1 Appendix:

1.1 Recovery of the Unobserved Biological Signal - Uber

Theorem 1. *Suppose we train an autoencoder model to achieve zero generalization loss, i.e.,* $\hat{X} = X$*, and impose on the biological code that* $S' \perp\!\!\!\perp T$*. Then*

$$H(S) = H(S').$$

Proof. Let S denote the unobserved biological signal, T the batch label, and S' the learned latent code.

Since $S \perp\!\!\!\perp T$, we have

$$H(S \mid S') = H(S \mid S', T) = H(S, T \mid S', T).$$

Given perfect reconstruction and that $X = f(S, T)$ with f invertible, we obtain:

$$H(S, T \mid S', T) = H(\hat{X} \mid S', T).$$

But since $\hat{X} = D(S', T)$, a deterministic function, it follows that:

$$H(\hat{X} \mid S', T) = 0.$$

Therefore:

$$H(S \mid S') = 0 \quad \Rightarrow \quad I(S; S') = H(S).$$

By symmetry of mutual information and the bound $I(S; S') \leq H(S')$, we have:

$$H(S') \geq I(S; S') = H(S) \quad \text{and} \quad H(S') \leq H(S),$$

which implies:

$$H(S') = H(S).$$

Thus, the latent code S' contains all the information present in S. □

Proposition 2. *Under the assumption of 1. Let* $A : \mathcal{S} \to \mathcal{S}'$ *be a transformation from the biological space* $\mathcal{S}$ *to the latent space* $\mathcal{S}'$*, where* $\phi = A$ *is the encoder. Assume that the decoder* $D : \mathcal{S}' \times \mathcal{T} \to \mathcal{X}$ *satisfies perfect reconstruction:*

$$D(A(S), t) = f(S, t) \quad \forall S \in \mathcal{S},\ t \in \mathcal{T},$$

where $f : \mathcal{S} \times \mathcal{T} \to \mathcal{X}$ *is the true generative function. Then* A *is invertible.*

Proof. To establish invertibility, we demonstrate that A is both injective and surjective.

Injectivity: Suppose $A(S_1) = A(S_2)$. Then for any batch t,

$$f(S_1, t) = D(A(S_1), t) = D(A(S_2), t) = f(S_2, t).$$

Assuming the generative model f is informative in S, meaning $f(S_1, t) = f(S_2, t)$ for all t implies $S_1 = S_2$, it follows that A is injective.

Surjectivity: By assumption, A maps $\mathcal{S}$ onto $\mathcal{S}'$, i.e., every point in the latent space has a pre-image in the biological space. Therefore, A is surjective.

Conclusion: Since A is both injective and surjective, it is bijective, and an inverse $A^{-1} : \mathcal{S}' \to \mathcal{S}$ exists. □

Corollary 3 (Batch Synthesis)**.** *Let $x = f(s,t)$ be a sample with biological state s and batch label t. Let $x' = f(s,t')$ be another sample with the same biological state but a new batch label t'. Under the assumptions of Theorem 1 and Proposition 2, the model can synthesize x' from the encoding of x and the target batch label t'. Formally:*

$$D(\phi(x), t') = x'.$$

Proof. The encoder ϕ maps the input data $x = f(s,t)$ to a latent representation. From Theorem 1, we know this learned latent code, denoted s', captures all the information in the biological signal s. We can represent this mapping from the true biological signal to the latent code as a transformation A, such that

$$s' = \phi(x) = A(s).$$

The decoder D is capable of perfect reconstruction. According to Proposition 2, for any biological signal S and batch label t, the decoder satisfies:

$$D(A(S), t) = f(S, t).$$

To synthesize the new sample x', we input the encoding of the original sample x and the new batch label t' into the decoder:

$$D(\phi(x), t') = D(A(s), t').$$

Using the perfect reconstruction property, we get:

$$D(A(s), t') = f(s, t').$$

By definition, $x' = f(s, t')$. Therefore, we have shown:

$$D(\phi(x), t') = x'.$$

This demonstrates that the model can effectively "translate" a sample to a different batch by preserving the core biological signal captured by the encoder and applying the new batch information. □

1.2 Generalization error - Sber

Theorem 4 (Low Generalization Error on Future Source and Target Samples)**.** *Let $h : \mathcal{Z} \to \mathcal{Y}$ be a hypothesis function, and let ℓ be a loss function satisfying the inverse triangle equality (l_1 satisfy the inverse triangle equality):*

$$\|\ell(x,y) - \ell(x,z)\| \leq \ell(y,z), \quad \forall x, y, z.$$

Let $\xi_{\mathcal{P}^t}(h)$ denote the expected error of h under distribution $\mathcal{P}^t$. Then, for batches $t, t' \in \mathcal{T}$—where t represents a labeled source batch and t' a labeled target batch the following inequality holds:

$$\xi_{\mathcal{P}^{t'}}(h) \leq \xi_{\mathcal{P}^t}(h) + Md(Q_t, Q_{t'}) + \chi_{t,t'} + \chi_{t',t},$$

where:

- *$\xi_{\mathcal{P}^t}(h)$ and $\xi_{\mathcal{P}^{t'}}(h)$ represent the expected error of h on the source and target distributions, respectively/.*
- *$d(Q_t, Q_{t'})$ quantifies the distributional (covariate) shift between batches t and t', based on their marginal distributions.*

- *$\chi_{t,t'}$ and $\chi_{t',t}$ measure the cross-batch conditional alignment (CBCA) error between the two distributions.*

Proof.

$$
\begin{aligned}
\epsilon_{\mathbb{P}^T}(h) &= \epsilon_{\mathbb{P}^T}(h) + \epsilon_{\mathbb{P}^S}(h) - \epsilon_{\mathbb{P}^S}(h) \\
&\quad + \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y) && (6)\\
&\le \epsilon_{\mathbb{P}^S}(h) + \left|\mathbb{E}_{x\in\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y) - \epsilon_{\mathbb{P}^S}(h)\right| \\
&\quad + \left|\epsilon_{\mathbb{P}^T}(h) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y)\right| && (7)\\
&= \epsilon_{\mathbb{P}^S}(h) + \left|\mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^S}\ \ell(h(x),y)\right| \\
&\quad + \left|\mathbb{E}_{x\sim\mathbb{Q}^T}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y)\right| && (8)\\
&= \epsilon_{\mathbb{P}^S}(h) + \left|\mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y) - \ell(h(x),y')\right| && (9)\\
&\quad + \left|\mathbb{E}_{x\sim\mathbb{Q}^T}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}_Y^T}\ \ell(h(x),y)\right| \\
&\le \epsilon_{\mathbb{P}^S}(h) + \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y,y'\sim\mathbb{P}_Y^S\times\mathbb{P}_Y^T}\left[\ell(h(x),y) - \ell(h(x),y')\right] \\
&\quad + \left|\int_{x\in\mathcal{X}} Q^S(x)\ \mathbb{E}_{y\sim\mathbb{P}_Y^T}[\ell(h(x),y)] - Q^T(x)\ \mathbb{E}_{y\sim\mathbb{P}_Y^T}[\ell(h(x),y)]\ dx\right| && (10)\\
&= \epsilon_{\mathbb{P}^S}(h) + \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y,y'\sim\mathbb{P}_Y^S\times\mathbb{P}_Y^T}|\ell(h(x),y) - \ell(h(x),y')| \\
&\quad + \left|\int_{x\in\mathcal{X}} \left[Q^S(x) - Q^T(x)\right]\ \mathbb{E}_{y\sim\mathbb{P}_Y^T}\ell(h(x),y)\ dx\right| && (11)\\
&\le \epsilon_{\mathbb{P}^S}(h) + \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y,y'\sim\mathbb{P}_Y^S\times\mathbb{P}_Y^T}\ell(y,y') + M\int_{x\in\mathcal{X}}\left|Q^S(x) - Q^T(x)\right| dx && (12)\\
&= \epsilon_{\mathbb{P}^S}(h) + \chi^S + M\, d_{\mathrm{TV}}(Q^T, Q^S). && (13)
\end{aligned}
$$

□

The step from (11) to (12) uses the Lipschitz-type assumption:

$$|\ell(x,y) - \ell(x,z)| \le \ell(y,z),$$

and the boundedness of the expected loss function by M.

Repeating the same derivation but swapping S and T yields:

$$\epsilon_{\mathbb{P}^T}(h) \le \epsilon_{\mathbb{P}^S}(h) + \chi^T + M\, d_{\mathrm{TV}}(Q^T, Q^S).$$

Thus, we obtain the final bound:

$$\epsilon_{\mathbb{P}^T}(h) \le \epsilon_{\mathbb{P}^S}(h) + M\, d(Q^S, Q^T) + \min\{\chi^S, \chi^T\}.$$

—

1.3 Supervised Batch Effect Removal

Architecture. Following a standard domain adaptation strategy, we adopt a Siamese network architecture. The final layer acts as the hypothesis function $\tilde{h}$, while the remaining layers form the mapping function ϕ from input data to a d-dimensional embedded space.

During training, we feed pairs of mini-batches (X_{B_i}, Y_{B_i}) and (X_{B_j}, Y_{B_j}) from source and target batches. Let $\phi(X_v) \in \mathbb{R}^{n \times d}$ denote the mapped features of batch $v \in \{B_i, B_j\}$.

The empirical loss terms are:

$$\hat{\epsilon}_{P_{B_i}}(\tilde{h}) + \hat{\epsilon}_{P_{B_j}}(\tilde{h}) = L(\tilde{h}(\phi(X_{B_j})), Y_{B_j}) + L(\tilde{h}(\phi(X_{B_i})), Y_{B_i}),$$

where $L(X, Y) \triangleq \sum_{i=1}^{n} \ell(X_i, Y_i)$.

The empirical covariate shift term $\hat{d}(Q_{B_i}, Q_{B_j})$ is implemented as:

$$L_{\text{UDA}}(\phi(X_{B_i}), \phi(X_{B_j})),$$

where L_{UDA} corresponds to the chosen unsupervised domain adaptation method. In our experiments, we used a second-order statistic alignment method which is parameter-free. We also tested MMD and Sinkhorn-based optimal transport, obtaining similar results.

Approximating the CBCA term. The CBCA term in (4) is estimated via kernel regression (KR) to generate labels for a batch based on its cross-batch features. For $v \in \{B_i, B_j\}$ and its cross-batch $\bar{v}$, KR generates $\hat{y}$ such that $(x, \hat{y})$ follows $P_{\bar{v}}$ for $x \sim Q_v$. These synthetic pairs are then used to approximate χ_v.

Algorithm 1 Cross-batch sample generation via KR

Input: 1. Mapped features of mini-batches $\phi(X^{B_i})$, $\phi(X^{B_j})$. 2. Labels Y^{B_j} of the cross-batch.
Output: Synthetic labels $\hat{Y}^{B_i}$ following $\tilde{P}_x^{B_j}$ for $x \sim B_i$.

1. Compute pairwise squared distances:

$$D_{B_i,B_j}[i,j] = \|\phi(X_i^{B_i}) - \phi(X_j^{B_j})\|^2.$$

2. Apply t-kernel similarity:

$$K_{B_i,B_j}[i,j] = \frac{1}{1 + D_{B_i,B_j}[i,j]^2}.$$

3. Compute kernel-weighted average of labels:

$$\hat{Y}^{B_i} = \text{diag}(K_{B_i,B_j}\mathbf{1})^{-1} K_{B_i,B_j} \mathbf{Y}_{B_j}.$$

Finally, the CBCA loss is computed as:

$$L(\hat{Y}_{B_i}, Y_{B_i}) + L(\hat{Y}_{B_j}, Y_{B_j}).$$

2 Metrics

2.0.1 Adjusted Rand Index (ARI)

To evaluate clustering performance, we employ the Adjusted Rand Index (ARI), which measures the agreement between clustering results and known annotations, such as batch and cell type labels. ARI adjusts for random chance, providing a robust assessment of clustering quality.

The ARI is computed as:

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}},$$

where: - n_{ij} is the number of points in the intersection of cluster i and true label j, - a_i and b_j are the sums of points in cluster i and true label j, respectively, - n is the total number of points.

2.0.2 Normalized Mutual Information (NMI)

The Normalized Mutual Information (NMI) is a measure used to evaluate the agreement between two clusterings. It adjusts the Mutual Information (MI) by the entropy of the clustering and ground truth labels, ensuring the score is normalized. NMI is defined as:

$$\text{NMI}(U, V) = \frac{2 \cdot I(U, V)}{H(U) + H(V)}$$

where:

- $I(U, V)$ is the mutual information between clustering U and ground truth V,
- $H(U)$ and $H(V)$ are the entropies of U and V, respectively.

The score ranges between 0 and 1:

- 0: No shared information between the clusterings.
- 1: Perfect agreement between the clusterings.

2.0.3 Maximum Mean Discrepancy (MMD)

We utilize Maximum Mean Discrepancy (MMD) with a **Gaussian kernel** to assess distributional differences between two batches. The Gaussian kernel enables MMD to capture nonlinear relationships, making it suitable for comparing complex data distributions.

Given two distributions P and Q, and samples $\{x_i\}_{i=1}^n \sim P$ and $\{y_j\}_{j=1}^m \sim Q$, the squared MMD with a Gaussian kernel is estimated as:

$$\text{MMD}^2(P, Q) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j) + \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} k(y_i, y_j) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(x_i, y_j),$$

where the **Gaussian kernel** is defined as:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right),$$

and σ is the kernel bandwidth hyperparameter.

This formulation allows us to compute the discrepancy between two empirical distributions directly from their sample points. A smaller MMD value indicates higher similarity between the distributions, while a larger value suggests greater dissimilarity.

2.0.4 Adjusted Silhouette Width (ASW)

To evaluate the integration performance, we employ the Adjusted Silhouette Width (ASW) metric. This metric quantifies the separation of data points within different biological or batch groups after dimensionality reduction (e.g., PCA). Specifically, ASW is computed for two key annotations: batch (to assess mixing) and cell type (to assess biological relevance).

The ASW for a given data point i is defined as:

$$\text{ASW}_i = \frac{b_i - a_i}{\max(a_i, b_i)},$$

where: - a_i is the average dissimilarity of i to all other points within the same cluster, - b_i is the lowest average dissimilarity of i to points in any other cluster.

The overall ASW is the mean of ASW_i across all data points. Higher ASW values indicate better-defined clusters.