MODEL ALIGNMENT USING INTER-MODAL BRIDGES

Ali Gholamzadeh,
MPI for Biological Cybernetics
& University of Tübingen
ali.gholamzadeh@tue.mpg.de

Noor Sajid
Kempner Institute, Harvard University
& MPI for Biological Cybernetics
noorsajid@g.harvard.edu

ABSTRACT

Foundation models have demonstrated remarkable performance across modalities such as language and vision. However, inter-modal model reuse remains limited due to the difficulty of aligning internal representations. Existing methods require extensive paired training data or are constrained to specific domains. We introduce a semi-supervised approach for model alignment via conditional flow matching. The conditional flow between latent spaces of different modalities (e.g., text-to-image or biological-to-artificial neuronal activity) can be learned in two settings: (1) solving a (balanced or unbalanced) optimal transport problem with an inter-space bridge cost, and (2) performing memory-efficient alignment using labelled exemplars. Despite being constrained by the original models' capacity, our method–under both settings–matches downstream task performance of end-to-end trained models on object recognition and image generation tasks across MNIST, ImageNet, and Majaj et al. (2015) datasets, particularly when labelled training data is scarce ($< 20\%$). Our method provides a data-efficient solution for inter-modal model alignment with minimal supervision.

1 INTRODUCTION

Foundation models like GPT-X, DeepSeek, Gemini, Sora, and Dall-E have demonstrated remarkable performance across modalities such as text, image, and video (Brown, 2020; OpenAI, 2023; Gemini et al., 2023). While these large-scale models represent significant investments in computational resources and data curation (Brown, 2020), their inter-modal model reuse[1] remains limited by the fundamental challenge of aligning internal representations (Imani et al., 2021; Klebe et al., 2023; Huh et al., 2024). Existing approaches for aligning models across modalities typically require extensive paired datasets to build correspondence (Zhai et al., 2022), yet such datasets are rarely available at scale (Gadre et al., 2024). Current alignment techniques are further constrained by their reliance on abundant paired data or their focus on specific domains (Gadre et al., 2024), limiting their broader applicability. Thus, developing alignment methods that can operate with minimal supervision while generalising across domains remains an open and critical challenge.

To address this, we propose *model space alignment via inter-modal bridges* for inter-modal integration of models with minimal supervision. Our approach centres on learning morph between latent spaces, where a noise distribution is mapped to the target space conditioned on source samples (Klein et al., 2023). These morphs are learned in a semi-supervised setting with access to a small set of paired samples between target and source distributions. We use these paired samples in two ways: through true alignment using the labelled pairs themselves, or by solving a balanced or unbalanced optimal transport (OT) problem (Peyré & Cuturi, 2019) that uses our inter-modal bridge cost(Sec. 3.3) to compute optimal couplings between distributions. The inter-modal bridge cost captures similarities between latent spaces using intra-space distances and paired samples.

Our contributions are as follows:

- Introduce an inter-modal bridge cost across distinct latent spaces using intra-space distances and paired inter-space samples (Sec. 3.3).
- Show improved conditional flow matching using global OT alignment and true alignment compared to a local OT alignment baseline (Klein et al., 2023) (Sec. 5).

[1]Reusing pre-trained models across different data modalities, such as using a vision model on text data, without needing to retrain them.

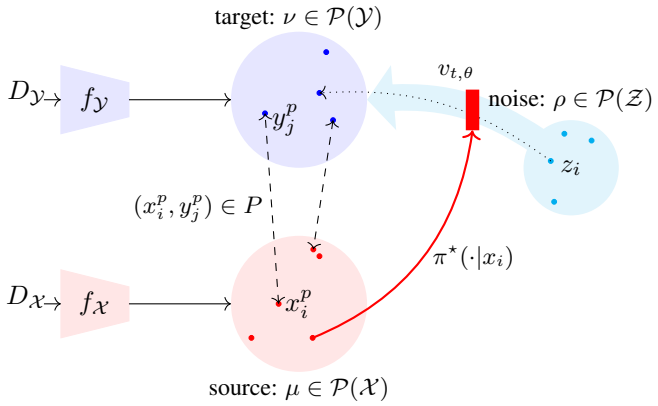


Figure 1: Pictorial representation of our approach for aligning model space using multi-modal bridges. We consider two pre-trained models; $f_{\mathcal{X}}$ and $f_{\mathcal{Y}}$ and their corresponding datasets $D_{\mathcal{X}}$ and $D_{\mathcal{Y}}$. Next, we obtain source and target latent distributions μ and ν and compute the optimal coupling $\pi^\star$ across these two distributions using paired samples $(x_i^p, y_j^p) \in P$ as inter-space bridges or using the paired sampled directly. Given this, we learn the velocity field $v_{t,\theta}$ that morphs noise ρ conditioned on x_i in the source distribution to some target y_i using samples from the optimal coupling $\pi^\star(\cdot \mid x_i)$.

- Validation of our model alignment method – between image-text and biological-artificial neural representations – on downstream object recognition and image generation tasks using minimal paired samples (i.e., $< 20\%$) (Sec. 5) across different datasets (Sec. 4).

2 Preliminaries

In this work, we consider two datasets, $D_{\mathcal{X}}$ and $D_{\mathcal{Y}}$, and two bounded sets, $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset \mathbb{R}^q$, referred to as the source and target domains respectively (Fig. 1). The model representation is defined as $f_{\mathcal{X}} : D_{\mathcal{X}} \to \mathcal{X}$ and the set of probability measures on $\mathcal{X}$ is denoted as $\mathcal{P}(\mathcal{X})$. For a coupling $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, the marginal distribution is denoted as $\pi_{\mathcal{X}}(x) = \int_{\mathcal{Y}} \pi(x, y)\, dy$. The entropy for $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ is given by $H(\pi) = -\int_{\mathcal{X}\times\mathcal{Y}} \pi(x, y) \log(\pi(x, y))\, d(x, y)$.

(Unbalanced) Linear entropic OT Given some cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the (unbalanced) linear entropic OT for $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$ is:

$$\pi^\star := \arg \inf_{\pi \in \mathcal{P}(\mathcal{X}\times\mathcal{Y})} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y)\pi(x, y)\mathrm{d}x\mathrm{d}y - \epsilon H(\pi) + \lambda_{\mathcal{X}} \mathrm{KL}(\pi_{\mathcal{X}} \parallel \mu) + \lambda_{\mathcal{Y}} \mathrm{KL}(\pi_{\mathcal{Y}} \parallel \nu)\,, \tag{1}$$

where $\epsilon \geq 0$ is a hyperparameter controlling the trade-off between minimising the transport cost and the smoothness of the solution, λ_is the unbalanced weighting parameters[2] and $\mathrm{KL}(\cdot \parallel \cdot)$ denotes the Kullback–Leibler divergence. For the discrete setting, the Sinkhorn algorithm (Cuturi, 2013) solves the linear entropic OT problem by iteratively updating the coupling to minimise the regularised cost while satisfying marginal constraints. In the unbalanced setting a variation of the Sinkhorn algorithm can be used (Frogner et al., 2015; Séjourné et al., 2023).

(Unbalanced) Quadratic entropic OT Given two intra-space cost functions $c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, this method extends the linear OT problem to distinct spaces by learning a coupling that encourages the matches of elements close in one probability distribution to be close in the other distribution as well (Vayer, 2020; Séjourné et al., 2023):

$$\begin{aligned} \pi^\star := \arg \inf_{\pi \in \mathcal{P}(\mathcal{X}\times\mathcal{Y})} & \int_{(\mathcal{X}\times\mathcal{Y})^2} \left|c_{\mathcal{X}}(x, x') - c_{\mathcal{Y}}(y, y')\right|^2 \, d\pi(x, y)\, d\pi(x', y') - \epsilon H(\pi) \\ & + \lambda_{\mathcal{X}} \mathrm{KL}^{\otimes}(\pi_{\mathcal{X}} \parallel \mu) + \lambda_{\mathcal{Y}} \mathrm{KL}^{\otimes}(\pi_{\mathcal{Y}} \parallel \nu)\,, \end{aligned} \tag{2}$$

where tensorised $\mathrm{KL}^{\otimes}(p \parallel q) = \mathrm{KL}(p \otimes p \parallel q \otimes q)$.

[2] In our experiments, instead of directly setting the λ_i's, we use an alternative parameter τ (see Appendix L.1)

(Unbalanced) Fused Gromov-Wasserstein (U-FGW) Given two partially comparable spaces, U-FGW extends U-GW by combining both the intra-space structural dissimilarity with an inter-space feature discrepancy (Titouan et al., 2019) and can be formalised as:

$$\pi^\star := \arg\inf_{\pi\in\mathcal{P}(\mathcal{X}\times\mathcal{Y})} \int_{(\mathcal{X}\times\mathcal{Y})^2} \left(\alpha|c_\mathcal{X} - c_\mathcal{Y}|^2 + (1-\alpha)c^2_{\mathcal{XY}}\right) d\pi(x,y)\, d\pi(x',y') - \epsilon H(\pi) \\ + \lambda_\mathcal{X}\mathrm{KL}^\otimes(\pi_\mathcal{X} \parallel \mu) + \lambda_\mathcal{Y}\mathrm{KL}^\otimes(\pi_\mathcal{Y} \parallel \nu)\,, \tag{3}$$

where $c_\mathcal{X} : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ and $c_\mathcal{Y} : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$ are the intra-cost functions: $c_{\mathcal{XY}} : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ the fused cost[3] and $\alpha \in [0,1]$ the trade-off parameter. For discrete settings, this can be solved by iteratively updating the coupling using conditional gradient updates (Vayer, 2020; Séjourné et al., 2021).

(Unbalanced) Generative entropic neural OT (U-GENOT) Like the OT solvers, U-GENOT aims to find an optimal mapping between the source and target distributions (Klein et al., 2023) using conditional flow matching (CFM). CFM was introduced as a simulation-free technique to train a velocity field by regressing it against a target vector field (Lipman et al., 2022). Here, the velocity field, $v_{t,\theta}(x)$, is a time-varying vector-valued function that describes the instantaneous direction and speed of movement for each point in the space from the initial distribution to the target distribution. Tong et al. (2023) used CFM to distil the discrete optimal map between two distributions in the same space (OT-CFM). See Appendix B for further details.

U-GENOT extends this to cases where the source and target distributions are in different spaces, i.e., $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$ (Klein et al., 2023). For this, a conditional time-varying velocity field $v_{t,\theta}(\cdot \mid x) : \mathcal{X} \to \mathcal{Y}$ on the target space was assumed and optimal parameters (θ^*) found by minimising:

$$\mathcal{L}_{\text{U-GENOT}}(\theta) = \mathbb{E}_{t,z\sim\rho(x,y),\, \sim\pi^\star} \|v_{t,\theta}(ty + (1-t)z \mid x) - (y-z)\|^2 \\ + \mathbb{E}_{x\sim\mu}\left[(\eta - \eta_\theta)(x)^2\right] + \mathbb{E}_{y\sim\nu}\left[(\xi - \xi_\theta)(y)^2\right], \tag{4}$$

where $t \sim U[0,1]$ and $\pi^\star$ is the optimal coupling found using any discrete solvers (e.g. U-GW and U-FGW) on mini-batches i.e., local alignment and η_θ, ξ_θ are the non-negative neural reweighting functions (for further details see Appendix L.1). Importantly, U-GENOT learns a separate flow for each point in the source distribution $x \in \mathcal{X}$ and transforms a noise distribution $\rho \sim \mathcal{N}(0, I_q) \in \mathcal{P}(\mathbb{R}^q)$ into a conditional coupling $\pi^\star(\cdot \mid x)$ (defined using an appropriate OT solver) for out-of-sample prediction.

3 Learning inter-modal morphs using bridge cost

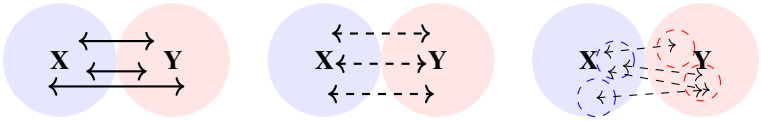
Building upon the neural OT formulation introduced in Sec. 2, we outline our approach for learning inter-modal morphs[4]. Specifically, we employ U-GENOT (Klein et al., 2023) to learn a transport function that maps between the latent distributions of two distinct domains (i.e., vision and text), leveraging features extracted from particular models and paired data points (Sec 3.1). This involves selecting the appropriate alignment strategy (Sec 3.2) to either find the optimal coupling ($\pi^\star$: Eq. 1-3) using the new bridge cost function (Sec 3.3) or using the labelled pairs directly, and an augmented neural architecture for learning the velocity field (Eq. 4; Appendix E.1).

3.1 Problem setting

Given datasets $D_\mathcal{X}$ and $D_\mathcal{Y}$ where each element represents the same object but originates from distinct spaces, we aim to learn a transport function $T : \mathcal{X} \to \mathcal{Y}$ mapping the latent distribution $\mu \in \mathcal{P}(\mathcal{X})$ to $\nu \in \mathcal{P}(\mathcal{Y})$ (Fig. 1). For this, we leverage models as feature extractors for each space: $f_\mathcal{X} : D_\mathcal{X} \to \mathcal{X}$ and $f_\mathcal{Y} : D_\mathcal{Y} \to \mathcal{Y}$. These models map data points from their respective domains to feature vectors, yielding latent distributions in the feature spaces, denoted as $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$, respectively. Then, our task is to learn the transport function T. To facilitate this, we assume access to a set of paired points $(x_i^p, y_i^p) \in P$, where $x_i^p = f_\mathcal{X}(d_i^p) \in \mathcal{X}$ and $y_i^p = f_\mathcal{Y}(d_i^p) \in \mathcal{Y}$ represent the latent representations of the same object across the two domains.

[3] We use "fused cost" to refer to inter-space distance functions $\mathcal{X}\times\mathcal{Y} \to \mathbb{R}$.

[4] Here, we use "morph" as a general term for mapping a source distribution to a target distribution, while "conditional flow matching" refers to the specific implementation used to learn this mapping.



A. True B. Global C. Local

Figure 2: Pictorial representation of alignment methods. Here, — represents true pairs, $x_{ip}, y_{ip} \in P$, $--$ represents coupling given some fused cost C_{XY} and OT solver, and ○ & ○ represent batches of samples.

The aim is for each pair of data points, $d_i^x \in D_{\mathcal{X}}$ and $d_i^y \in D_{\mathcal{Y}}$ to satisfy: $f_{\mathcal{Y}}(d_i^y) \approx T(f_{\mathcal{X}}(d_i^x))$. In other words, the transport function T should map the latent representation $x_i = f_{\mathcal{X}}(d_i^x)$ in space $\mathcal{X}$ to approximately match the corresponding latent representation $y_i = f_{\mathcal{Y}}(d_i^y)$ in domain $\mathcal{Y}$.

Our approach supports both supervised and semi-supervised learning paradigms. In this context, we assume access to n source domain points, denoted as $X = [x_i]_{i=1}^n$, and m target domain points, denoted as $Y = [y_j]_{j=1}^m$. Among these, we assume a set of paired points denoted by P, with size $|P| = l$, where $0 < l \leq \max(n, m)$. This serves as anchor for learning the inter-domain mapping. Accordingly, the supervised setting would be where $l = n = m$, i.e., all points are paired across domains. The semi-supervised setting occurs when $l < \min(n, m)$, allowing us to leverage both paired and unpaired data points to construct inter-space cost (see Sec. 3.3). This enables us to tackle a wide range of practical scenarios.

3.2 Alignment Strategy

To compute the optimal coupling ($\pi^\star$) used to train the velocity field $v_{t,\theta}$, we propose three distinct alignment strategies: true, global, and local (Fig. 2). True alignment focuses on paired samples to construct a joint distribution between the source and target domains. Given a paired set P of size l, the coupling $\pi^{\text{true}} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ is:

$$\pi^{\text{true}}(x_i, y_j) = \begin{cases} \frac{1}{l}, & \text{if } (x_i, y_j) \in P \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This approach ensures direct alignment for paired data but does not fully use the information available in unpaired samples.

Conversely, global alignment computes a full OT plan using the entire dataset. First, we construct a fused-cost matrix C_{XY} – as defined in Sec 3.3 and Appendix F – and then solve the OT problem: $\pi^{\text{global}} = OT_{\epsilon,\tau}(X, Y, C_{XY})$, where ϵ is the entropy regularisation parameter and τ is unbalanced parameter. This can be implemented using either linear OT with the Sinkhorn solver or quadratic OT with the FGW solver[5]. Global alignment provides an appropriate mapping but incurs a significant memory cost of $O(nm)$, as it requires computing and storing the distance between each pair. To address scalability concerns, we compare with local alignment following (Klein et al., 2023). This solves the OT problem on subsets of the data at each iteration. For batches X^b and Y^b drawn from the source and target distributions, we compute: $\pi_b^{\text{local}} = OT_{\epsilon,\tau}(X^b, Y^b, C_{XY}^b)$. While this strategy improves scalability, it can introduce misalignment due to batch approximations (Fatras et al., 2021) and the time complexity ranges from $O(b^2)$ to $O(b^3)$ depending on the solver for batch size b (Cuturi, 2013; Scetbon et al., 2022). This is because we are calculating a 'local' π_b^{local} using a 'local' C_{XY}^b at each iteration, and then sample from this to learn the conditional flow.

3.3 Inter-modal Bridge cost

We propose a bridge cost function that leverages paired samples to define the inter-domain cost (Fig.3). For this, we use paired points as bridges between two spaces to calculate the inter-space cost function. Using the intra-space cost matrices C_{XX} and C_{YY}, we define C_{XY}^{bridge} as:

$$C_{XY}^{\text{bridge}}(x_i, y_j) = \begin{cases} 0 & \text{if } (x_i, y_j) \in P \\ \min\limits_{(x_i^p, x_j^p) \in P} \left(C_{XX}(x_i, x_i^p) + C_{YY}(y_j^p, y_j) \right) & \text{otherwise} \end{cases} \tag{6}$$

[5]To calculate the inter-space cost C_{XX} and C_{YY}, we use cosine distance unless specified otherwise.

The C_{XY}^{bridge} is determined as the minimum sum of intra-costs, using the paired samples as zero-cost links between the spaces. This represents the optimal bridging cost between the paired elements. The intra-cost, C_{XX} and C_{YY} was calculated as the cosine distance between $x_i, x_j \in X$ and $y_i, y_j \in Y$.

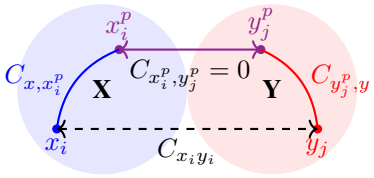


Figure 3: Pictorial representation of bridge cost via $(x_i^p, y_j^p) \in P$ and $(x_i, y_j) \notin P$.

3.4 Algorithm Overview

To learn the multi-modal morphs, we extract the features using some models. Next, the optimal coupling $\pi^\star$ is calculated using one of the three proposed alignment strategies: true, global, or local. For global alignment, this involves calculating the bridge cost using paired samples, which informs the construction of the cost matrix, C_{XY} that is used for training the velocity field $v_{t,\theta}$ via (Eq. 4). We follow a similar approach for local alignment except the fused cost and optimal coupling are recalculated at each iteration. For true alignment, we use the paired samples to define the optimal coupling. Next, at each iteration, we sample from the coupling, generate noise, and optimise the velocity field to minimise the transport cost. Once trained, the velocity field can be used for out-of-sample prediction from the source to the target space. This iterative process continues until convergence or for a predefined number of iterations. See Appendix D for pseudo-code.

4 Dataset and Experimental Setup

We evaluated our approach in two settings: alignment between image-text representations and between biological-artificial neural network representations. For image-text alignment, we used:

- **MNIST** contains $60,000$ training and $10,000$ test samples of handwritten digits across ten classes (LeCun et al., 2010). For these experiments, we used $50,000$ samples from the training set to train two variational auto-encoders (VAE) (Kingma & Welling, 2014) (See Appendix G.1.1 for more details): *VAE$_{image}$* for reconstructing images and *VAE$_{text}$* for reconstructing the one-hot encoded labels. Afterwards, these trained networks were used as 'pre-trained' models and features were extracted for the $10,000$ remaining training samples. We used these to train the model morph from the latent space of *VAE$_{text}$* to the one of *VAE$_{image}$* and vice-versa[6]. The test data were used to evaluate different morph formulations.
- **ImageNet** contains approximately 1.2 million training samples across $1,000$ classes (Russakovsky et al., 2015). In these experiments, we used two pre-trained models: *ViT-Base* (Dosovitskiy et al., 2021), a vision transformer, for image feature extraction, and *MiniLM-L6* (Wang et al., 2020), a pre-trained sentence encoder, for textual features. Image features were derived from the classification token of the final layer of ViT-Base, while the textual features were encoded in the format 'A photo of a *class name*', following the protocol introduced in CLIP (Radford et al., 2021). We used 50% train$/10\%$ validation split to train inter-modal morphs, and the remaining 40% for evaluation.

For (potentially more noisy) alignment of biological-artificial neural representations, we used:

- **Majaj et al. (2015)** dataset that contains neural activity recordings from the visual area (V4) and the inferior temporal cortex (IT) of monkeys viewing distinct visual stimuli. The stimuli consisted of eight categories, each containing 8 core images, resulting in 64 unique stimuli. Each stimulus was paired with 50 randomly selected backgrounds, generating a final set of $3,200$ images. To reduce noise, neural activity for each unique stimulus was averaged across approximately 50 presentations, with a minimum of 28 repetitions per stimulus. We used randomly selected splits for train/validation/test datasets, $60\%/20\%/20\%$, that were consistently used for all analyses. Using this dataset, we considered how aligned the neural activity across a biological and artificial network could be when exposed to the same stimulus. To extract artificial neural representations, we used a pre-trained EfficientNet-B0 as its variants have shown effectiveness in Brain-Score metrics (Schrimpf et al., 2018).

[6]Bi-directionality was modelled to evaluate how morphing between text-to-image vs image-to-text could differ and their influence on the downstream task performance.

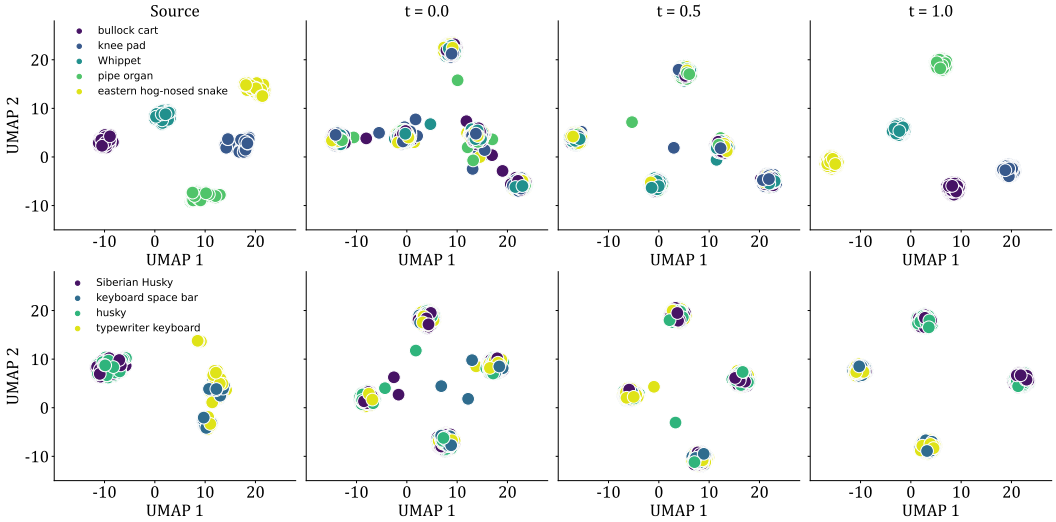


Figure 4: Noise distribution trajectory to the target latent space of a language model at $t = (0, 0.5, 1)$ using *true alignment* with 1% paired points. The latent source and target feature spaces for ImageNet are visualised using UMAP (McInnes et al., 2018). **Top:** Classes with minimal overlap in the image latent space. **Bottom:** Classes with high overlap in the image latent space.

Using these datasets, we assessed the impact of alignment modalities (Sec 5.1-5.2), model quality (Sec 5.1.2), the inter-modal bridge cost (Sec 5.1.3) (Sec 5.1.4) and velocity field architecture (Sec 5.1.4) for learning appropriate morphs. All reported experiments used 5 different random seeds and had a training budget of 18 hours on one $A100$ GPU. The models used for each experiment are presented in Appendix. G, with evaluation metrics incl. morph quality, and downstream task performance measures (Appendix. H).

5 Results

5.1 Image-text alignment

To align image-text representations, we used GENOT to learn conditional flow matching. In experiments with local and global alignment, the optimal coupling was computed using linear and FGW OT solvers.

5.1.1 Image-to-text conditional flow matching

We considered whether the degree of overlap in source feature space (i.e., images; Fig.4 source column) could influence the conditional flow matching to the target latent space (i.e., text; Fig. 4, $t = 1.0$ column) using UMAP projects of the feature spaces with 1% paired points[7]. Using true alignment, we observed that when the source feature space (i.e., images) had minimal overlap, the resulting distribution showed a clear separation (Fig. 4; top row, $t = 1.0$). Conversely, when the feature spaces are similar (e.g., Husky and Siberian Husky), the resulting distribution mirrors the source, exhibiting substantial overlap (Fig. 4; bottom row, $t = 1.0$).

5.1.2 Model quality

Building on this, we quantified how the quality of the model influenced the performance of the learned flow from image to text. We measured quality in terms of feature space overlap, reflecting how well-disentangled the encoded space was, using varying numbers of randomly selected classes from each dataset (Appendix H.1). For both datasets, we observed a decline in performance (refer to Appendix H.3 for how accuracy was computed) as the feature space overlap increased (Fig.5.A-B). The degree of overlap was directly correlated with the number of classes used (Fig. 5.C). Fig. 5.D

[7]We use UMAP to provide an intuitive low-dimensional representation, that captures simple correlations and local structure. Therefore, UMAP visualisations should be interpreted as qualitative approximations rather than definitive measures of alignment quality.

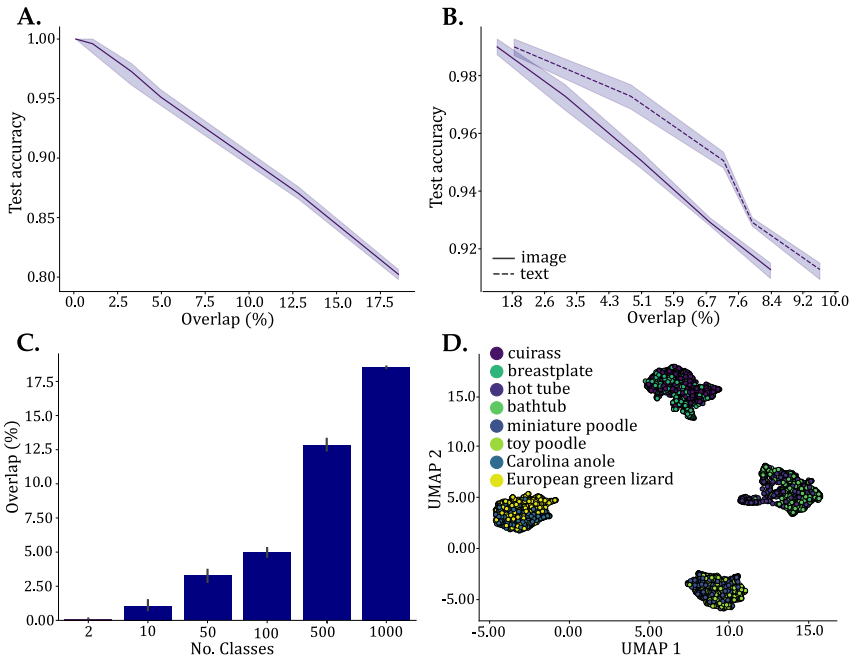


Figure 5: Latent space overlaps on conditional flow matching from image-to-text domain, using true alignment in a fully supervised setting. **A)** MNIST experiment. **B)** ImageNet experiment. **C)** Relationship between the number of classes and the degree of overlap in the latent space for the ImageNet dataset. **D)** UMAP visualisations of the classes with the highest overlap in the latent space of the ViT-B model.

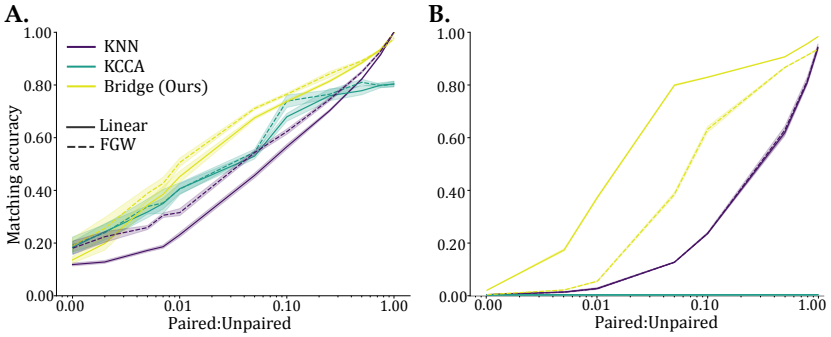


Figure 6: Matching accuracy across fused costs for linear and FGW solvers. The optimised value of $\alpha^\star$ was used for FGW solvers. Matching accuracy was calculated by sampling from the optimal coupling $\pi^\star$ and averaging the number of correct matches. **A)** MNIST experiment. $\alpha^\star$: KNN: 0.25, KCCA: 0.5, bridge: 0.5. **B)** ImageNet experiment. $\alpha^\star$: KNN: 0.25, KCCA: 0.5, bridge: 0.25.

presents a UMAP projection of the ViT-B feature space, highlighting several classes with significant overlap in the ImageNet experiments. This high overlap may result from the training capacity of the base ViT model (Tsipras et al., 2020) or potential mislabelling within the original dataset (Beyer et al., 2020). These results suggest that the latent spaces' degree of disentanglement and overall quality play a critical role in shaping the learned flow.

5.1.3 Inter-space bridge cost

We examined the effectiveness of the bridge cost function in learning the optimal coupling by evaluating the matching accuracy (i.e., the average number of correct responses based on the ground truth labels) for the Linear and FGW OT solvers. We compared the performance against KNN and KCCA cost functions (Appendix F), using large sample sizes ($100,000$ for ImageNet and $10,000$ for MNIST). Our results show that the bridge cost consistently outperformed the other cost func-

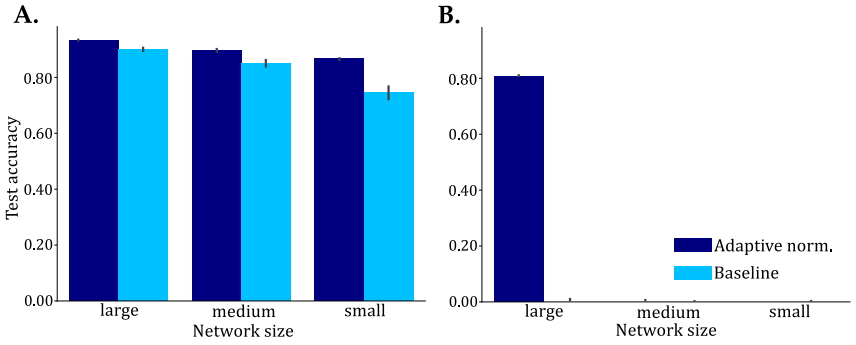


Figure 7: Velocity Field v_θ architecture benchmark. Comparison of the baseline feed-forward architecture and the adaptive normalisation architecture (with blocks) for image-to-text tasks in a fully supervised setting, using true alignment, across various sizes. Here, **A**) MNIST experiments, and **B**) ImageNet experiments.

tions across both datasets and various paired point configurations (Fig. 6). Additionally, we observe a clear association between the number of paired points and improvements in coupling quality and matching accuracy. Next, we evaluated the performance of discrete solvers. We find that FGW (Eq.3), which considers both intra- and inter-space costs using optimised $\alpha^\star$(Appendix. K), performed the best for MNIST and linear OT solver (Eq.1) for ImageNet. Based on these results, we employed the bridge cost and the best-performing OT solvers for all subsequent experiments.

5.1.4 Velocity field network architecture

We evaluated two distinct architectures – Klein et al. (2023) baseline and neural network with adaptive normalisation – for parameterising the conditional velocity field network $v_{t,\theta}$ (Appendix E.1). The baseline follows the design of GENOT (Klein et al., 2023), utilising a neural network that takes time, source, and latent noise as inputs. Each input vector is embedded independently using a multi-layer perceptron (MLP) block before concatenation. The second architecture, inspired by Diffusion Transformers (DiT) (Peebles & Xie, 2023), integrates adaptive layer norm (Perez et al., 2018) (adaLN) blocks (see Appendix E.1). Here, input latent noise is normalised in each block, conditioned on time and source data. We tested three size variants of each architecture. Our results indicate that the adaLN-based architecture consistently outperformed its counterpart (Fig. 7). For the MNIST dataset (Fig.7.A), all adaLN architectures outperformed the baseline architectures with similar parameter counts. For the ImageNet dataset, only the larger adaLN architecture successfully learned the mapping (Fig.7.B). Furthermore, in terms of sample efficiency, the larger adaLN networks achieved the target accuracy significantly faster – potentially due to dynamic activation normalisation (Appendix E). Given this, the velocity field in all remaining experiments was parameterised using the adaLN-Large architecture.

5.1.5 Downstream task performance

For the ImageNet dataset, we evaluated the accuracy of image-to-text feature space using varying numbers of paired samples (Fig. 8). We compared our approach against the classification head on ViT reported as 83.97 in Dosovitskiy et al. (2021). This represents the upper bound for the model's performance potential. Under the given training time constraints (i.e., 18 hours), local alignment failed to converge and exhibited poor performance, since it requires computing the fused-cost and optimal coupling at each iteration. Similarly, the global solver under-performed in settings with very few paired samples, likely due to misalignment issues stemming from the discrete solver in these regimes (Fig. 5). However, as the number of paired samples increased to $\approx 10\%$, performance improved to a level comparable to the classifier.

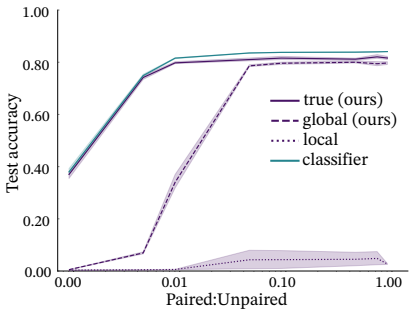


Figure 8: Image-to-text test accuracy for ImageNet.

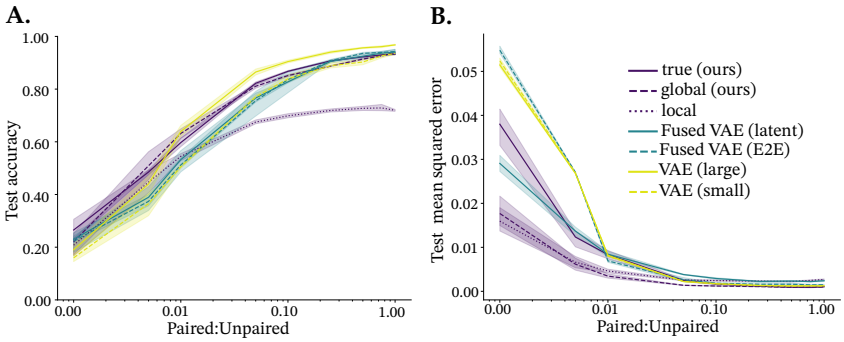


Figure 9: Downstream task performance across different alignment strategies using bridge cost with FGW solver, and the baseline models for MNIST. **A)** Test accuracy for text reconstruction from image input. **B)** Mean squared error between text-image constructions and true images pixel-wise 2D distributions for each class using test data.

Finally, we compared the morph results across different numbers of paired points for the MNIST dataset. The image-to-text transformation revealed that the performance of both global and true alignment was on par with the baseline methods(Fig. 9.A). Importantly, with a small number of paired samples (i.e., $< 10\%$), our method outperforms these baselines. However, local alignment had significantly worse performance compared to other methods. This is due to misalignment and flow misguidance–also noted in (Fatras et al., 2019), this effect can be mitigated by increasing the batch size (Klein et al., 2023). For the text-to-image transformation (Fig. 9.B), we generated images for each class from the corresponding labels. After morphing from VAE_{text} to the VAE_{image} latent space, we reconstructed the images and evaluated them using mean squared error (Appendix H.2). In scenarios with limited paired samples, both the local and global alignment methods outperformed all other approaches.

5.2 Biological-artificial neural representation alignment

Based on our initial evaluation Appendix. L.2 we observed that Unbalanced setup works better, U-GENOT to learn the conditional flow matching, U-EOT solver[8] to compute the optimal coupling for global alignment and used Pearson correlation to calculate the intra-space cost (Appendix. H.4).

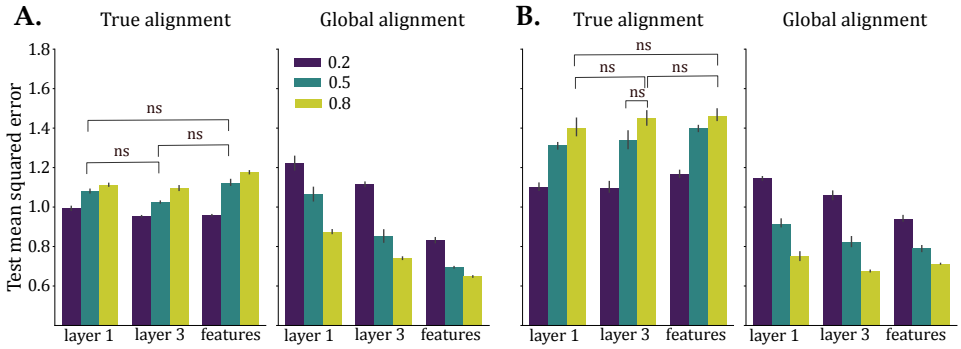


Figure 10: Neural prediction errors for the learned conditional flow matching from EfficientNet representations at *layer 1*, *layer 3*, and the *features* layer to neural recordings in **A)** IT and **B)** V4 using true and global alignment. Statistically significant differences were found for all pairwise comparisons, **except** those explicitly annotated with ns (Wilcoxon-Mann-Whitney test).

5.2.1 Model-to-neural activity conditional flow matching

To assess the quality of the learnt conditional flow matching, we compared the prediction error under different alignment strategies – true and global – while varying the proportion of paired data from 0.2 to 0.8. For true alignment, increasing the proportion of paired data led to over-fitting, as

[8]We used linear since it performed better than U-FGW.

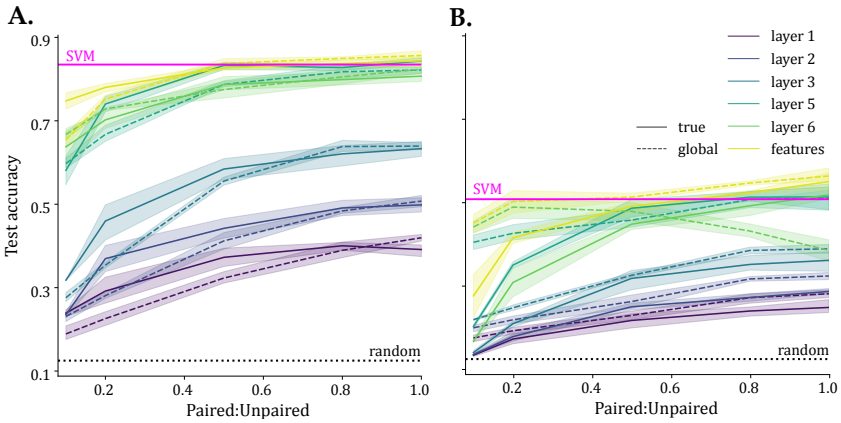


Figure 11: Downstream task performance for classifying core image categories using the learnt conditional flow maps from different layers of EfficientNet mapped to neural recordings from **A)** IT and **B)** V4 regions. The neural activity was decoded using an SVM from (Majaj et al., 2015); the model was optimised using the training data to predict the category from neural representations. The black dotted line indicates random assignment accuracy and the pink line is SVM's accuracy on the actual brain activity test set.

reflected in a consistent increase in prediction error when morphing between model layers and neural activity for V4 and IT (Fig. 10). For the global alignment given the highly noisy neural data, we adopt an unbalanced OT setting (Eq. 1-3) and following Klein et al. (2023) optimise the degree of mass conservation between source and target distributions (via τ_α and τ_β) (Eq. 4; Appendix L). Our results show that increasing the proportion of paired data corresponds to a decrease in prediction error when morphing between model layers and neural activity for V4 and IT (Fig. 10).

5.2.2 Downstream task performance

Next, we evaluated category classification accuracy across different images using varying numbers of paired samples (Fig. 11). For this, we did not apply neural re-weighting (Eq. 4). We compared our approach to an SVM trained to predict category labels, which serves as an upper bound for the models performance. For IT and V4, deeper layers of EfficientNet performed better on the downstream task for both true and global alignment. This aligns with prior findings that later layers encode higher-order semantic features, making them more relevant for predicting activity in higher-order brain areas, while earlier layers primarily capture low-level visual features (Yamins et al., 2014).

6 Conclusion

We investigated inter-modal model alignment across text-image and biological-artificial neural representations. To achieve this, we introduced an inter-modal bridge cost for fusing feature spaces. Our results show this bridge cost enables effective alignment between distributions from separate modalities even with limited paired samples between source and target spaces. Furthermore, we found that global alignment (using samples from the computed optimal coupling) achieves competitive downstream performance while avoiding overfitting in noisy settings compared to true alignment (using labelled pairs). These findings emphasise two key factors for morphing quality: intra-space separation within feature spaces and inter-space alignment between them. However, the effectiveness of our method may be limited by the quality of pre-trained feature extractors and the availability of paired samples. Therefore, future work should focus on developing more disentangled representations to improve model reusability across modalities. Separately, our future work will look to validate this model space alignment approach on larger models, different datasets and modalities.

ACKNOWLEDGEMENTS

The authors thank Peter Dayan for his valuable feedback on the manuscript. This work was supported by the Max Planck Society.

REFERENCES

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Abdulkadir Canatar, Jenelle Feather, Albert Wakhloo, and SueYeon Chung. A spectral theory of neural prediction and alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):1–11, 2013.

Adrián Csiszárik, Péter Kőrösi-Szabó, Akos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. *Advances in Neural Information Processing Systems*, 34:5656–5668, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Marco Cuturi, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

Pinar Demetci, Rebecca Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh. Scot: single-cell multi-omics alignment with optimal transport. *Journal of computational biology*, 29(1):3–18, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Eric Elmoznino and Michael F Bonner. High-performing neural network models of visual cortex benefit from high latent dimensionality. *PLOS Computational Biology*, 20(1):e1011792, 2024.

Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch wasserstein: asymptotic and gradient properties. *arXiv preprint arXiv:1910.04091*, 2019.

Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets, 2024.

Team Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024. URL http://github.com/google/flax.

Matthieu Heitz, Nicolas Bonneel, David Coeurjolly, Marco Cuturi, and Gabriel Peyré. Ground metric learning on graphs. *Journal of Mathematical Imaging and Vision*, 63:89–107, 2021.

Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics: methodology and distribution*, pp. 162–190. Springer, 1992.

Guillaume Huguet, Alexander Tong, María Ramos Zapatero, Christopher J Tape, Guy Wolf, and Smita Krishnaswamy. Geodesic sinkhorn for fast and accurate optimal transport on manifolds. In *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2023.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Ehsan Imani, Wei Hu, and Martha White. Representation alignment in neural networks. *arXiv preprint arXiv:2112.07806*, 2021.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

L Kantorovich. On the transfer of masses (in Russian). In *Doklady Akademii Nauk*, volume 37, 1942.

Kohitij Kar, Jonas Kubilius, Kailyn Schmidt, Elias B Issa, and James J DiCarlo. Evidence that recurrent circuits are critical to the ventral streams execution of core object recognition behavior. *Nature neuroscience*, 22(6):974–983, 2019.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Dustin Klebe, Tal Shnitzer, Mikhail Yurochkin, Leonid Karlinsky, and Justin Solomon. Gera: Label-efficient geometrically regularized alignment. *arXiv preprint arXiv:2310.00672*, 2023.

Dominik Klein, Théo Uscidda, Fabian Theis, and Marco Cuturi. Generative entropic neural optimal transport to map within and across spaces. *arXiv preprint arXiv:2310.09254*, 2023.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *arXiv preprint arXiv:1912.11370*, 2(8), 2019.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Najib J Majaj, Ha Hong, Ethan A Solomon, and James J DiCarlo. Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. *Journal of Neuroscience*, 35(39):13402–13418, 2015.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. *arXiv preprint arXiv:2209.15162*, 2022.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences*, 1781.

Kevin R Moon, Jay S Stanley III, Daniel Burkhardt, David van Dijk, Guy Wolf, and Smita Krishnaswamy. Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ronan Perry, Gavin Mischler, Richard Guo, Theodore Lee, Alexander Chang, Arman Koul, Cameron Franz, Hugo Richard, Iain Carmichael, Pierre Ablin, Alexandre Gramfort, and Joshua T. Vogelstein. mvlearn: Multiview machine learning in python. *Journal of Machine Learning Research*, 22(109):1–7, 2021.

Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport: With Applications to Data Science*. Now Publishers, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Meyer Scetbon, Gabriel Peyré, and Marco Cuturi. Linear-time gromov wasserstein distances using low rank couplings and costs. In *International Conference on Machine Learning*, pp. 19347–19365. PMLR, 2022.

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv preprint*, 2018. URL https://www.biorxiv.org/content/10.1101/407007v2.

Thibault Séjourné, François-Xavier Vialard, and Gabriel Peyré. The unbalanced gromov wasserstein distance: Conic formulation and relaxation. *Advances in Neural Information Processing Systems*, 34:8766–8779, 2021.

Thibault Séjourné, Gabriel Peyré, and François-Xavier Vialard. Unbalanced optimal transport, from theory to numerics. *Handbook of Numerical Analysis*, 24:407–471, 2023.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pp. 6275–6284. PMLR, 2019.

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2023.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *International Conference on Machine Learning*, pp. 9625–9635. PMLR, 2020.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Titouan Vayer. A contribution to optimal transport on incomparable spaces. *arXiv preprint arXiv:2011.04447*, 2020.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18123–18133, 2022.

A Optimal transport

Optimal Transport (OT) was introduced by (Monge, 1781) as a way of transferring dirt from one place to another by minimising the transport cost between the source and target distributions. Given two probability measures $\mu \in \mathcal{P}(\mathcal{X})$and $\nu \in \mathcal{P}(\mathcal{Y})$, and a cost function $c(x, y)$ that quantifies the distance between pairs (x, y) where $x \in X$ and $y \in Y$, the objective is to find a push-forward map T that minimises the total cost of transporting mass from μ to ν. This problem can be mathematically formulated as finding the optimal map $T^\star$ that solves:

$$T^\star := \arg \inf_{T_\# \mu = \nu} \int_X c(x, T(x)) d\mu(x) \ , \tag{7}$$

subject to the constraint that the push-forward of μ under T equals ν. However, solving the Monge problem is challenging, and the map $T^\star$ may not be unique or even exist in some cases. Kantorovitch introduced a relaxation of the original Monge problem (Kantorovich, 1942) i.e., instead of seeking a deterministic mapping between two distributions, Kantorovich proposed finding a probabilistic mapping π, known as a coupling, which is a joint probability distribution over $\mathcal{X} \times \mathcal{Y}$. The Kantorovich problem is defined as:

$$\pi^\star := \arg \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \pi(x, y) \mathrm{d}x \mathrm{d}y. \tag{8}$$

For computational efficiency, the entropy-regularised version of this problem is usually considered in OT formulations, i.e. Eq. 1.

B Flow matching

Given a smooth time-varying vector field $v : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ we can define an ordinary differential equation (ODE):

$$\frac{dx}{dt} = v_t(x) \ , \tag{9}$$

where the solution is a flow denoted by $\phi_t(x)$ describing the trajectory of a point x over time with an initial condition $\phi_0(x) = x$. The evolution of an initial probability distribution $\rho_0 \in \mathcal{P}(\mathbb{R}^d)$ to a probability path $p_t(x)$ under this flow is governed by the continuity equation:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (p_t v_t) \tag{10}$$

The distribution p_t is the push forward of the initial distribution ρ_0 by the flow ϕ_t, denoted $p_t = (\phi_t)_\# \rho_0$, which describes how the distribution evolves under the influence of the flow.

In Continuous Normalising Flows (Chen et al., 2018) (CNFs), this vector field $v_{t,\theta}(x)$ is parameterised using a neural network (θ) that is optimised to satisfy the terminal condition $\rho_1 = (\phi_t)_\# \rho_0$, where ϕ_t is a flow associated with the vector field. Conditional Flow Matching (CFM) (Lipman et al., 2022) extends this such that a probability path is constructed using samples from source and target distributions and CNF vector field $v_\theta(t, x)$ learnt by optimising the following:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, x_0 \sim \rho_0, x_1 \sim \rho_1} \| v_{t,\theta}(t x_1 + (1 - t) x_0) - (x_1 - x_0) \|^2 \ , \tag{11}$$

where $t \sim U[0, 1]$.

C Related literature

Fusion techniques have been introduced to combine different modalities. For example, multi-modal encoders such as CLIP (Radford et al., 2021) and AIGN (Jia et al., 2021) map data from distinct domains into a shared representation using a contrastive objective (Oord et al., 2018). These models often surpass traditional approaches in zero-shot transfer tasks on new datasets (Chen et al., 2020; Kolesnikov et al., 2019), but typically require large amounts of paired data for training and come with substantial computational costs.

Model stitching (Lenc & Vedaldi, 2015) represents another line of work, where intermediate latent representations from one model are transformed into another by learning a stitching module.

This technique has been employed to align representations within models (Lenc & Vedaldi, 2015; Csiszárik et al., 2021) or across different modalities (Merullo et al., 2022). While effective in learning transformations, this method generally requires end-to-end training of the stitching module, which is impractical when only the latent representations of the source and target models are accessible during training.

Within a semi-supervised learning setup, Klebe et al. (2023) proposed to learn a shared embedding space by mapping the representations from two pre-trained multi-modal models into a common space. This approach necessitates only a small amount of labelled data but requires an additional model to be trained in the joint space for downstream tasks. In contrast, our approach directly identifies the transformation between the two latent representations, bypassing the need for a shared embedding space and additional models.

D ALGORITHM FOR LEARNING MULTI-MODAL BRIDGES

We proposed model space alignment via multi-modal bridges using three different alignment strategies. For this, we align latent space distributions by either solving an OT problem (local and global) or true paired samples (true), and then learn flow matching for out-of-sample predictions (Algorithm 1).

Algorithm 1 Learning inter-modal morphs via true, global, or local alignment

Input:
Two pre-trained models f_x and f_y
Two datasets $D_{\mathcal{X}}$ and $D_{\mathcal{Y}}$ ▷ source and target domains
Paired samples P ▷ optional paired samples
Entropy regularisation parameter ϵ
Unbalanced weighting parameters $\tau = (\tau_{\mathcal{X}}, \tau_{\mathcal{Y}})$
Reweighting neural networks η_θ and ξ_θ
Batch size b
Number of iterations T_{iter}
OT solver and cost function

$X \leftarrow f_x(D_{\mathcal{X}}), Y \leftarrow f_y(D_{\mathcal{Y}})$ ▷ Extract latent features using pre-trained models
$\pi \leftarrow \pi^{\text{true}}(X, Y, P)$ ▷ true alignment based on paired samples P using Eq.5
$C_{XY} \leftarrow \text{fused\_cost}(X, Y, P)$
$\pi \leftarrow OT_{\epsilon,\tau}(X, Y, C_{XY})$

for $t = 1, \ldots, T_{\text{iter}}$ **do**
  Sample $x_1, \ldots, x_b \sim X$ and $y_1, \ldots, y_b \sim Y$
  $C^b_{XY} \leftarrow \text{fused\_cost}\left([x_i]^b_{i=1}, [y_i]^b_{i=1}, P\right)$
  $\pi \leftarrow OT_{\epsilon,\tau}\left([x_i]^b_{i=1}, [y_i]^b_{i=1}, C^b_{XY}\right)$
  Sample $(i_1, j_1), \ldots, (i_b, j_b) \sim \pi$
  Sample $z_1, \ldots, z_b \sim \mathcal{N}(0, 1)$, $t_1, \ldots, t_b \sim \mathcal{U}([0, 1])$
  $\mathcal{L}(\theta) \leftarrow \sum_k \|v_{t,\theta}([z_k, y_{j_k}]|t, x_{i_k}) - (y_{j_k} - z_k)\|_2^2 +$
    $\sum_k \left(\eta_\theta(\mathbf{x}_k) - b\pi^k_{\mathcal{X}}\right)^2 + \left(\xi_\theta(\mathbf{y}_k) - b\pi^k_{\mathcal{Y}}\right)^2$.
  $\theta \leftarrow \text{Update}\left(\theta, \frac{1}{b}\nabla\mathcal{L}(\theta)\right)$
end for

Algorithm 1 was implemented in `JAX` (Bradbury et al., 2018) using `Flax` (Heek et al., 2024). For discrete OT solvers, we used the `OTT-JAX` library (Cuturi et al., 2022). To compute the KCCA between samples, we used the `MVLearn` library (Perry et al., 2021). We used these default hyperparameters for training – unless explicitly specified otherwise:

- **optimiser**: adam (learning rate = 10^{-4})
- **batch size** = 256

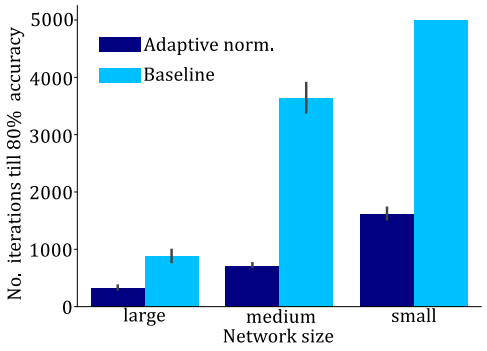


Figure 12: Iterations required to achieve 80% accuracy in MNIST experiments.

- **entropy regularisation** $\epsilon = 5 \times 10^{-3}$ with normalised cost matrices
- **fused penalty** $\alpha = 0.5$
- **max number of iterations** $T_{iter} = 10,000$
- **Unbalanced weighting parameters** $\tau = (1, 1)$
- **Reweighting neural networks** η_θ, ξ_θ: multi-layer perceptron (MLP) used in Klein et al. (2023).

Out-of-sample prediction At inference time, we solved Eq. 9 for $t_1 = 1$ using the velocity field $v_{t,\theta}$, a sampled point from the noise distribution z as the initial condition at $t_0 = 0$ conditioned on some out-of-sample point x.

The solution follows the form:

$$\hat{y} = \text{ODESolve}(v_{t,\theta}(\cdot \mid x), z, t_0 = 0, t_1 = 1), \tag{12}$$

where the function ODESolve numerically solves the ODE from t_0 to t_1, yielding $\hat{y}$ as the transported output at time $t_1 = 1$, while the condition x modifies the evolution of the ODE as necessary.

E Learning the velocity field

To learn the velocity field $v_{t,\theta}$ two different architectures were considered: 1) **MLP (i.e., Baseline):** three separate blocks for latent noise, time, and condition, which were concatenated and processed by a final MLP block following (Klein et al., 2023), and 2) **adaLN:** blocks with adaptive layer normalisation (adaLN) following (Perez et al., 2018). For each, we had three variations–small, medium, and large (Table 1)– with the SiLU activation function applied after every layer in all models. For each setting, we measured the number of iterations required to achieve 80% accuracy in image-to-text experiments on the MNIST dataset, using true alignment in a supervised setting. The results demonstrate the effectiveness of the AdaLN architecture, which converges more rapidly and attains acceptable performance levels more efficiently (Figure 12).

Model	Layers N	Hidden size d	Parameters
MLP-Small	4	256	700K
MLP-Medium	6	512	3M
MLP-Large	8	1680	49M
adaLN-Small	5	128	700K
adaLN-Medium	7	256	3M
adaLN-Large	8	1024	49M

Table 1: Different architectures for velocity field network$v_{t,\theta}$. Here, MLP-X architecture refers to the Klein et al. (2023) baseline.

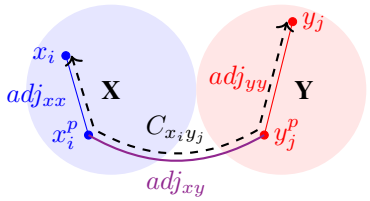


Figure 13: Overview of KNN fused cost. We construct a fused graph from K-Nearest Neighbour (KNN) graphs for spaces $\mathcal{X}$ (blue) and $\mathcal{Y}$ (red). Intra-space adjacency matrices adj_{XX} and adj_{YY} and adj_{XY} are used to form this graph. The fused graph allows estimation of the fused cost matrix C_{XY}^{knn} via shortest path distances using a heat kernel approximation.

E.1 Adaptive layer normalisation

Diffusion Transformers (DiT) (Peebles & Xie, 2023) and earlier works on diffusion models with U-net backbones (Dhariwal & Nichol, 2021) demonstrated the effectiveness of adaLN. For our formulation, we similarly replaced the standard MLP blocks with adaptive layer norm blocks. Unlike traditional layer normalisation, which directly learns the scale and shift parameters, adaLN regresses these parameters based on a time-dependent condition vector. The normalised output is then combined with the original input through a residual connection, where dimension-wise scaling parameters, initialised to zero, are applied. This adaptive mechanism enables more flexible and dynamic normalisation in response to the varying conditions during training.

F Fused costs

K-Nearest Neighbour cost The use of K-Nearest Neighbour (KNN) graphs and the shortest path distances, induced by Euclidean distance, has been proposed as a means to approximate geodesic distances on data manifolds (Crane et al., 2013). Notably, several studies have demonstrated the effectiveness of this data-driven cost function (Moon et al., 2018; Demetci et al., 2022; Huguet et al., 2023). Inspired by this approach, we first calculate the intra-space matrices C_{XX} and C_{YY} using Euclidean distance:

$$C_{XX}[i,j] = |x_i - x_j|^2, \quad C_{YY}[i,j] = |y_i - y_j|^2, \tag{13}$$

where $x_i, x_j \in \mathcal{X}$ and $y_i, y_j \in \mathcal{Y}$ represent data points in their respective spaces. Based on this, we compute intra-domain K-Nearest Neighbour adjacency matrix:

$$adj_{XX}[i,j] = |x_i - x_j|^2. \tag{14}$$

we construct an inter-space graph to approximate the fused-cost function. Given two intra-space k-nearest neighbour (kNN) adjacency matrix, adj_{XX} and adj_{YY}, and a paired set P, we define the inter-space graph G_{fused} as:

$$G_{\text{fused}} = \text{graph\_from\_adj} \begin{bmatrix} \text{adj}_{XX} & \text{adj}_{XY} \\ \text{adj}_{XY} & \text{adj}_{YY} \end{bmatrix} \tag{15}$$

where adj_{XX} and adj_{YY} are the adjacency matrices of G_{XX} and G_{YY}, respectively. The matrix adj_{XY} is:

$$\text{adj}_{XY}[i,j] = \begin{cases} 1 & \text{if } (x_i, y_j) \in P \\ 0 & \text{otherwise} \end{cases}$$

Then, the fused cost matrix C_{XY}^{knn} is the shortest path in G_{fused}:

$$C_{XY}^{knn}[i,j] = \text{ShortestPath}(G_{\text{fused}}, x_i, y_j)\ , \tag{16}$$

estimated using the heat kernel in our experiments (Crane et al., 2013; Heitz et al., 2021).

The heat kernel provides an approximation of the shortest path by modelling heat diffusion across the graph. Nodes that are closer in terms of the shortest path will exhibit faster heat diffusion, which allows us to estimate distances between them based on the behaviour of the heat kernel for small diffusion times.

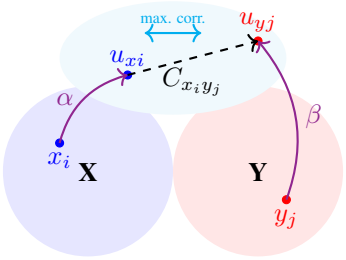


Figure 14: Overview of KCCA fused cost. Using paired samples, we find the projection vectors α and β to a joint space. The points x_i and y_j are then transformed into this joint correlation-maximising space, denoted as u_{xi} and u_{yj}. Finally, we compute the distance between the points in this joint space.

Kernel canonical correlation analysis cost Kernel canonical correlation analysis (Hardoon et al., 2004) (KCCA) extends canonical correlation analysis (CCA) (Hotelling, 1992) by projecting the data into a high-dimensional feature space using some kernel function $k(\cdot, \cdot)$. It then finds projections in this high-dimensional space that maximise the correlation between the two sets of data. Using the paired point matrices $X_p = [x_{ip}]_{p=1}^{l}$ and $Y_p = [y_{jp}]_{p=1}^{l}$, where $(x_{ip}, y_{jp}) \in P$, we calculate the projection vectors α and β (Hardoon et al., 2004):

$$\rho = \max_{\alpha,\beta} \frac{\alpha K_{X_p} K_{Y_p} \beta}{\sqrt{\alpha K_{X_p}^2 \alpha \cdot \beta K_{X_p}^2 \beta}}. \tag{17}$$

where these kernel matrices $K_{X_p} = k(X_p, X_p)$ and $K_{Y_p} = k(Y_p, Y_p)$ based on paired samples and a kernel function $k(\cdot, \cdot)$. we define the fused cost C_{XY}^{kcca} as:

$$\begin{aligned} u_X &= K_X \alpha \\ u_Y &= K_Y \beta \\ C_{XY}^{kcca}[i,j] &= \text{cosine\_distance}(u_X^{(i)}, u_Y^{(j)}) \,, \end{aligned} \tag{18}$$

where u_X and u_Y are the projections of the joint embedding space for the entire dataset. We used paired samples to compute the projection vectors α and β. For our experiments, KCCA is formalised using a Gaussian RBF kernel and paired points. By maximising the correlation between the projected variables from these two sets, KCCA tries to find a joint space for the maximum possible shared information.

G MODELS

G.1 MNIST EXPERIMENTS

Here, we provide details about the different models considered for text-image alignment using the MNIST dataset; including the pre-trained models (*VAE$_{image}$* and *VAE$_{text}$*) and the fusion baseline (Fused VAE) (Kingma & Welling, 2014):

G.1.1 *VAE$_{image}$* AND *VAE$_{text}$*

VAE$_{image}$ was trained on $50,000$ image samples of size $(28, 28, 1)$ from the MNIST training dataset, using mean binary cross-entropy as the reconstruction loss. The image pixel values were converted to 1 if the value was higher than 0.5, otherwise to 0. Separately, *VAE$_{text}$* was trained to compress one-hot encoded labels into a compact space and reconstruct the original labels from the input, and was trained using softmax cross entropy as the reconstruction loss. Table 2 provides architecture details.

G.1.2 VAE BASELINES

The baseline models – depending on the task were trained in an end-to-end fashion – for either reconstructing images from text using binary cross-entropy or reconstructing labels from image

Model Type	Architecture
VAE$_{text}$	Input: (10) Encoder: FC $64, 32$ Latents: 4 Decoder: FC $32, 16, 10$, softmax output
VAE$_{image}$	Input: $(28 \times 28 \times 1)$ Encoder: Conv $128, 256, 512$ Latents: 16 Decoder: ConvT $256, 128, 1$, sigmoid output

Table 2: *VAE*$_{image}$ and *VAE*$_{text}$ architectures. After each layer, we apply a ReLU non-linearity. For the convolutional (conv) layers, we used 3×3 kernels, with strides set to (2, 2) and *same* padding consistently across all models.

input using the softmax cross-entropy function. Each of these baselines was trained using different latent dimensions (large $= 128$ and small $= 16$; Table 3)

For the Fused VAE, we modified the vanilla VAE model to have two separate encoders and decoders for each modality. For this model, the encoders learn to map data from different modalities to a joint embedding space, and each decoder reconstructs the output based on the representation in this joint space. To construct the Fused VAE, we modified the ELBO:

$$\mathcal{L}_{\text{ELBO}-q_{\phi 1}} = \underbrace{\mathbb{E}_{q_{\phi 1}(\mathbf{z}|\mathbf{x})}\left[\log p_{\theta 1}(\mathbf{x}|\mathbf{z})\right]}_{\text{Reconstruction Loss 1}} + \underbrace{\mathbb{E}_{q_{\phi 1}(\mathbf{z}|\mathbf{x})}\left[\log p_{\theta 2}(\mathbf{y}|\mathbf{z})\right]}_{\text{Reconstruction Loss 2}} - \underbrace{\text{KL}(q_{\phi 1}(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))}_{\text{KL Divergence}}, \tag{19}$$

where q_{ϕ_1} and q_{ϕ_2} represent the distributions parameterisations by ϕ_1 and ϕ_2 for the first and second encoders, respectively. Similarly p_{θ_1} and p_{θ_2} denote the distributions parameterised by θ_1 and θ_2 for the first and second decoders. Given these, Fused VAE loss was defined as:

$$\mathcal{L}_{\text{ELBO-fused}} = \mathcal{L}_{\text{ELBO}-q_{\phi 1}} + \mathcal{L}_{\text{ELBO}-q_{\phi 2}} \ . \tag{20}$$

We trained two variants of the Fused VAE using paired samples from the two domains; Fused VAE (E2E) was trained on the raw data Fused VAE (Latent) was trained on the latent spaces of pre-trained (*VAE*$_{image}$) and (*VAE*$_{text}$). This allows for a nuanced comparison of how each model variant handles multi-modal data integration.

G.2 ImageNet Experiments

G.2.1 Vision Transformer

The Vision Transformer (ViT) (Dosovitskiy et al., 2021) leverages the transformer (Vaswani, 2017) architecture to process images by dividing them into patches and applying self-attention to capture global relationships. The original paper introduced three variants of ViT: ViT-B (Base), ViT-L (Large), and ViT-H (Huge), each differing in scale and complexity. For feature extraction for ImageNet, we used ViT-B model pre-trained (Wu et al., 2020) on ImageNet-21k (Deng et al., 2009) (which contains 14 million images and $21,843$ classes) at a resolution of 224×224. For image pre-processing, we followed the procedure outlined in Dosovitskiy et al. (2021).

G.2.2 Sentence Transformer

The Sentence Transformer (Reimers & Gurevych, 2019), commonly known as SBERT, converts sentences and paragraphs into embeddings that capture the high-level semantic meaning of the text. A common application of sentence transformers is measuring semantic similarity between sentences using cosine similarity. For our experiments, we used the pre-trained model `all-MiniLM-L6-v2` from the Hugging Face repository, which is based on the MiniLM architecture (Wang et al., 2020), to extract textual features from the input prompts.

Model Type	Task	Architecture
VAE (small)	text->image	Input: (10) Encoder: FC $64, 32$. Latents: 16 Decoder: ConvT $1024, 512, 1$, sigmoid output.
VAE (large)	text->image	Input: (10) Encoder: FC $64, 32$. Latents: 128 Decoder: ConvT 1024,512,1, sigmoid output.
VAE (small)	image->text	Input: $(28 \times 28 \times 1)$ Encoder: Conv $512, 1024, 2048$. Latents: 16 Decoder: FC $32, 16, 10$, softmax output.
VAE (large)	image->text	Input: $(28 \times 28 \times 1)$ Encoder: Conv $512, 1024, 2048$. Latents: 128 Decoder: FC $32, 16, 10$, softmax output.
Fused VAE (E2E)	image<->text	Input1: $(28 \times 28 \times 1)$ Input2: (10) Encoder1: Conv $512, 1024, 2048$. Encoder2: FC $64, 32$. Latents: 128 Decoder1: ConvT $1024, 512, 1$, sigmoid output. Decoder2: FC $32, 16, 10$, softmax output.
Fused VAE (latent)	image<->text	Input1: (16) Input2: (4) Encoder1: FC 8×1024 Encoder2: FC 8×1024 Latents: 128 Decoder1: FC $8 \times 1024, 16$. Decoder2: FC $8 \times 1024, 4$.

Table 3: Baseline models architectures for the MNIST experiments. Here, FC is for the fully connected layer, Conv is for the convolutional layer and ConvT is for the convolutional transpose layer.

G.3 Majaj et al. (2015) Experiments

G.3.1 EfficientNet-B0

For our experiments, we used a pre-trained EfficientNet-B0 (Tan & Le, 2019) (top-1 accuracy: 0.76 on ImageNet) to extract artificial neural network activations. Briefly, this entailed extracting the activations from all ReLU non-linearities after each intermediate convolutional layer for each unique stimulus in the dataset (Canatar et al., 2024). Following (Schrimpf et al., 2018), we kept the first 1000 principal components per layer using 1000 validation images. For all further analysis, we selected layers based on their performance in neural predictivity (Brain-Score)(Schrimpf et al., 2018), effective dimensionality (ED)(Elmoznino & Bonner, 2024), and representational similarity analysis (RSA) (Kriegeskorte et al., 2008) for IT and V4 regions. The final layers selected for morphing were: *layer* 1, *layer* 2, *layer* 3, *layer* 5, *layer* 6, and *features*. Note, the same process can be applied to extract neural activity from other networks as well e.g. ResNets (He et al., 2016), ConvNeXts (Liu et al., 2022), ViTs (Dosovitskiy et al., 2021) .

H Metrics

Here, we introduce the different metrics used in Sec. 5.

H.1 FEATURE OVERLAP

We compute the overlap in the feature space, assuming the feature space is Euclidean, for a pre-trained model using a metric derived from k-nearest neighbours (kNN). We randomly select a batch of b samples, $X = \{x_i\}_{i=1}^{b}$, from the domain $\mathcal{X}$, with corresponding labels $L = \{l_i\}_{i=1}^{b}$. First, we calculate the k-nearest neighbours for each point x_i in the feature space. For each point x_i, the overlap is defined as the proportion of its k-nearest neighbours that have a label different from l_i. If $\psi_{knn}(x_i)$ represents the set of the k-nearest neighbours of x_i, then the overlap for a given point x_i is:

$$\text{overlap}(x_i) := \frac{1}{k} \sum_{x_j \in \psi_{knn}(x_i)} \mathbb{1}(l_j \neq l_i) \,, \tag{21}$$

where $\mathbb{1}(l_j \neq l_i)$ is the indicator function, which equals 1 if the label l_j of the neighbour x_j differs from the label l_i, and 0 otherwise. We approximate the feature space overlap as the mean of the individual overlaps for all samples x_i in the batch. This metric provides a measure of how often points in the feature space are surrounded by neighbours with different labels, reflecting the degree of separation within the feature space. For all experiments reported in Sec. 5, we used $k = 15$ and five randomly selected batches, each with size $b = 100K$ for ImageNet experiment and the $b = 5K$ for MNIST expriments.

H.2 EVALUATING MNIST

Image-to-text We use the reconstruction cost as the reported test accuracy for the VAE and fused-VAE (E2E) baseline models. For models that rely on latent spaces, such as morphing models and latent VAE, after transforming the test split images from the VAE_{image} to the VAE_{text} latent space, we reconstruct the labels using the VAE_{text} decoder and report this as the test accuracy.

Text-to-image: For the baseline models with end-to-end training we construct images from the labels. However, for other models that transfer representations between latent spaces, we use the respective transformed representations and construct an image using VAE_{image}. Assuming we have n text-image pairs $\{(x_i, y_i)\}_{i=1}^{n}$ in the test dataset, where each $x_i \in \{0,1\}^{28\times 28}$, and let $\{\tilde{x_i}\}_{i=1}^{n}$ be the samples reconstructed using the transportation method T. Assume that S_c is the set of indices such that $i \in S_c \Rightarrow y_i = c$ and $|S_c| = N_c$. We construct the 2D pixel-wise distributions for the original and reconstructed images for each class c:

$$\begin{aligned} P(X^{j,k} \mid c) &= \frac{1}{N_c} \sum_{i \in S_c} x_i^{j,k}, \\ P(\tilde{X}^{j,k} \mid c) &= \frac{1}{N_c} \sum_{i \in S_c} \tilde{x}_i^{j,k}, \end{aligned} \tag{22}$$

where $X^{j,k}$ represents the pixel at position (j,k) in the original images, and $\tilde{X}^{j,k}$ represents the corresponding pixel in the reconstructed images. From this, we calculate the Next, we define the mean squared error (MSE) metric between the pixel-wise distributions for the original images (X) and the reconstructed ones ($\tilde{X}$) for class c, denoted as $MSE(c)$:

$$MSE(c) = \frac{1}{28^2} \sum_{j=1}^{28} \sum_{k=1}^{28} \Big(P(X^{j,k} \mid c) - P(\tilde{X}^{j,k} \mid c) \Big)^2 . \tag{23}$$

Finally, the mean squared error for the transportation function is defined as:

$$MSE = \frac{1}{10} \sum_{c=0}^{9} MSE(c). \tag{24}$$

H.3 Evaluating ImageNet

For the ImageNet experiments, after training the velocity field $v_{t,\theta}$, for each image d_j^x with label l_j in the test split, we compute its representation x_j in the embedding space of the pre-trained image model. We then use Eq.12 to obtain the corresponding prediction $\hat{y}_j$ in the target space.

The ImageNet dataset contains 1000 unique classes, and we assume their representations in the embedding space of the language domain are denoted as $Y = [y_i]_{i=1}^{1000}$, corresponding to the labels $[l_i]_{i=1}^{1000}$. To classify each predicted point $\hat{y}_j$, we compute the cosine distance between $\hat{y}_j$ and all points y_i in the target space. The nearest neighbour y_{l_k} is the point that minimises cosine distance:

$$k = \arg\min_i \text{cosine\_distance}(\hat{y}_j, y_i) \tag{25}$$

where l_k is the label corresponding to the closest. The accuracy is then computed as follows:

$$\text{accuracy} = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}(l_j = l_k) \tag{26}$$

where $\mathbb{1}$ is the indicator function and n is the total number of test samples.

H.4 Calculating intra-space cost for Majaj et al. (2015)

To compute the intra-space cost matrix C_{XX} for neural activity responses, we use a correlation-driven cost similar to Yamins et al. (2014). Let $x_i, x_j \in \mathcal{X}$ be the neural responses to stimuli s_i and s_j, respectively. We define the cost matrix as:

$$C_{XX}(x_i, x_j) = 1 - \frac{\text{cov}(x_i, x_j)}{\sqrt{\text{var}(x_i)\text{var}(x_j)}}. \tag{27}$$

Similar to previous experiments, we normalise the intra-space cost matrix by its mean before computing the optimal coupling.

H.5 Artificial-to-biological neural representation evaluation

Following Majaj et al. (2015); Kar et al. (2019), we evaluated the morphs' performance in a category object classification task (Fig. 11) using support vector machine (SVM) classifiers Chang & Lin (2011). For V4 and IT region, we trained separate SVMs to decode neural activity corresponding to the category of core images using the entire training dataset. We employed a linear C-SVC model with a linear kernel and hinge loss with L_2 regularisation, and performed 5-fold cross-validation for hyperparameter optimisation.

I GENOT Validation

To ensure that baselines were consistent with reported results in Klein et al. (2023), we replicate the results for the Swiss roll ($\mathbb{R}^3$; source distribution) to spiral ($\mathbb{R}^2$; target distribution) using the GW solver (unsupervised) and local alignment. Fig.15 shows the evolution of the noise distribution into the target distribution in $\mathbb{R}^2$ space.

J Alignment strategies

We evaluated the runtime per iteration for various alignment strategies (Fig.16) using the same computational setup. We observed that local alignment exhibits significantly higher time complexity compared to other strategies. This is primarily due to the necessity of solving an OT problem at each iteration (Algorithm 1). In scenarios involving more complex distributions, such as those in the ImageNet dataset, the difference in computational cost becomes even more pronounced. The need to compute the fused cost and solve the OT problem in each iteration further exacerbates the time complexity in such cases.

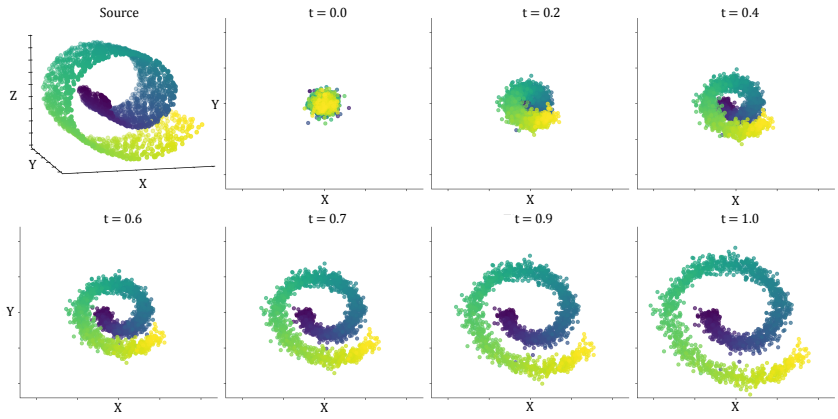


Figure 15: Mapping from Swiss roll to Spiral using local alignment

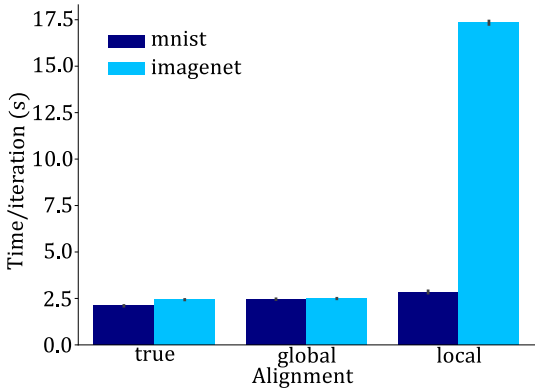


Figure 16: Runtime per iteration (in seconds) for different alignment strategies across MNIST and ImageNet datasets.

K OT SOLVER OPTIMISATION

We examined the effectiveness of different OT discrete solvers in learning the optimal coupling, using large sample sizes ($100,000$ for ImageNet and $5,000$ for MNIST). Both global and local alignment strategies rely on OT discrete solvers, which can be either linear or fused FGW, and the performance of these solvers has a significant impact on the morphing quality. The FGW solver integrates intra-domain costs, C_{XX} and C_{YY}, derived using cosine distance, and an inter-domain cost, which is computed based on the formulation introduced in Sec.3.3 and Appendix F. Additionally, the hyperparameter α controls the trade-off between quadratic and linear OT objectives.

For the ImageNet and MNIST experiments, we evaluated the matching accuracy of the discrete solvers as a function of the ratio of paired samples, varying both the values of α and the inter-domain cost function. In accordance with Eq. 3, as $\alpha \to 1$, the quadratic costs dominate over the linear component, favouring a solution that leans towards unsupervised alignment, as paired samples are used exclusively in the inter-domain cost. Conversely, when $\alpha = 0$, the problem simplifies to a linear OT problem.

L UNBALANCED SETTING

L.1 THEORETICAL BACKGROUND FOR U-GENOT

Unbalanced optimal transport (U-OT) is an extension of the classical OT problem, where the marginals of the optimal coupling $\pi^\star$ found in Eqs. 1-3 can differ from the true source (μ) and

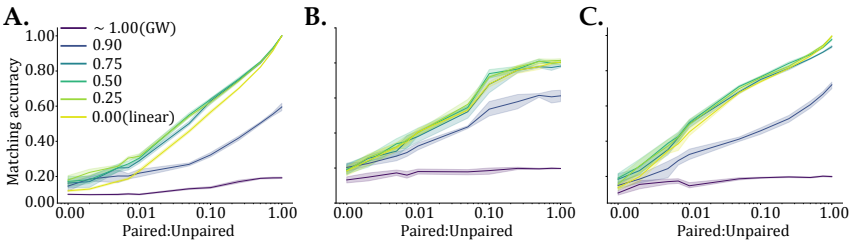


Figure 17: Matching accuracy across different values of α for discrete OT solvers in the MNIST experiment. Accuracy was computed by evaluating the correct matches from the optimal coupling $\pi^\star$ for different fused costs. **A)** KNN, **B)** KCCA, **C)** Bridge.

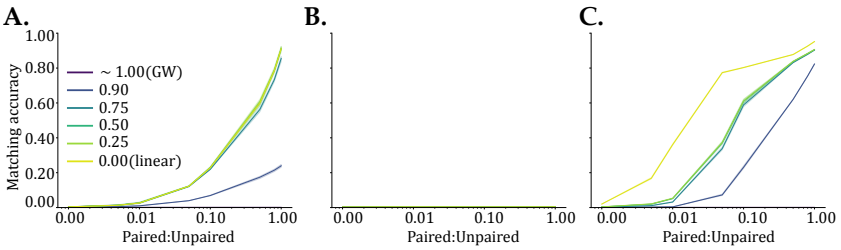


Figure 18: Matching accuracy across different values of α for discrete OT solvers in the ImageNet experiment. Accuracy was computed by evaluating the correct matches from the optimal coupling $\pi^\star$ for different fused costs. **A)** KNN, **B)** KCCA, **C)** Bridge.

target (ν) distributions (Séjourné et al., 2023). By relaxing the constraint that mass must be exactly preserved between distributions, U-OT can ignore or down-weight outliers and noisy samples that would otherwise force suboptimal transport plans. This makes U-OT particularly well-suited for aligning neural activity data, which often contains measurement noise.

The unbalanced weighting parameters, $\lambda_\mathcal{X}$ and $\lambda_\mathcal{Y}$, control the extent to which the marginals of the optimal coupling can diverge from the true source and target distributions. We follow the convention used by Klein et al. (2023) to define:

$$\tau_\mathcal{X} = \frac{\lambda_\mathcal{X}}{\lambda_\mathcal{X} + \epsilon}, \quad \tau_\mathcal{Y} = \frac{\lambda_\mathcal{Y}}{\lambda_\mathcal{Y} + \epsilon}, \tag{28}$$

where we recover the classical OT problem by setting $\tau_i = 1$ when $\lambda_i \to \infty$. We note that the unbalancedness parameter τ_i is influenced by the entropy regularisation parameter ϵ in this definition.

When using unbalanced OT solvers in U-GENOT, two reweighting functions, $\eta : \mathcal{X} \to \mathbb{R}^+$ and $\xi : \mathcal{Y} \to \mathbb{R}^+$, are employed for the source and target space, respectively. These reweighting functions are defined as $\pi^\star_\mathcal{X} = \eta \cdot \mu$ and $\pi^\star_\mathcal{Y} = \xi \cdot \nu$ in the unbalanced setting. Practically, these functions can be approximated by parameterising neural reweighting functions, η_θ and ξ_θ, which are trained to re-balance the U-OT using Eq. 4.

L.2 Experimental considerations for aligning neural representations

For the neural activity model experiments presented in Sec.5.2, we used the unbalanced OT setting to account for noise in the neural recordings. However, the choice of unbalanced weighting parameters was empirically determined. In our case, since we are using a global strategy (with a low number of data points and low memory requirements), we solve the OT problem only once. The quality of the learned mapping by U-GENOT depends on the performance of the discrete solver. Therefore, we tuned the hyperparameters τ_i and ϵ for different layers of EfficientNet and varying numbers of paired points.

It is worth noting that the source and target distributions are uniform over the training set. However, when using the unbalanced setting, some samples may be excluded from the joint distribution by

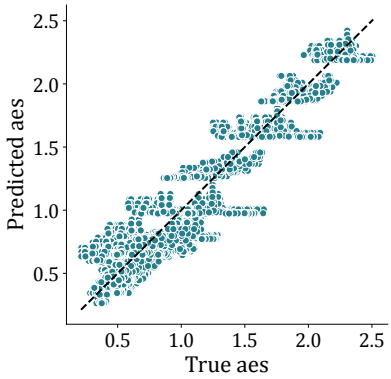


Figure 19: Predicted versus true aes ratio.

assigning them low probability. When the unbalancedness parameters are low ($\tau_i < 0.9$), the true strategy is recovered, as the paired points have zero inter-space cost. However, this can lead to issues (Fig. 20) and requires tuning the hyperparameters to maximise matching accuracy under the optimal coupling $\pi^\star_{\epsilon,\tau}$, i.e., to minimise the number of excluded samples from the marginals using the validation set. We define the ratio of excluded samples for a coupling π:

$$\text{excluded\_ratio}(\pi) = 1 - \frac{1}{N}\left|\{x_1, \dots x_N \sim \pi_{\mathcal{X}}\}\right| - \frac{1}{N}\left|\{y_1, \dots y_N \sim \pi_{\mathcal{Y}}\}\right| \ , \tag{29}$$

where N is the total number of data points in the validation set and $|\cdot|$ shows the set size. Using this, we defined the accuracy to excluded samples (i.e., aes) ratio to optimise this trade-off:

$$\text{aes} = \frac{\text{matching\_acc}(\pi)}{\text{excluded\_ratio}(\pi)} \ . \tag{30}$$

To find a general rule for all layers and regions, we specified ϵ, $\tau_{\mathcal{X}}$, $\tau_{\mathcal{Y}}$, and the paired:unpaired ratio as independent variables, with aes ratio as the dependent variable. Fitting an ordinary least squares regression model resulted in an adjusted R^2 of 0.909 (Fig. 19).

For the experiments reported in Sec. 5.2 we set the entropy regularisation parameter to $\epsilon = 10^{-3}$ and show in Fig. 20 the average value of the aes ratio for different pairwise combinations of $\tau_{\mathcal{X}}$ and $\tau_{\mathcal{Y}}$ as well as the paired:unpaired ratio. We found that $\tau_{\mathcal{X}} = \tau_{\mathcal{Y}} = 0.99$ is suitable for varying levels of paired:unpaired sampled.

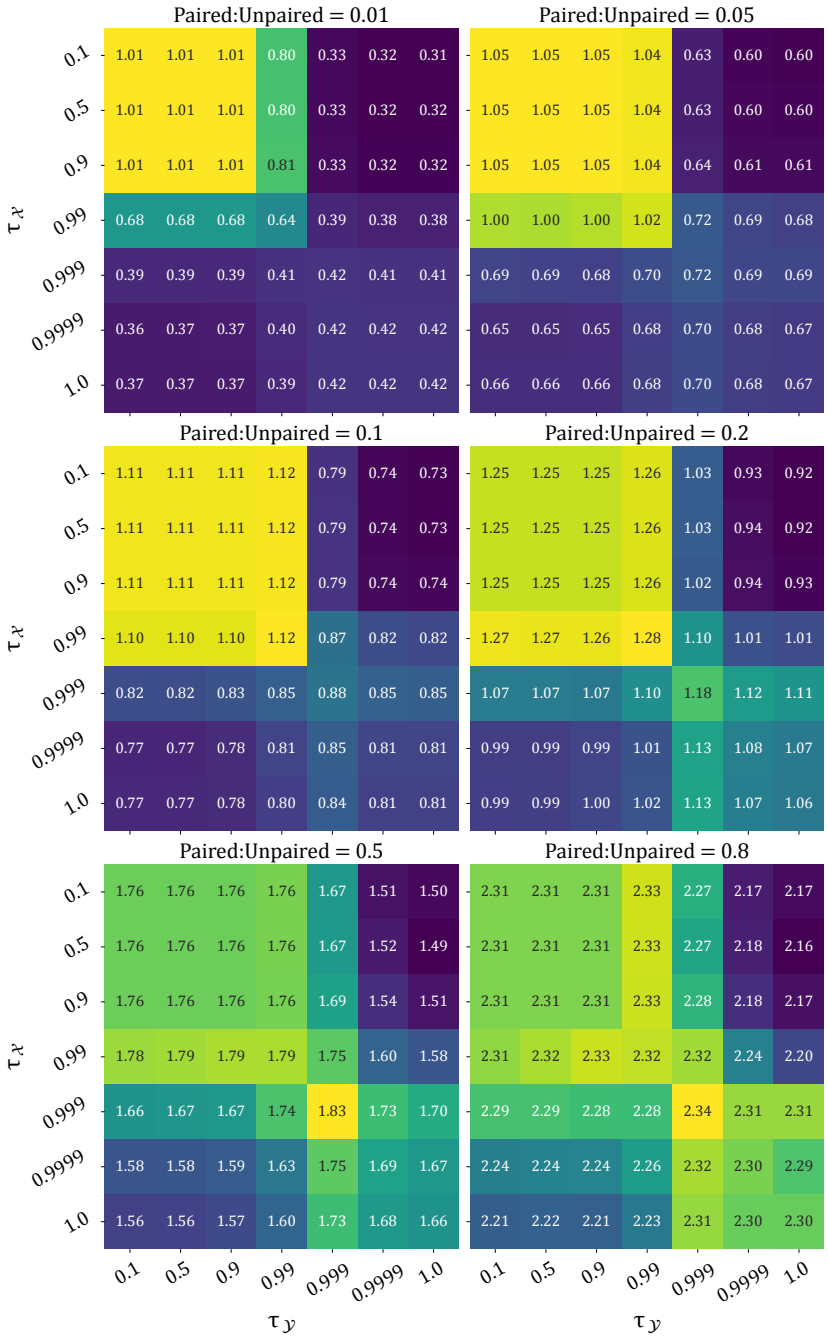


Figure 20: Heatmaps of the average values of the aes ratio as a function of $\tau_{\mathcal{X}}$ and $\tau_{\mathcal{Y}}$, with $\epsilon = 10^{-3}$.