# 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 $S$. □

## Proof of corrolary 4

For the next claims, we assume that we are given a trained UBER model that is trained to perfect reconstruction and zero dependence, i.e., $R(\phi, d) = 0$ and $I(t, s') = 0$ for all $x \in \mathcal{X}$ and $t \in \mathcal{T}$. Let $\mathcal{X}_t = \{x \in \mathcal{X} : \text{there exists s} \in \mathcal{S} \text{ such that } x = f(s, t)\}$ be a subset of the set $\mathcal{X}$ corresponding to a fixed value $t$. Let $\phi_t$ be the restriction of the encoder $\phi$ to $\mathcal{X}_t$.

**Lemma 2.** *The map* $\phi_t : x \mapsto s'$ *is injective.*

*Proof.* Take $x_1 = f(s_1, t)$, $x_2 = f(s_2, t) \in X_t$ with $\phi_t(x_1) = \phi_t(x_2)$. Then

$$x_1 \;=\; d(\phi(x_1), t) \;=\; d(\phi(x_2), t) \;=\; x_2,$$

due to perfect reconstruction. □

Similarly, Let $d_t$ be the restriction of the decoder $d$ to $\mathrm{Im}(\phi(t))$.

**Lemma 3.** *The map* $d_t : Im(\phi_t) \to \mathcal{X}_t$*, is bijective.*

*Proof of Lemma 3.* This follows immediately from the assumption of perfect reconstruction, for any $x \in \mathcal{X}_t$, $d_t \circ \phi_t(x) = x$, thus $d_t \circ \phi_t$ is both surjective and injective. In particular, $d_t$ is injective. □

**Lemma 4.** *For a given encoder $\phi$, there exists an invertible map $A_t : S \mapsto S'$.*

*Proof.* Fix $t \in \mathcal{T}$ and let $f_t := f(\cdot, t) : S \to \mathcal{X}_t$. Define

$$A_t := \phi_t \circ f_t : \mathcal{S} \to \mathrm{Im}(\phi_t).$$

From Lemma 1, the restricted encoder $\phi_t$ is injective. Since $f_t$ is invertible too by assumption, $A_t$ is invertible. □

**Lemma 5.** *For any two batches $t_1, t_2 \in \mathcal{T}$, the maps*

$$A_t \;=\; \phi_t \circ f_t : \mathcal{S} \to \mathrm{Im}(\phi_t)$$

*satisfy*

$$\mathrm{Im}(A_{t_1}) \;=\; \mathrm{Im}(A_{t_2}).$$

*Proof.* Suppose for contradiction that $\mathrm{Im}(A_{t_1}) \neq \mathrm{Im}(A_{t_2})$. Then there exists $s' \in \mathrm{Im}(A_{t_1}) \setminus \mathrm{Im}(A_{t_2})$. Thus $\exists\, s \in \mathcal{S}$ such that $A_{t_1}(s) = s'$, but $\nexists\, s \in \mathcal{S}$ with $A_{t_2}(s) = s'$. Hence $s'$ is a feature value that occurs only under batch $t_1$, contradicting the assumption $S' \perp\!\!\!\perp T$. Therefore $\mathrm{Im}(A_{t_1}) = \mathrm{Im}(A_{t_2})$. □

**Theorem 6** (Uniqueness of per-batch map under asymmetry)**.** *Let $(\phi, d)$ be a trained UBER model satisfying perfect reconstruction and batch-independence:*

$$R(\phi, d) = 0, \qquad S' \perp\!\!\!\perp T.$$

*Assume further that the underlying distribution $p_S$ of the "biology" variable $S$ is asymmetric in the sense that for any measurable, invertible map $g : \mathcal{S} \to \mathcal{S}$,*

$$g_{\#}p_S = p_S \quad \Longrightarrow \quad g = \mathrm{id},$$

*where $g_{\#}p_S$ denotes the pushforward of the distribution $p_S$ through $g$* [1] [2].

*Under these conditions, all per-batch maps coincide: there exists a single invertible map $A : \mathcal{S} \to \mathcal{S}'$ such that*

$$A_t = A \quad \forall\, t \in \mathcal{T}.$$

*Proof by Contradiction.* Suppose, for the sake of contradiction, that there exist two batches $t_1, t_2 \in \mathcal{T}$ such that

$$A_{t_1} \neq A_{t_2}.$$

From lemma 5, we can define the map $g$

$$g := A_{t_2}^{-1} \circ A_{t_1} : \mathcal{S} \to \mathcal{S}.$$

By Lemma 4, $g$ is well-defined and invertible. The pushforward of $p_S$ through $g$ is thus

$$g_{\#}p_S = (A_{t_2}^{-1} \circ A_{t_1})_{\#}p_S.$$

[1] Intuitively, this means that $p_S$ admits no nontrivial invariances—any transformation that preserves it must be the identity.

[2] See lemma 9 for a proof that the set of distributions obeying the assymetry property is nonempty

Since $S' \perp\!\!\!\perp T$, we have

$$A_{t_1}(S) \stackrel{d}{=} A_{t_2}(S),$$

where $\stackrel{d}{=}$ denotes equality in distribution, as otherwise the encoding $S'$ carries information on the batch $T$. Hence

$$A_{t_2}^{-1} A_{t_1}(p_S) \stackrel{d}{=} A_{t_2}^{-1} A_{t_2}(p_S) \stackrel{d}{=} p_S.$$

By the asymmetry of $p_S$, this implies that

$$g = \mathrm{id}.$$

But $g = \mathrm{id}$ is equivalent to $A_{t_2}^{-1} \circ A_{t_1} = \mathrm{id}$, which gives

$$A_{t_1} = A_{t_2},$$

in contrudiction to our assumption that $A_{t_1} \neq A_{t_2}$. Therefore, all per-batch maps must coincide. □

**Lemma 7.** *Let $A$ be the transformation from Lemma 6. Then, for the trained decoder $d : Im(\phi) \times \mathcal{T} \to \mathcal{X}_t$ and generation function $f : S \times T \to X$, it holds that $d_t = f_t \circ A^{-1}$.*

*Proof.* The decomposition of the decoder follows from perfect reconstruction, as for all $(s', t) \in \mathrm{Im}(\phi) \times \mathcal{T}$:

$$d(s', t) = d_t(A(s)) = \hat{x} = x = f_t(s) = f_t(A^{-1}(s')).$$

□

**Corollary 8** (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 7, the model can synthesize $x'$ from the encoding of $x$ and the target batch label $t'$. Formally:*

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

*Proof.* Consider the input sample $x = f(s, t)$. Its encoding is

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

Passing this encoding with a new batch label $t'$ into the decoder gives

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

Applying Proposition 7 again yields

$$d(A(s), t') = f_t \circ A^{-1}(A(s)).$$

Since $A$ is invertible (Lemma 3), we have $A^{-1}(A(s)) = s$. Therefore,

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

By definition, $x' = f(s, t')$. Hence,

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

which establishes the claim. □

**Lemma 9.** *[Existence of asymmetric distributions] Let $\mathcal{S} \subset \mathbb{R}^d$ contain $n$ distinct points $x_1, \ldots, x_n$, and define the probability measure*

$$p_S = \sum_{i=1}^{n} p_i \, \delta_{x_i},$$

*where $\delta_{x_i}$ is the Dirac measure at $x_i$, and the masses $p_i$ are positive and pairwise distinct:*

$$p_i > 0, \quad p_i \neq p_j \text{ for } i \neq j, \quad \sum_{i=1}^{n} p_i = 1.$$

*(Choose $n \geq 2$. For linear maps, choose $x_1, \ldots, x_n$ such that at least $d+1$ points are affinely independent.)*

*Then any measurable bijection $g : \mathbb{R}^d \to \mathbb{R}^d$ that preserves $p_S$, i.e., $g_{\#}p_S = p_S$, must satisfy $g = \mathrm{id}$ on the support $\{x_i\}$. If $g$ is additionally continuous (or linear) and the support affinely spans $\mathbb{R}^d$, then $g = \mathrm{id}$ on all of $\mathbb{R}^d$.*

**Proof.**

1. *Pushforward on atoms forces permutation of support.* Since $p_S$ is supported exactly on $\{x_1, \ldots, x_n\}$, the equality $g_{\#}p_S = p_S$ implies that the image of the support is the support. Thus $g$ permutes the support:

   $$\forall i \; \exists j \text{ such that } g(x_i) = x_j.$$

   Hence $g$ induces a permutation $\sigma$ on $\{1, \ldots, n\}$ defined by $g(x_i) = x_{\sigma(i)}$.

2. *Mass equality forces identity permutation.* For any index $j$,

   $$p_j = p_S(\{x_j\}) = p_S(g^{-1}(\{x_j\})) = p_{\sigma^{-1}(j)}.$$

   Since the masses $p_i$ are pairwise distinct, this forces $\sigma(i) = i$ for all $i$. Hence

   $$g(x_i) = x_i \quad \forall i.$$

3. *Extension to the whole space (optional).* If $g$ is linear and $\{x_i\}$ contains $d+1$ affinely independent points, a linear map fixing these points must be the identity on $\mathbb{R}^d$. Therefore, $g = I$.

   If $g$ is continuous and the support has nonempty interior, continuity plus density arguments extend the identity to the whole space.

This proves that in the atomic example, there is no nontrivial measurable bijection $g$ preserving the law: every such $g$ fixes the support pointwise.

## 1.2 Generalization error - Sber

**Theorem 10** (Low Generalization Error on Future Source and Target Samples)**.** *Let $h : \mathcal{Z} \to \mathcal{Y}$ be a hypothesis function, and let $\ell$ be a loss function satisfying the inverse triangle equality ($\ell_1$ 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}^T_Y}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}^T_Y}\ \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}^T_Y}\ \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}^T_Y}\ \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}^T_Y}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}^S_Y}\ \ell(h(x),y)\right| \\
&\quad + \left|\mathbb{E}_{x\sim\mathbb{Q}^T}\mathbb{E}_{y\sim\mathbb{P}^T_Y}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}^T_Y}\ \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}^T_Y}\ \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}^T_Y}\ \ell(h(x),y) - \mathbb{E}_{x\sim\mathbb{Q}^S}\mathbb{E}_{y\sim\mathbb{P}^T_Y}\ \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}^S_Y\times\mathbb{P}^T_Y}\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}^T_Y}[\ell(h(x),y)] - Q^T(x)\ \mathbb{E}_{y\sim\mathbb{P}^T_Y}[\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}^S_Y\times\mathbb{P}^T_Y}|\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}^T_Y}\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}^S_Y\times\mathbb{P}^T_Y}\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 $\phi$ 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 $L_{\text{UDA}}$ 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 $\chi_v$.

---

**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 $\sigma$ 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 $\text{ASW}_i$ across all data points. Higher ASW values indicate better-defined clusters.