighted linear regression would be (up to an approximation of the Lambert W function) $w_{t,i} = \frac{1}{1+2D_{KL}((X_t,Y_t)\|(X_i,Y_i)|\beta_i)}$ as derived in the linear-drift analysis in Appendix B.

Because the scales of SWD and KL can differ by orders of magnitude, we introduce a tunable hyperparameter λ and define the weights

$$w_{t,i} = \frac{1}{1 + \frac{1}{\lambda} D_{SWD}(X_t, X_i)}, \quad \tilde{w}_{t,i} = \frac{\max(w_{t,i}, \epsilon)}{\sum_k \max(w_{t,k}, \epsilon)}.$$

The parameter λ controls the strength of weighting. In practice, setting λ to half the ratio of the average KL divergence to the average SWD yields weights on a comparable scale to the Bayesian-

optimal solution in the linear-drift setting, without requiring assumptions about the data-generating process. We also include a small constant $\epsilon > 0$ to ensure that no day is completely ignored.

4 Experiments

4.1 Synthetic Results

To get a sense of when SWD would perform well, we start with a linear model. For each day, $Y = X\beta + \epsilon$ with X and β generated from a multivariate normal distribution with a mean that changes incrementally over days. Each day has different parameters and for the multivariate normal distribution and the linear model, respectively. and change incrementally based on previous day's values and some noise, representing changes over time. That is $\beta_{i+1} \sim \mathcal{N}(\beta_i, \tau^2 I)$ and $Y_{j,i} \sim \mathcal{N}(\beta_i^T X_{j,i}, \sigma^2 I)$. We see in Figure 2 that even in the Gaussian regime, a non-parametric version of the KL divergence Perez-Cruz [2008] weights do not outperform weighting based on Sliced-Wasserstein distance in this model. The SWD based weights handle model misspecification much better than the KL weights and both outperform an unweighted scheme. Note that an implementation of KL weighting that uses the Gaussian assumption to also calculate the KL divergence gets nearly 0 MSE for the Gaussian case (matching the theory that it would be optimal), but has MSE significantly worse than even the unweighted for Laplace noise, making it hard to justify its use or inclusion here.

4.2 BCI Results

To apply the SWD weighting to the BCI data, we use 100 random projections and then calculate the Sliced-1-Wasserstein distance using those projections. To make the SWD weighting of a similar scale to the theoretically motivated KL, we set $\lambda = .03$ for the Sliced-Wasserstein weights and applied them to the processed spike counts from Willett et al. [2023]. For each day, we retrain a model after seeing 10% of the (unlabeled) data for the day and train a model using weighted samples based on the distance from the empirical distribution of the neural data observed. We get large performance gains over the baseline GRU, as seen in Figure 3. Not only does the SWD-weighted RNN outperform on the first session after training, while using only 10 percent of the data to learn the weights, it also keeps performance steady for multiple sessions after the GRU has already started to significantly degrade. Although we acknowledge these results come from just a single participant and a limited number of sessions, they consistently show the ability of SWD to stabilize BCI performance.

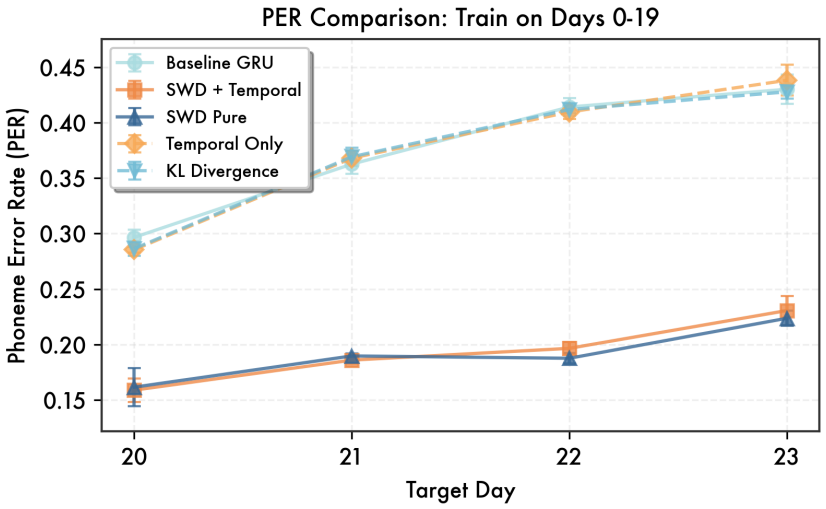

Figure 3: Comparison of five different methods for speech decoding. Labeled training data stops at day 19. Lower PER is better, dataset from Willett et al. [2023]. We see that the Sliced-Wasserstein based weighting scheme outperforms all other weighting schemes and the baseline by a sizable margin and does not require explicit incorporation of temporal based distance.

References

Gopala K. Anumanchipalli, Josh Chartier, and Edward F. Chang. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1119-1.

Sebastian Bischoff, Alana Darcher, Michael Deistler, Richard Gao, Franziska Gerken, Manuel Gloeckler, Lisa Haxel, Jaivardhan Kapoor, Janne K Lappalainen, Jakob H Macke, et al. A practical guide to sample-based statistical distances for evaluating generative models in science. *arXiv preprint arXiv:2403.12636*, 2024.

Clément Bonet, Benoît Malézieux, Alain Rakotomamonjy, Lucas Drumetz, Thomas Moreau, Matthieu Kowalski, and Nicolas Courty. Sliced-Wasserstein on Symmetric Positive Definite Matrices for M/EEG Signals. In *Proceedings of the 40th International Conference on Machine Learning*, pages 2777–2805. PMLR, 2023.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, January 2015. ISSN 1573-7683. doi: 10.1007/s10851-014-0506-3. URL `https://doi.org/10.1007/s10851-014-0506-3`.

Mark M. Churchland and Krishna V. Shenoy. Temporal Complexity and Heterogeneity of Single-Neuron Activity in Premotor and Motor Cortex. *Journal of Neurophysiology*, 97(6):4235–4257, 2007. ISSN 0022-3077. doi: 10.1152/jn.00095.2007.

Alan D. Degenhart, William E. Bishop, Emily R. Oby, Elizabeth C. Tyler-Kabara, Steven M. Chase, Aaron P. Batista, and Byron M. Yu. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature Biomedical Engineering*, 4(7):672–685, 2020. ISSN 2157-846X. doi: 10.1038/s41551-020-0542-9.

John Duchi. Derivations for Linear Algebra and Optimization.

Christian Herff, Dominic Heger, Adriana de Pesters, Dominic Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz. Brain-to-text: Decoding spoken phrases from phone representations in the brain. *Frontiers in Neuroscience*, 8, 2015. ISSN 1662-453X. doi: 10.3389/fnins.2015.00217.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced Wasserstein Discrepancy for Unsupervised Domain Adaptation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10277–10287, 2019. doi: 10.1109/CVPR.2019.01053.

David A. Moses, Sean L. Metzger, Jessie R. Liu, Gopala K. Anumanchipalli, Joseph G. Makin, Pengfei F. Sun, Josh Chartier, Maximilian E. Dougherty, Patricia M. Liu, Gary M. Abrams, Adelyn Tu-Chan, Karunesh Ganguly, and Edward F. Chang. Neuroprosthesis for Decoding Speech in a Paralyzed Person with Anarthria. *New England Journal of Medicine*, 385(3):217–227, 2021. ISSN 0028-4793. doi: 10.1056/NEJMoa2027540.

Paul Nuyujukian, Jonathan C. Kao, Joline M. Fan, Sergey D. Stavisky, Stephen I. Ryu, and Krishna V. Shenoy. Performance sustaining intracortical neural prostheses. *Journal of Neural Engineering*, 11 (6):066003, 2014. ISSN 1741-2552. doi: 10.1088/1741-2560/11/6/066003.

Amy L. Orsborn, Siddharth Dangi, Helene G. Moorman, and Jose M. Carmena. Closed-Loop Decoder Adaptation on Intermediate Time-Scales Facilitates Rapid BMI Performance Improvements Independent of Decoder Initialization Conditions. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(4):468–477, 2012. ISSN 1558-0210. doi: 10.1109/TNSRE.2012.2185066.

Fernando Perez-Cruz. Kullback-Leibler divergence estimation of continuous distributions. In *2008 IEEE International Symposium on Information Theory*, pages 1666–1670, July 2008. doi: 10.1109/ISIT.2008.4595271. URL `https://ieeexplore-ieee-org.stanford.idm.oclc.org/document/4595271`. ISSN: 2157-8117.

Tsam Kiu Pun, Mona Khoshnevis, Tommy Hosman, Guy H. Wilson, Anastasia Kapitonava, Foram Kamdar, Jaimie M. Henderson, John D. Simeral, Carlos E. Vargas-Irwin, Matthew T. Harrison, and Leigh R. Hochberg. Measuring instability in chronic human intracortical neural recordings towards stable, long-term brain-computer interfaces. *Communications Biology*, 7(1):1–14, 2024. ISSN 2399-3642. doi: 10.1038/s42003-024-06784-4.

Gopal Santhanam, Michael D. Linderman, Vikash Gilja, Afsheen Afshar, Stephen I. Ryu, Teresa H. Meng, and Krishna V. Shenoy. HermesB: A continuous neural recording system for freely behaving primates. *IEEE transactions on bio-medical engineering*, 54(11):2037–2050, 2007. ISSN 0018-9294. doi: 10.1109/TBME.2007.895753.

Abdul Satti, Cuntai Guan, Damien Coyle, and Girijesh Prasad. A Covariate Shift Minimisation Method to Alleviate Non-stationarity Effects for an Adaptive Brain-Computer Interface. In *2010 20th International Conference on Pattern Recognition*, pages 105–108, 2010. doi: 10.1109/ICPR.2010.34.

Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Matthew F. Glasser, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 620 (7976):1031–1036, 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06377-x.

A Alternative Methods

Although we only presented our own method in section 3, we also implemented three others to see if it was merely the act of weighting data that improved performance, or whether it was the Sliced-Wasserstein specifically that led to improvements. Note that for each method, if it requires access to target day data, it is limited to just 10%. This is a small enough fraction that the user would likely still be able to use the BCI normally.

The temporal weights were calculated using the absolute value of the day indices as the "distance" between two days (i.e. day five is two away from day seven). Other than that the specification of the weights is the same as $\frac{1}{1+\frac{1}{\lambda}D}$. The only difference is the choice of λ for different choices of distance.

Method	λ
Sliced Wasserstein	0.03
KL Divergence	0.5
Temporal	1

Table 1: Comparison of λ values across methods.

Then for SWD + Temporal, they are combined as the simple multiplication of the SWD weight and the temporal weight for that day and then normalized.

B KL Divergence Weighting

Using the same generative model as in section 4.1, we calculate the optimal bayesian weightings.

B.1 Optimal Bayesian estimate

Let $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{y} = (y_1, \ldots, y_d)$

$$\begin{aligned} p(\beta_d \mid \mathbf{x}, \mathbf{y}) &\propto p(\beta_d)\, p(\mathbf{y} \mid \beta_d, \mathbf{x}) \\ &\propto p(\beta_d) \prod_{i=1}^{d} p(y_i \mid \beta_d, x_i, \mathbf{x}_{<i}, \mathbf{y}_{<i}) \end{aligned}$$

Let's forget about the dependence on $\mathbf{x}_{<i}$ and $\mathbf{y}_{<i}$, and treat each observation as conditionally independent. Then,

$$\begin{aligned} p(\beta_d \mid \mathbf{x}, \mathbf{y}) &\approx p(\beta_d) \prod_{i=1}^{d} p(y_i \mid \beta_d, x_i) \\ &= p(\beta_d) \prod_{i=1}^{d} \int p(y_i \mid \beta_i, x_i)\, p(\beta_i \mid \beta_d)\, \mathrm{d}\beta_i \end{aligned}$$

Under the random walk model, the conditional distribution of β_i is,

$$p(\beta_i \mid \beta_d) = \mathcal{N}(\beta_d, (d-i)\tau^2 I)$$

Then the marginal likelihood of y_i is,

$$\begin{aligned} p(y_i \mid x_i, \beta_d) &= \int p(y_i \mid \beta_i, x_i)\, p(\beta_i \mid \beta_d)\, \mathrm{d}\beta_i \\ &= \int \mathcal{N}(y_i \mid \beta_i^\top x_i, \sigma^2)\, \mathcal{N}(\beta_i \mid \beta_d, (d-i)\tau^2 I)\, \mathrm{d}\beta_i \\ &= \mathcal{N}(y_i \mid \beta_d^\top x_i, (d-i)\tau^2 x_i^\top x_i + \sigma^2) \end{aligned}$$

Putting it all together and assuming an uninformative prior,

$$\begin{aligned} p(\beta_d \mid \mathbf{x}, \mathbf{y}) &\approx \prod_{i=1}^{d} \mathcal{N}(y_i \mid \beta_d^\top x_i, (d-i)\tau^2 x_i^\top x_i + \sigma^2) \\ &\propto \mathcal{N}(\beta_d \mid \mu_d, \Sigma_d) \end{aligned}$$

where

$$\begin{aligned} w_i &= [(d-i)\tau^2 x_i^\top x_i + \sigma^2]^{-1} \\ J_d &= \sum_{i=1}^{d} w_i x_i x_i^\top \\ h_d &= \sum_{i=1}^{d} w_i x_i y_i \\ \Sigma_d &= J_d^{-1} \\ \mu_d &= J_d^{-1} h_d = \left(\sum_{i=1}^{d} w_i x_i x_i^\top \right)^{-1} \left(\sum_{i=1}^{d} w_i x_i y_i \right) \end{aligned}$$

Or in more familiar terms, this is a weighted least squares estimate of β_d with weights w_i. If you assume the inputs are unit norm then the weights simplify to $w_i \propto [1 + (d-i)\frac{\tau^2}{\sigma^2}]^{-1}$

B.2 KL Divergence Between Days

Note that we know very quickly the distribution of the joint (X_d, Y_d) on a given day, **assuming the X are generated from $n \times p$ standard normals**. Then, assuming WLOG that $d' > d$, we know that the joint of the $(X_{d'}, Y_{d'})$ for day d is

$$\begin{pmatrix} X_{d'} \\ Y_{d'} \end{pmatrix} | \beta_d \sim \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} I_p & \beta_d \\ \beta_d^\top & \sigma^2 + p(d'-d)\tau^2 + \|\beta_d\|^2 \end{pmatrix} \right)$$

Hence using the known equation that for two multivariate normals the KL divergence Duchi is

$$D_{KL}(P_1 \| P_2) = \frac{1}{2} \left(\log|\Sigma_2| - \log|\Sigma_1| - (p+1) + (\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^\top \Sigma_2^{-1}(\mu_2 - \mu_1) \right)$$

Since $\mu_{d'} = \mu_d = \beta_d$, we get that the last term is 0 and using the Schur complement we know that $|\Sigma_{d'}| = \det(I)\det(\sigma^2 + p(d'-d)\tau^2 + \|\beta_d\|^2 - \beta_d^\top \beta_d) = \sigma^2 + p(d'-d)\tau^2$. Finally, again using the Schur complement (which for Σ_d is $S = \sigma^2 + \|\beta_d\|^2 - \|\beta_d\|^2 = \sigma^2$) we see that

$$\Sigma_d^{-1} = \begin{pmatrix} I_p + \frac{1}{\sigma^2}\beta_d\beta_d^\top & -\frac{1}{\sigma^2}\beta_d \\ -\frac{1}{\sigma^2}\beta_d^\top & \frac{1}{\sigma^2} \end{pmatrix}$$

Thus

$$(\Sigma_d^{-1}\Sigma_{d'}) = (I_p) + 1 + \frac{\tau^2}{\sigma^2}p(d'-d) = p + 1 + \frac{\tau^2}{\sigma^2}p(d'-d)$$

we get

$$\begin{aligned} D_{KL}((X_{d'}, Y_{d'})\|(X_d, Y_d)|\beta_d) &= \frac{1}{2}\left(\log(\sigma^2) - \log(\sigma^2 + p(d'-d)\tau^2) - (p+1) + (p+1) + \frac{\tau^2}{\sigma^2}p(d'-d)\right) \\ &= \frac{1}{2}\left(\log(\sigma^2) - \log(\sigma^2 + p(d'-d)\tau^2) + \frac{\tau^2}{\sigma^2}p(d'-d)\right) \\ &= \frac{1}{2}\left(\frac{\tau^2}{\sigma^2}p(d'-d) - \log\left(1 + \frac{\tau^2}{\sigma^2}p(d'-d)\right)\right) \end{aligned}$$

Thus, we can see that up to a log factor $w_i = \frac{1}{1+2D_{KL}((X_d,Y_d)\|(X_i,Y_i)|\beta_i)}$ so using an inverse KL divergence should get something close to the optimal weighting. Alternatively, note that the Lambert W function is defined as the function such that $W(x)e^{W(x)} = x$. Using this we get

$$\begin{aligned} D_{KL}((X_{d'}, Y_{d'})\|(X_d, Y_d)|\beta_d) &= \frac{1}{2}(\frac{\tau^2}{\sigma^2}p(d'-d) - \log(1 + \frac{\tau^2}{\sigma^2}p(d'-d)) \\ w_d &= \frac{1}{1 + \frac{\tau^2}{\sigma^2}p(d'-d)} \end{aligned}$$

So we have

$$\begin{aligned} 1 + 2D_{KL} &= \frac{1}{w_d} + \log w_d \\ e^{-(1+2D_{KL})} &= \frac{1}{w_d}e^{-\frac{1}{w_d}} \\ -e^{-(1+2D_{KL})} &= -\frac{1}{w_d}e^{-\frac{1}{w_d}} \end{aligned}$$

Which implies by the definition of the Lambert W function that

$$-\frac{1}{w_d} = W_0(-e^{-(1+2D_{KL})}) \implies w_d = -\frac{1}{W_0(-e^{-(1+2D_{KL})})}$$

C Sliced-Wasserstein Distance

Definition (Sliced-Wasserstein distance). Let μ, ν be probability measures on $\mathbb{R}^p$ with finite cth moments and let W_c denote the 1D c-Wasserstein distance. For a unit vector $\theta \in \mathbb{S}^{p-1}$, define the 1D projection $P_\theta(x) = \theta^\top x$ and the pushforward $(P_\theta)_\#\mu$. The Sliced-Wasserstein distance is

$$D_{\mathrm{SW}_c}(\mu,\nu) = \Big(\int_{\mathbb{S}^{p-1}} W_c\big((P_\theta)_\#\mu,\,(P_\theta)_\#\nu\big)^c\, d\sigma(\theta)\Big)^{1/c},$$

where σ is the uniform measure on the unit sphere. In practice we estimate the integral by Monte Carlo over L random directions.

Empirical estimator used in this paper. Given samples $X = \{x_i\}_{i=1}^n$, $Y = \{y_j\}_{j=1}^m$, draw $\{\theta_\ell\}_{\ell=1}^L \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(\mathbb{S}^{p-1})$, compute projected samples $a^{(\ell)} = \mathrm{sort}(X\theta_\ell)$, $b^{(\ell)} = \mathrm{sort}(Y\theta_\ell)$ (with linear quantile interpolation if $n \neq m$), and set

$$\widehat{D_{\mathrm{SW}_c}}^c(X,Y) = \frac{1}{L}\sum_{\ell=1}^{L}\frac{1}{M}\sum_{k=1}^{M}\big|a_k^{(\ell)} - b_k^{(\ell)}\big|^c, \quad M = \min(n,m).$$

We use the monte carlo estimator with $c = 1$ and $L = 100$ unless stated otherwise; the computational cost is $\mathcal{O}\big(L\,[(n+m)d + (n\log n + m\log m)]\big)$.