Harnessing the Universal Geometry of Embeddings

Rishi Jha Collin Zhang Vitaly Shmatikov John X. Morris
Department of Computer Science
Cornell University

Abstract

We introduce the first method for translating text embeddings from one vector space to another without any paired data, encoders, or predefined sets of matches. Our unsupervised approach translates any embedding to and from a universal latent representation (i.e., a universal semantic structure conjectured by the Platonic Representation Hypothesis). Our translations achieve high cosine similarity across model pairs with different architectures, parameter counts, and training datasets.

The ability to translate unknown embeddings into a different space while preserving their geometry has serious implications for security. An adversary with access to a database of only embedding vectors can extract sensitive information about underlying documents, sufficient for classification and attribute inference.

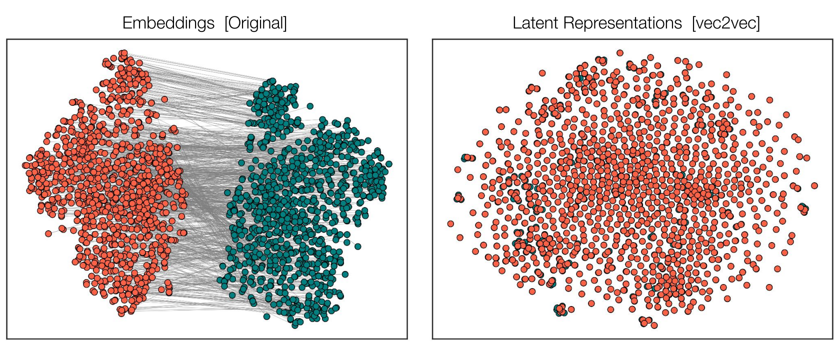

Figure 1: Left: input embeddings from different model families (T5-based GTR [47] and BERT-based GTE [32]) are fundamentally incomparable. Right: given unpaired embedding samples from different models on different texts, our model learns a latent representation where they are closely aligned.

1 Introduction

Text embeddings are the backbone of modern NLP, powering tasks like retrieval, RAG, classification, and clustering. There are many embedding models trained on different datasets, data shufflings, and initializations. An embedding of a text encodes its semantics: a good model maps texts with similar semantics to vectors close to each other in the embedding space. Since semantics is a property of text, different embeddings of the same text should encode the same semantics. In practice, however, different models encode texts into completely different and incompatible vector spaces.

The Platonic Representation Hypothesis Huh et al. [18] conjectures that all vision models of sufficient size converge to the same latent representation. We propose a stronger, constructive version of this hypothesis for text models: the universal latent structure of text representations can be learned and, furthermore, harnessed to translate representations from one space to another without any paired data or encoders.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

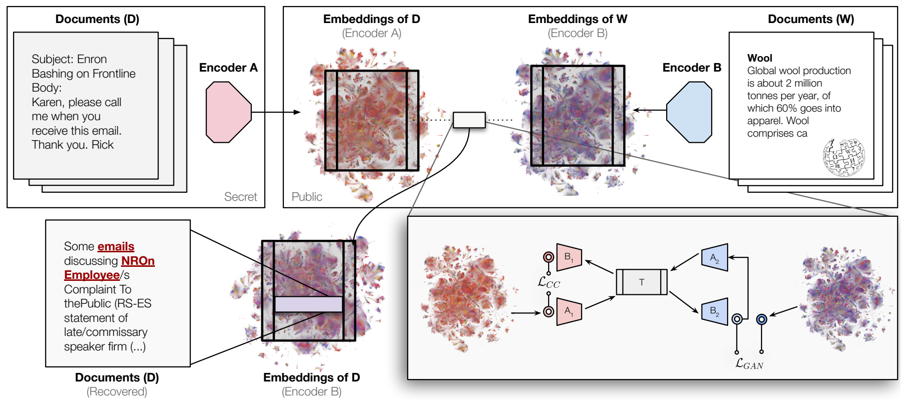

Figure 2: Given only a vector database from an unknown model, `vec2vec` translates the database into the space of a known model using latent structure alone. Converted embeddings reveal sensitive information about the original documents, such as the topic of an email (pictured, real example).

In this work, we show that the Strong Platonic Representation Hypothesis holds in practice. Given unpaired examples of embeddings from two models with different architectures and training data, our method learns a latent representation in which the embeddings are almost identical (Figure 1).

We draw inspiration from research on aligning word embeddings across languages [62, 10, 15, 9] and unsupervised image translation [36, 70]. Our `vec2vec` method uses adversarial losses and cycle consistency to learn to encode embeddings into a shared latent space and decode with minimal loss. This makes unsupervised translation possible. We use a basic adversarial approach with vector space preservation [46] to learn a mapping from an unknown embedding distribution to a known one.

`vec2vec` is the **first method to successfully translate embeddings from the space of one model to another without paired data.**[1] `vec2vec` translations achieve cosine similarity as high as 0.96 to the ground-truth vectors in their target embedding spaces and perfect matching on over 8000 shuffled embeddings (without access to the set of possible matches in advance).

To show that our translations preserve not only the relative geometry of embeddings but also the semantics of underlying inputs, we extract information from them using zero-shot attribute inference and inversion, without any knowledge of the model that produced the original embeddings.[2]

2 Problem formulation: unsupervised embedding translation

Consider a collection of embedding vectors $\{u_1, \ldots u_n\}$, for example, a dump of a compromised vector database, where each $u_i = M_1(d_i)$ is generated by an unknown encoder $M_1 : \mathbb{V}^s \to \mathbb{R}^{d_{M_1}}$ from an unknown document d_i. We cannot make queries to M_1 and do not know its training data, nor architectural details. Our goal is to extract any information about the documents d_i.

We do assume access to a different encoder M_2 that we can query at will to generate new embeddings in some other space. We also assume high-level distributional knowledge about the hidden documents: their modality (text) and language (e.g., English). To extract information, we may translate $\{u_1, \ldots u_n\}$ into the output space of M_2 and apply techniques like inversion that require the encoder.

Limitations of correspondence methods. There is significant prior research on the problem of *matching* or *correspondence* between sets of embedding vectors [1, 49, 8, 54]. These methods typically assume that the two (or more) sets of embeddings are generated by different encoders on the *same or highly-overlapping inputs*. In other words, for each unknown vector, there must already exist a set of candidate vectors in a different embedding. In practice, it is unrealistic to expect that such a database be available, so these methods are not directly applicable. Some matching methods,

[1]Prior work has successfully translated *word* embeddings between languages, typically relying on overlapping vocabularies across languages. In contrast, we translate embeddings of entire sequences between model spaces.

[2]Our code is available on GitHub.

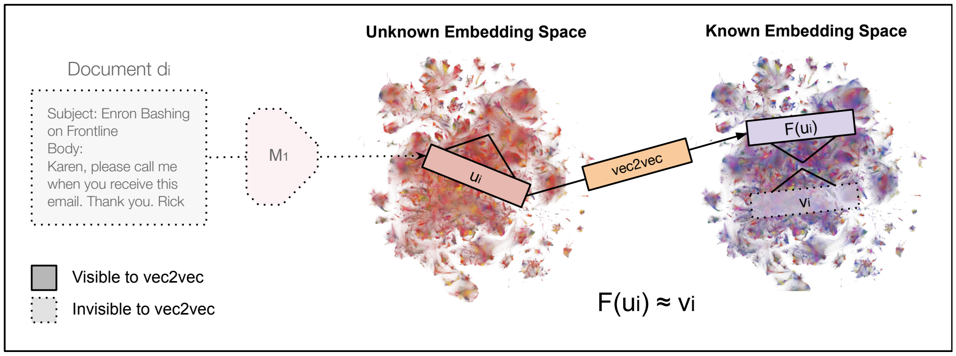

Figure 3: **Unsupervised embedding translation**. With access to only $u_i = M_1(d_i)$, `vec2vec` seeks to generate a translation $F(u_i)$ that is close in M_2's embedding space to the ideal embedding $v_i = M_2(d_i)$ without access to d_i, v_i, or M_1.

however, support translation between embedding spaces without overlapping inputs. Our experiments demonstrate that these methods struggle significantly, even when correspondence exists.

Our task is inherently more challenging than matching, because we do not assume access to encoder M_1, nor do we have additional representations of documents $d_1, \ldots, d_n$ beyond their embeddings $u_i = M_1(d_i)$. Therefore, we rely solely on unsupervised *translation* from M_1 to M_2. The effectiveness of such unsupervised translation approaches thus critically depends on identifying and leveraging shared geometric structures within the embedding spaces produced by M_1 and M_2.

The Strong Platonic Representation Hypothesis. Our hope that unsupervised embedding translation is possible at all rests on the stronger version of the Platonic Representation Hypothesis [18]. Our conjecture is as follows: *neural networks trained with the same objective and modality, but with different data and model architectures, converge to a universal latent space such that a translation between their respective representations can be learned without any pairwise correspondence.*

Translation enables information extraction. Solving unsupervised translation will allow us to use information extraction tools designed to operate on vectors produced by known encoders. For example, we could apply inversion models [43, 67] to recover unknown documents $\{d_i\}$.

3 Our method: `vec2vec`

Unsupervised translation has been successful in computer vision, using a combination of cycle consistency and adversarial regularization [36, 70]. Our design of `vec2vec` is inspired in part by these methods. We aim to learn embedding-space translations that are cycle-consistent (mapping to and from an embedding space should end in the same place) and indistinguishable (embeddings for the same text from either space should have identical latents).

3.1 Architecture

We propose a modular architecture, where embeddings are encoded and decoded using space-specific adapter modules and passed through a shared backbone network. Figure 2 shows these components. Input adapters $A_1 : \mathbb{R}^d \to \mathbb{R}^Z$ and $A_2 : \mathbb{R}^d \to \mathbb{R}^Z$ transform embeddings from each encoder-specific space into a universal latent representation of dimension Z. The shared backbone $T : \mathbb{R}^Z \to \mathbb{R}^Z$ extracts a common latent embedding from adapted inputs. Output adapters $B_1 : \mathbb{R}^Z \to \mathbb{R}^d$ and $B_2 : \mathbb{R}^Z \to \mathbb{R}^d$ translate these common latent embeddings back into the encoder-specific spaces. Thus, translation functions F_1, F_2 and additional reconstruction mappings R_1, R_2 are defined as:

$$F_1 = B_2 \circ T \circ A_1, \quad F_2 = B_1 \circ T \circ A_2 \quad R_1 = B_1 \circ T \circ A_1 \quad R_2 = B_2 \circ T \circ A_2$$

Parameters of all components are collectively denoted $\theta = \{A_1, A_2, T, B_1, B_2\}$.

Unlike images, embeddings do not have any spatial bias. Instead of CNNs, we use multilayer perceptrons (MLP) with residual connections, layer normalization, and SiLU nonlinearities. Discriminators mirror this structure but omit residual connections to simplify adversarial learning.

3.2 Optimization

In addition to the 'generator' networks F and R, we introduce discriminators operating on both the latent representations of F (D_1^ℓ, D_2^ℓ) and the output embeddings (D_1, D_2).

Our goal is to train the parameters of θ by solving:

$$\theta^* = \arg\min_\theta \max_{D_1, D_2, D_1^\ell, D_2^\ell} \mathcal{L}_{\text{adv}}(F_1, F_2, D_1, D_2, D_1^\ell, D_2^\ell) + \lambda_{\text{gen}} \mathcal{L}_{\text{gen}}(\theta), \tag{1}$$

where $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{gen}}$ represent adversarial and generator-specific constraints respectively and hyperparameter λ_{gen} controls their tradeoff.

Adversarial. The adversarial loss encourages generated embeddings to match the empirical distributions of original embeddings both at the embedding and latent levels. Specifically, applying the standard GAN loss formulation [13] to both levels yields:

$$\begin{aligned}\mathcal{L}_{\text{adv}}(F_1, F_2, D_1, D_2, D_1^\ell, D_2^\ell) = \mathcal{L}_{\text{GAN}}(D_1, F_1) + \mathcal{L}_{\text{GAN}}(D_2, F_2) \\ + \mathcal{L}_{\text{GAN}}(D_1^\ell, T \circ A_1) + \mathcal{L}_{\text{GAN}}(D_2^\ell, T \circ A_2).\end{aligned}$$

Generator. Because adversarial losses alone do not guarantee that translated embeddings preserve semantics [70], we introduce three additional constraints to help the generator learn a useful mapping:

Reconstruction enforces that an embedding, when mapped into the latent space and back into its original embedding space, closely matches its initial representation:

$$\mathcal{L}_{\text{rec}}(R_1, R_2) = \mathbb{E}_{x\sim p} \|R_1(x) - x\|_2^2 + \mathbb{E}_{y\sim q} \|R_2(y) - y\|_2^2.$$

where p and q are distributions of embeddings sampled from M_1 and M_2, respectively.

Cycle-consistency acts as an unsupervised proxy for supervised pair alignment, ensuring that F and G can translate an embedding to the other embedding space and back again with minimal corruption:

$$\mathcal{L}_{\text{CC}}(F_1, F_2) = \mathbb{E}_{x\sim p} \|F_2(F_1(x)) - x\|_2^2 + \mathbb{E}_{y\sim q} \|F_1(F_2(y)) - y\|_2^2.$$

Vector space preservation (VSP) ensures that pairwise relationships between translated embeddings are consistent with the target space [46, 65]. Given a batch of B embeddings $x_1, ..., x_B$ and $y_1, ..., y_B$, we sum their average pairwise distances after translation by both F_1 and F_2:

$$\begin{aligned}\mathcal{L}_{\text{VSP}}(F_1, F_2) = \frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} \Big[&\|M_1(x_i) \cdot M_1(x_j) - F_2(M_2(y_i)) \cdot F_2(M_2(y_j))\|_2^2 \\ &+ \|M_2(y_i) \cdot M_2(y_j) - F_1(M_1(x_i)) \cdot F_1(M_1(x_j))\|_2^2 \Big]\end{aligned}$$

Combining these losses yields: $\mathcal{L}_{\text{gen}}(\theta) = \lambda_{\text{rec}} \mathcal{L}_{\text{rec}}(R_1, R_2) + \lambda_{\text{CC}} \mathcal{L}_{\text{CC}}(F_1, F_2) + \lambda_{\text{VSP}} \mathcal{L}_{\text{VSP}}(F_1, F_2)$, where hyperparameters λ_{CC}, λ_{rec}, and λ_{VSP} control relative importance.

4 Experimental setup

4.1 Preliminaries

Datasets. We use the *Natural Questions (NQ)* [25] dataset of user queries and Wikipedia-sourced answers for training (a 2-million subset) and evaluation (a 65536 subset). To evaluate information extraction, we use *TweetTopic* [2], a dataset of tweets multi-labeled by 19 topics; a random 8192-record subset of *Pseudo Re-identified MIMIC-III (MIMIC)* [28], a pseudo re-identified version of the MIMIC dataset [19] of patient records multi-labeled by 2673 MedCAT [24] disease descriptions; and a random 50-email subset of the *Enron Email Corpus (Enron)* [21], an unlabeled, public dataset of internal emails from a defunct energy company. In Appendix D, we ablate a model on *MS COCO* [34], a captioned image dataset, to evaluate performance on multimodal retrieval.

Models. Table 1 lists the embedding models representing four size categories, five transformer backbones, and two output dimensionalities. Granite is multilingual; CLIP is multimodal. Since Qwen is very compute-intensive, we only evaluate it for a single model pair in Appendix C.

Model	Params (M)	Backbone	Year	Dims	Max Seq.
[47] gtr	110	T5	2021	768	512
[50] clip	151	CLIP	2021	512	77
[58] e5	109	BERT	2022	768	512
[32] gte	109	BERT	2023	768	512
[68] stella	109	BERT	2023	768	512
[14] granite	278	RoBERTa	2024	768	512
[69] qwen	4000	Qwen3	2025	2560	32K

Table 1: Embedding models used in our experiments.

Training. Unless otherwise specified, each `vec2vec` is trained on two sets of embeddings generated from disjoint sets of 1 million 64-token sequences sampled from NQ (see Section 7 for experiments with fewer embeddings). Due to GAN instability [53], we select the best of multiple initializations (see Appendix E) and leave more robust training to future work. See Appendix A for compute details.

4.2 Evaluating translation

Let $u_i = M_1(d_i)$ and $v_i = M_2(d_i)$ denote the source and target embeddings of the same input d_i. The goal of translation is to generate a vector that is as close to v_i as possible. We say that (u_i, v_j) are "aligned" by the translator F if v_j is the closest embedding to $F(u_i)$: $j = \arg\min_k \cos\big(F(u_i), v_k\big)$. A perfect translator F^* satisfies $i = \arg\min_k \cos\big(F^*(u_i), v_k\big)$ for all i.

Given (unknown) embeddings $\{M_2(d_j)\}_{j=0}^{n}$ ordered by decreasing cosine similarity to $F(u_i)$, let r_i be the rank of the correct embedding $v_i = M_2(d_i)$. To measure quality of F, we use three metrics. **Mean Cosine Similarity** measures how close translations are, on average, to their targets. **Top-1 Accuracy** is the fraction of translations whose target is closer than any other embedding. **Mean Rank** is the average rank of targets with respect to translations. The ideal translator F^* achieves mean similarity of 1.0, top-1 accuracy of 1.0, and mean rank of 1.0. Recall that a random alignment corresponds to a mean rank of $\frac{n}{2}$. Formally,

$$\cos(u_i, v_i) = \frac{1}{n}\sum_{i=1}^{n}\big[1 - \cos\big(F(u_i), v_i\big)\big] \quad \text{Top-1}(r) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{r_i = 1\} \quad \text{Rank(r)} = \frac{1}{n}\sum_{i=1}^{n} r_i$$

`vec2vec` is the first unsupervised embedding translator, thus there is no direct baseline. As our **Naïve** baseline, we simply use $F(x) = x$ to measure geometric similarity between embedding spaces. The second (pseudo)baseline is **Oracle-aided optimal transport**. It assumes that candidate targets are known and is thus strictly easier than `vec2vec` and the Naïve baseline. We solve optimal assignment, $\pi^* = \arg\min_\pi \sum_{i=1}^{n} \cos(u_i, v_{\pi(i)})$, via either the Hungarian, Earth Mover's Distance, Sinkhorn, or (Entropic) Gromov-Wasserstein algorithms, choosing the solver with the lowest rank for each experiment. See Appendix B for more details.

4.3 Evaluating information extraction

We measure whether translation preserves semantics via *attribute inference*: for each translated embedding $F(M_1(d_i))$, our goal is to infer attributes $c_i \subseteq \mathcal{C}$ of d_i.

The first method we use is **zero-shot embedding attribute inference**: calculate pairwise cosine similarities between $F(M_1(d_i))$ and the embeddings of all attributes in $\mathcal{C}$, identify top k closest attributes, and measure whether they are correct via *top-k accuracy*: $\frac{1}{n}\sum_{i=0}^{n}\mathbf{1}\left\{|c_i^k \cap c_i| \geq 1\right\}$.

The second method is **embedding inversion** that recovers text inputs from embeddings. Since [43] requires a pre-trained inversion model for each embedding space, we use [67] instead to generate an approximation d_i' of d_i from $F(M_1(d_i))$ in a zero-shot manner. We measure the extracted information using *LLM judge accuracy*: the fraction of translated embeddings for which GPT-4o determines that d' reveals information in d. See Appendix H for our prompt.

In addition to the Naïve baseline, we also consider an **Oracle attribute inference**: zero-shot classification with the correct embedding $M_2(d)$ and class labels $M_2(\mathcal{C})$.

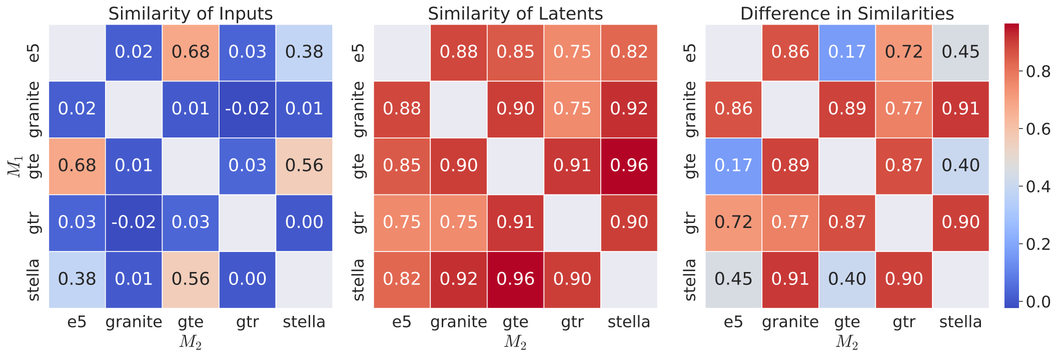

Figure 4: Pairwise cosine similarities of input embeddings (left) and their `vec2vec` latents (middle) across different embedding pairs. The absolute difference between the heatmaps plots is on the right. All numbers are computed on the same batch of 1024 NQ texts.

5 `vec2vec` learns to translate embeddings without any paired data

We first show that `vec2vec` learns a universal latent space, then demonstrate that this space preserves the geometry of all embeddings. Therefore, we can use it like a **universal language of text encoders** to translate their representations without any paired data.

`vec2vec` learns a universal latent space. `vec2vec` projects embeddings $M_{1,2,...}$ into a shared latent space via compositions of input adapters ($A_{1,2,...}$) and a shared translator T. Figure 4 shows that even when the embeddings $u_i = M_1(d_i)$ and $v_i = M_2(d_i)$ are far apart (*i.e.,* have low cosine similarity), their representations in `vec2vec`'s latent space are incredibly close: $T(A_1(u_i)) \approx T(A_2(v_i))$. Figure 1 visualizes this (via two-dimensional projections) for `vec2vec` trained on GTE and GTR embeddings: the embeddings are far apart, but their latents are *nearly overlapping*.

		`vec2vec`			Naïve Baseline			OT Baseline		
M_1	M_2	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
gra.	gtr	**0.80 (0.0)**	**0.99**	**1.19 (0.1)**	-0.03 (0.0)	0.00	4168.73 (9.2)	0.70 (0.0)	0.00	2773.72 (8.6)‡
	gte	**0.87 (0.0)**	**0.95**	**1.18 (0.0)**	0.01 (0.0)	0.00	4088.58 (9.2)	0.85 (0.0)	0.00	2680.02 (8.6)‡
	ste.	**0.79 (0.0)**	**0.98**	**1.05 (0.0)**	0.01 (0.0)	0.00	4208.26 (9.2)	0.67 (0.0)	0.00	3446.52 (8.8)‡
	e5	**0.85 (0.0)**	**0.98**	**1.11 (0.0)**	0.02 (0.0)	0.00	4111.60 (9.2)	0.83 (0.0)	0.00	3569.59 (8.7)‡
gtr	gra.	**0.81 (0.0)**	**0.99**	**1.02 (0.0)**	-0.03 (0.0)	0.00	4169.76 (9.2)	0.70 (0.0)	0.00	2775.17 (8.6)‡
	gte	**0.87 (0.0)**	**0.93**	**2.31 (0.1)**	0.04 (0.0)	0.00	4080.92 (9.2)	0.85 (0.0)	0.00	3070.69 (8.9)‡
	ste.	**0.80 (0.0)**	**0.99**	**1.03 (0.0)**	0.00 (0.0)	0.00	4198.78 (9.2)	0.67 (0.0)	0.00	3559.06 (9.1)‡
	e5	**0.83 (0.0)**	**0.84**	**2.88 (0.2)**	0.03 (0.0)	0.00	4082.84 (9.2)	**0.83 (0.0)**	0.00	3888.01 (8.9)‡
gte	gra.	**0.75 (0.0)**	**0.95**	**1.22 (0.0)**	0.01 (0.0)	0.00	4079.81 (9.3)	0.69 (0.0)	0.00	2664.38 (8.6)‡
	gtr	**0.75 (0.0)**	**0.91**	**2.64 (0.1)**	0.04 (0.0)	0.00	4084.15 (9.2)	0.70 (0.0)	0.00	3064.16 (8.9)‡
	ste.	**0.89 (0.0)**	**1.00**	**1.00 (0.0)**	0.56 (0.0)	**1.00**	**1.00 (0.0)**	0.71 (0.0)	**1.00**	**1.00 (0.0)**†
	e5	**0.87 (0.0)**	0.99	5.19 (0.5)	0.68 (0.0)	**1.00**	**1.00 (0.0)**	0.84 (0.0)	**1.00**	**1.00 (0.0)**†
ste.	gra.	**0.80 (0.0)**	**0.98**	**1.08 (0.0)**	0.01 (0.0)	0.00	4209.08 (9.3)	0.69 (0.0)	0.00	3419.44 (8.8)‡
	gtr	**0.82 (0.0)**	**1.00**	**1.10 (0.0)**	0.00 (0.0)	0.00	4192.31 (9.2)	0.70 (0.0)	0.00	3555.64 (9.0)‡
	gte	**0.92 (0.0)**	**1.00**	**1.00 (0.0)**	0.56 (0.0)	**1.00**	**1.00 (0.0)**	0.87 (0.0)	**1.00**	**1.00 (0.0)**†
	e5	**0.86 (0.0)**	**1.00**	**1.00 (0.0)**	0.38 (0.0)	0.99	1.03 (0.0)	0.83 (0.0)	**1.00**	**1.00 (0.0)**†
e5	gra.	**0.81 (0.0)**	**0.99**	**2.20 (0.2)**	0.02 (0.0)	0.00	4120.60 (9.3)	0.69 (0.0)	0.00	3526.02 (8.7)‡
	gtr	**0.74 (0.0)**	**0.82**	**2.56 (0.0)**	0.03 (0.0)	0.00	4080.76 (9.3)	0.70 (0.0)	0.00	3877.03 (8.8)‡
	gte	**0.90 (0.0)**	**1.00**	1.01 (0.0)	0.68 (0.0)	**1.00**	**1.00 (0.0)**	0.86 (0.0)	**1.00**	**1.00 (0.0)**†
	ste.	**0.78 (0.0)**	**1.00**	**1.00 (0.0)**	0.38 (0.0)	**1.00**	**1.00 (0.0)**	0.69 (0.0)	**1.00**	**1.00 (0.0)**†

Table 2: In-distribution translations: `vec2vec`s trained on NQ and evaluated on a 65536 text subset of NQ (chunked in batches of size 8192). The rank metric varies from 1 to 8192, thus 4096 corresponds to a random ordering. Standard errors are shown in parentheses. Bold denotes best value. Symbols denote the lowest-rank solver for specific experiments: Sinkhorn† and Gromov-Wasserstein‡.

		TweetTopic			MIMIC		
M_1	M_2	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
gran.	gtr	0.74 (0.0)	0.99	1.09 (0.1)	0.74 (0.0)	0.60	23.38 (1.6)
	gte	0.85 (0.0)	0.95	1.26 (0.1)	0.85 (0.0)	0.08	346.21 (7.8)
	stel.	0.77 (0.0)	0.96	1.11 (0.0)	0.72 (0.0)	0.13	242.23 (6.1)
	e5	0.83 (0.0)	0.87	3.10 (0.7)	0.84 (0.0)	0.12	361.06 (8.7)
gtr	gran.	0.79 (0.0)	0.98	2.41 (0.6)	0.78 (0.0)	0.51	35.27 (1.9)
	gte	0.85 (0.0)	0.96	1.29 (0.2)	0.84 (0.0)	0.12	279.56 (6.9)
	stel.	0.77 (0.0)	0.96	1.10 (0.0)	0.72 (0.0)	0.27	127.92 (4.4)
	e5	0.80 (0.0)	0.53	13.38 (1.2)	0.82 (0.0)	0.01	1413.80 (18.3)
gte	gran.	0.73 (0.0)	0.94	1.33 (0.1)	0.73 (0.0)	0.09	342.15 (7.8)
	gtr	0.71 (0.0)	0.95	1.29 (0.1)	0.69 (0.0)	0.12	256.63 (6.4)
	stel.	0.86 (0.0)	1.00	1.00 (0.0)	0.85 (0.0)	1.00	1.00 (0.0)
	e5	0.83 (0.0)	0.91	1.57 (0.2)	0.86 (0.0)	0.54	17.71 (0.9)
stel.	gran.	0.79 (0.0)	0.99	1.09 (0.1)	0.77 (0.0)	0.14	221.95 (5.9)
	gtr	0.77 (0.0)	1.00	1.00 (0.0)	0.75 (0.0)	0.56	17.70 (1.0)
	gte	0.90 (0.0)	1.00	1.00 (0.0)	0.91 (0.0)	1.00	1.00 (0.0)
	e5	0.85 (0.0)	0.98	1.05 (0.0)	0.85 (0.0)	0.51	26.33 (1.2)
e5	gran.	0.79 (0.0)	0.98	1.08 (0.0)	0.78 (0.0)	0.21	151.09 (4.6)
	gtr	0.67 (0.0)	0.80	3.10 (0.6)	0.66 (0.0)	0.01	1029.64 (14.9)
	gte	0.87 (0.0)	0.99	1.02 (0.0)	0.87 (0.0)	0.60	32.59 (2.6)
	stel.	0.75 (0.0)	0.98	1.06 (0.0)	0.75 (0.0)	0.46	32.12 (1.4)

Table 3: Out-of-distribution translations: `vec2vec`s trained on NQ and evaluated on the entire TweetTopic test set (800 tweets) and an 8192-record subset of MIMIC. The rank metric varies from 1 to 800 (for TweetTopic) and 8192 (for MIMIC), thus 400 and, respectively, 4096 correspond to a random ordering. Standard errors are shown in parentheses.

`vec2vec` **translations mirror target geometry.** Table 2 shows that `vec2vec` generates embeddings with near-optimal assignment across model pairs, achieving cosine similarity scores up to 0.92, top-1 accuracies up to 100%, and ranks as low as 1. In same-backbone pairings (e.g., (gte, e5)), `vec2vec`'s top-1 accuracy and rank are comparable to both the naïve baseline and (surprisingly) the oracle-aided optimal transport. Although the embeddings generated by `vec2vec` are significantly closer to the ground truth than the naïve baseline, in same-backbone pairings the embeddings are close enough to be compatible. In cross-backbone pairings, `vec2vec` is far superior on all metrics, while baseline methods perform similarly to random guessing.

Table 3 shows that this performance extends to out-of-distribution data. Our `vec2vec` translators were trained on NQ (drawn from Wikipedia), yet exhibit high cosine similarity, high accuracy, and low rank when evaluated on tweets (which are far more colloquial and use emojis) and medical records (which contain domain-specific jargon unlikely to appear in NQ). In Appendix F, we show that baseline methods fail on cross-backbone embedding pairs.

		`vec2vec`			OT Baseline		
M_1	M_2	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
gra.	clip	**0.78 (0.0)**	**0.35**	**226.62 (3.2)**	0.76 (0.0)	0.00	4073.58 (9.4)‡
gtr	clip	**0.73 (0.0)**	**0.13**	**711.23 (5.9)**	0.59 (0.0)	0.00	4096.78 (9.2)‡
gte	clip	0.62 (0.0)	**0.00**	**3233.41 (9.8)**	**0.76 (0.0)**	0.00	4026.96 (9.4)‡
ste.	clip	**0.77 (0.0)**	**0.31**	**286.69 (3.6)**	0.76 (0.0)	0.00	3955.71 (8.9)‡
e5	clip	0.64 (0.0)	**0.01**	**2568.21 (9.4)**	**0.77 (0.0)**	0.00	3771.52 (9.1)‡
clip	gra.	**0.74 (0.0)**	**0.72**	**4.46 (0.1)**	0.69 (0.0)	0.00	4053.11 (9.4)‡
clip	gtr	**0.67 (0.0)**	**0.27**	**155.11 (2.1)**	0.49 (0.0)	0.00	4096.35 (9.2)‡
clip	gte	0.75 (0.0)	**0.00**	**2678.90 (8.9)**	**0.85 (0.0)**	0.00	4025.81 (9.3)‡
clip	ste.	**0.72 (0.0)**	**0.61**	**22.50 (0.5)**	0.67 (0.0)	0.00	3951.73 (8.9)‡
clip	e5	0.73 (0.0)	**0.01**	**1692.28 (8.2)**	**0.83 (0.0)**	0.00	3771.38 (9.0)‡

Table 4: Translations between unimodal and multimodal (CLIP) embeddings: `vec2vec`s trained on NQ and evaluated on a 65536 text subset of NQ (chunked in batches of size 8192). Rank varies from 1 to 8192, thus 4096 corresponds to a random ordering. Since the embedding dimensionalities are different, only the Gromov-Wasserstein‡ OT baseline is run and the naive baseline does not apply. Bold denotes best value.

M_1	M_2	TweetTopic ($k = 1$) vec2vec	Naïve	M_1	M_2	MIMIC ($k = 10$) vec2vec	Naïve	M_1	M_2
gran.	gtr	0.25	0.10	0.30	0.24	0.19	0.11	0.76	0.88
	gte	0.32	0.09	0.30	0.34	0.36	0.13	0.76	1.00
	stel.	0.24	0.10	0.30	0.28	0.27	0.04	0.76	0.96
	e5	0.31	0.18	0.30	0.31	0.19	0.20	0.76	0.97
gtr	gran.	0.34	0.08	0.24	0.30	0.16	0.12	0.88	0.76
	gte	0.33	0.13	0.24	0.34	0.28	0.05	0.88	1.00
	stel.	0.30	0.10	0.24	0.28	0.25	0.07	0.88	0.96
	e5	0.30	0.04	0.24	0.31	0.09	0.09	0.88	0.97
gte	gran.	0.37	0.04	0.34	0.30	0.18	0.11	1.00	0.76
	gtr	0.24	0.13	0.34	0.24	0.10	0.03	1.00	0.88
	stel.	0.31	0.20	0.34	0.28	0.68	0.83	1.00	0.96
	e5	0.37	0.30	0.34	0.31	0.37	0.63	1.00	0.97
stel.	gran.	0.35	0.07	0.28	0.30	0.23	0.09	0.96	0.76
	gtr	0.26	0.13	0.28	0.24	0.22	0.09	0.96	0.88
	gte	0.38	0.36	0.28	0.34	0.90	0.98	0.96	1.00
	e5	0.35	0.34	0.28	0.31	0.38	0.46	0.96	0.97
e5	gran.	0.33	0.15	0.31	0.30	0.14	0.07	0.97	0.76
	gtr	0.26	0.22	0.31	0.24	0.11	0.04	0.97	0.88
	gte	0.34	0.28	0.31	0.34	0.47	0.66	0.97	1.00
	stel.	0.26	0.16	0.31	0.28	0.36	0.40	0.97	0.96

Table 5: Information leakage via top-k zero-shot attribute inference: vec2vecs trained on NQ and evaluated on the TweetTopic test set (800 tweets) and an 8192-record subset of MIMIC. M_1 and M_2 represent *ideal* zero-shot inference: attributes and embeddings are encoded using the same model.

Finally, Table 4 shows that vec2vec can even translate to and from the space of CLIP, a multimodal embedding model which was trained in part on *image* data. While the translations are not as strong as in Table 2, vec2vec consistently outperforms the optimal transport baseline. These results show the promise of our method at adapting to new modalities: in particular, the embedding space of CLIP has been successfully connected to other modalities such as heatmaps, audio, and depth charts [12].

6 Using vec2vec translations to extract information

In this section, we show that vec2vec translations not only preserve the geometric structure of embeddings but also retain sufficient semantics to enable attribute inference.

Zero-shot attribute inference. Table 5 shows that attribute inference on vec2vec translations consistently outperforms the naïve baseline and often does better than the ideal zero-shot baseline which performs inference on ground-truth document and attribute embeddings in the same space (this baseline is imaginary since these embeddings are not available in our setting).

vec2vec translations even work for embeddings of medical records, which are much further from the training distribution than tweets. The attributes in this case are MedCAT disease descriptions, very few of which occur in the training data. Attribute inference on translated embeddings is comparable to the naïve baseline in same-backbone pairings and outperforms it (often greatly) in cross-backbone pairings. The fact that vec2vec preserves the semantics of concepts like "alveolar periostitis" (which never appears in its training data) is evidence that its latent space is indeed a universal representation.

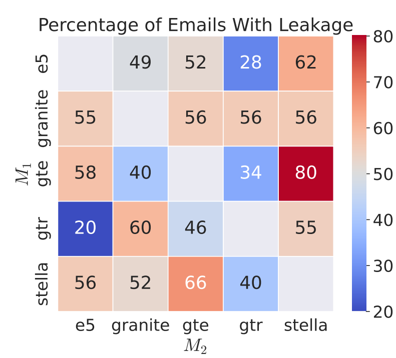

Figure 5: Leakage of information via inversion. Trained on NQ and evaluated on a 50-email subset of the Enron Email Corpus. Cells denote judge accuracy.

Zero-shot inversion. Inversion, i.e., reconstruction of text inputs, is more ambitious than attribute inference. vec2vec translations retain enough semantic information that off-the-shelf, zero-shot inversion methods like [67], developed for embeddings computed by standard encoders, extract

Ground Truth: “Subject: Enron Bashing on Frontline \n Body:..."

Generation: “Some emails discussing NROn Employee/s Complaint To thePublic ..."

Ground Truth: “Subject: Trades for 3/1/02 \n Body: \n John, \n The following trades..."

Generation: “... future transactions may await John G..."

Ground Truth: “ The following expense report is ready for approval..."

Generation: “ The upcoming expense statement from YYYY MM Dec..."

Figure 6: Examples of Enron Email Corpus inversions that infer entities and content.

information for as many as 80% of emails and 67% of tweets given *only* their translated embeddings, for some model pairs (Figure 5 and Appendix G). These inversions are imperfect and we leave development of specialized inverters for translated embeddings to future work. Nevertheless, as exemplified in Figure 6, they still extract potentially sensitive information such as individual and company names, dates, promotions, financial information, outages, and even lunch orders. In Appendix H, we show the prompt we use to measure extraction.

7 Ablations

Method	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
`vec2vec`	0.75 (0.0)	0.91	2.64 (0.1)
Naïve Baseline	0.04 (0.0)	0.00	4084.15 (9.2)
OT Baseline	0.70 (0.0)	0.00	3064.16 (8.9)
– VSP loss	0.58 (0.0)	0.00	4196.64 (9.2)
– CC loss	0.50 (0.0)	0.00	3941.36 (9.3)
– latent GAN	0.49 (0.0)	0.00	3897.09 (9.5)
– VSP *and* CC loss	0.47 (0.0)	0.00	3365.24 (9.3)
– hyperparam. tuning	0.50 (0.0)	0.00	4011.73 (9.3)

Table 6: gte $\rightarrow$ gtr translators trained without individual components of our method on NQ and evaluated on a 65536-text subset of NQ (chunked in batches of 8192). The rank metric varies from 1 to 8192, thus 4096 corresponds to a random ordering. Standard errors are shown in parentheses.

Each component of our method is important. We ablate our method subtractively, measuring the key metrics after removing individual components of our algorithm (described in Section 3). Table 6 shows that each component appears to be *critical* to building good translations. While `vec2vec`’s $\cos(\cdot)$ is higher than the naïve baseline, it performs worse across the board than the OT baseline and does not preserve the geometry of the vector space.

N	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
1000000	0.75 (0.0)	0.92	2.73 (0.2)
10000	0.57 (0.0)	0.01	1462.21 (20.)
50000	0.74 (0.0)	0.81	3.91 (0.6)
100000	0.74 (0.0)	0.85	4.52 (0.4)
500000	0.75 (0.0)	0.92	2.73 (0.2)

Table 7: gte $\rightarrow$ gtr translators trained with different amounts of GTE data: `vec2vec` models trained on NQ and evaluated an 8192-record subset of NQ. The rank metric varies from 1 to 8192, thus 4096 corresponds to a random ordering. Standard errors are shown in parentheses.

`vec2vec`s can be trained with significantly less data. In Sections 5 and 6, we use 1M-point subsets of NQ to train our `vec2vec` models. Now, we train the gte $\rightarrow$ gtr `vec2vec` with 1M GTR embeddings but fewer GTE embeddings. Table 7 shows that with as few as 10K embeddings, the translators still learn something (i.e. are better than random). Translations trained on 50K embeddings are almost as good as those trained on 1M. Translations generally improve with more training data.

8 Related work

Representation alignment. Similarities between representations of different neural networks are investigated in [26, 31, 59, 5, 18, 61, 30]. Methods based on CCA [42], SVCCA, [51], CKA [23, 38], ICA [63], time-series [39], and GUIs [16] have been used to compare embeddings from different subspaces. [37, 45, 40, 57, 48] harness representation similarity for zero-shot stitching, substitution, domain transfer, and multimodal adaptation. All rely on some amount of paired data, which is difficult to reduce [6]. Our method does not just measure similarity, we learn how to *translate* representations across spaces without any paired data.

Optimal transport. The problem of unsupervised optimal transport has been studied for image style transfer [17, 36, 70], word translation [62, 10, 15, 9, 20], and natural language sequence translation [52, 27, 1, 4, 64, 3]. Our method builds on these works, which often employ a combination of cycle-consistency and adversarial loss. Importantly, unlike prior word and sequence translation methods, multiple representations of the same input (e.g., heavily overlapping word vocabularies) are unavailable in our setting. [54] proposes a solver for matching small sets of embeddings between different vision-language models. Our method goes well beyond matching by taking unknown embeddings and *generating* matching embeddings in the space of another model.

Embedding inversion. An emerging line of research investigates decoding text from language model embeddings [55, 29, 43] and outputs [44, 7, 66]. `vec2vec` helps apply these to unknown embeddings, without an encoder or paired data, by translating them to the space of a known model.

Bridging modality gaps. Previous work has noted an inherent "gap" between image- and text-based models [33] and proposed various ways to unify the modalities [56]. Some approaches feed image embeddings directly into language models [22, 60, 11, 35], while others generate captions from image embeddings [41] or even from text embeddings themselves [43]. [12] introduces a shared embedding space that integrates inputs from multiple modalities, including text, audio, and vision. In contrast, our post-hoc approach directly translates between representations and complements these systems by enabling inputs from a wide variety of embedding models.

9 Discussion and Future Work

The Platonic Representation Hypothesis conjectures that the representation spaces of modern neural networks are converging. We assert the Strong Platonic Representation Hypothesis: the latent universal representation can be learned and harnessed to translate between representation spaces without any encoders or paired data.

In Section 5, we demonstrated that our `vec2vec` method successfully translates embeddings generated from unseen documents by unseen encoders, and the translator is robust to (sometimes very) out-of-distribution inputs. This suggests that `vec2vec` learns domain-agnostic translations based on the universal geometric relationships which encode the same semantics in multiple embedding spaces.

In Section 6, we showed that `vec2vec` translations preserve sufficient input semantics to enable attribute inference. We extracted sensitive disease information from patient records and partial content from corporate emails, with access only to document embeddings and no access to the encoder that produced them. Better translation methods will enable higher-fidelity extraction, confirming once again that embeddings reveal (almost) as much as their inputs.

Our findings provide compelling evidence for the Strong Platonic Representation Hypothesis for text-based models. Our preliminary results on CLIP suggest that the universal geometry can be harnessed in other modalities, too. The results in this paper are but a *lower bound* on inter-representation translation. Better and more stable learning algorithms, architectures, and other methodological improvements will support scaling to more data, more model families, and more modalities.

Acknowledgments and Disclosure of Funding

This research is supported in part by the Google Cyber NYC Institutional Research Program. RJ is supported by the Digital Life Initiative Fellowship and JM by the National Science Foundation.

References

[1] David Alvarez-Melis and Tommi S. Jaakkola. "Gromov-Wasserstein Alignment of Word Embedding Spaces". 2018. arXiv: 1809.00013 [cs.CL].

[2] Dimosthenis Antypas, Asahi Ushio, Francesco Barbieri, and Jose Camacho-Collados. "Multilingual Topic Classification in X: Dataset and Analysis". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 20136–20152.

[3] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 789–798.

[4] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. "Unsupervised Neural Machine Translation". 2018. arXiv: 1710.11041 [cs.CL].

[5] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. "Revisiting Model Stitching to Compare Neural Representations". 2021. arXiv: 2106.07682 [cs.LG].

[6] Irene Cannistraci, Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, and Emanuele Rodolà. "Bootstrapping Parallel Anchors for Relative Representations". 2023. arXiv: 2303.00721 [cs.LG].

[7] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Itay Yona, Eric Wallace, David Rolnick, and Florian Tramèr. "Stealing Part of a Production Language Model". 2024. arXiv: 2403.06634 [cs.CR].

[8] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. "Graph Optimal Transport for Cross-Domain Alignment". 2020. arXiv: 2006.14744 [cs.CL].

[9] Xilun Chen and Claire Cardie. "Unsupervised Multilingual Word Embeddings". 2018. arXiv: 1808.08933 [cs.CL].

[10] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. "Word Translation Without Parallel Data". 2018. arXiv: 1710.04087 [cs.CL].

[11] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. "DreamLLM: Synergistic Multimodal Comprehension and Creation". 2024. arXiv: 2309.11499 [cs.CV].

[12] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. "ImageBind: One Embedding Space To Bind Them All". 2023. arXiv: 2305.05665 [cs.CV].

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks". In: *Commun. ACM* 63.11 (Oct. 2020), pp. 139–144.

[14] IBM Granite Embedding Team. "Granite Embedding Models". Dec. 2024.

[15] Edouard Grave, Armand Joulin, and Quentin Berthet. "Unsupervised Alignment of Embeddings with Wasserstein Procrustes". 2018. arXiv: 1805.11222 [cs.LG].

[16] Florian Heimerl, Christoph Kralj, Torsten Moller, and Michael Gleicher. "embComp: Visual Interactive Comparison of Vector Embeddings". In: *IEEE Transactions on Visualization and Computer Graphics* 28.8 (Aug. 2022), pp. 2953–2969.

[17] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. "Multimodal Unsupervised Image-to-Image Translation". 2018. arXiv: 1804.04732 [cs.CV].

[18] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. "The Platonic Representation Hypothesis". 2024. arXiv: 2405.07987 [cs.LG].

[19] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. “MIMIC-III, a freely accessible critical care database”. In: *Scientific data* 3.1 (2016), pp. 1–9.

[20] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. “Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 2979–2984.

[21] Bryan Klimt and Yiming Yang. “The enron corpus: a new dataset for email classification research”. In: *Proceedings of the 15th European Conference on Machine Learning*. ECML’04. Pisa, Italy: Springer-Verlag, 2004, pp. 217–226.

[22] Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. “Grounding language models to images for multimodal inputs and outputs”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 17283–17300.

[23] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. “Similarity of Neural Network Representations Revisited”. 2019. arXiv: `1905.00414 [cs.LG]`.

[24] Zeljko Kraljevic, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A Folarin, Angus Roberts, et al. “Multi-domain clinical natural language processing with MedCAT: the medical concept annotation toolkit”. In: *Artificial intelligence in medicine* 117 (2021), p. 102083.

[25] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. “Natural Questions: A Benchmark for Question Answering Research”. In: *Transactions of the Association for Computational Linguistics* 7 (2019). Ed. by Lillian Lee, Mark Johnson, Brian Roark, and Ani Nenkova, pp. 452–466.

[26] Aarre Laakso and Garrison Cottrell. “Content and cluster analysis: Assessing representational similarity in neural systems”. In: *Philosophical Psychology* 13.1 (2000), pp. 47–76.

[27] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. “Unsupervised Machine Translation Using Monolingual Corpora Only”. 2018. arXiv: `1711.00043 [cs.CL]`.

[28] Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. “Does BERT Pretrained on Clinical Notes Reveal Sensitive Data?” In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, June 2021, pp. 946–959.

[29] Haoran Li, Mingshi Xu, and Yangqiu Song. “Sentence Embedding Leaks More Information than You Expect: Generative Embedding Inversion Attack to Recover the Whole Sentence”. 2023. arXiv: `2305.03010 [cs.CL]`.

[30] Jiaang Li, Yova Kementchedjhieva, Constanza Fierro, and Anders Søgaard. “Do Vision and Language Models Share Concepts? A Vector Space Alignment Study”. In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 1232–1249.

[31] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. “Convergent Learning: Do different neural networks learn the same representations?” 2016. arXiv: `1511.07543 [cs.LG]`.

[32] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. “Towards General Text Embeddings with Multi-stage Contrastive Learning”. 2023. arXiv: `2308.03281 [cs.CL]`.

[33] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. “Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning”. 2022. arXiv: `2203.02053 [cs.CL]`.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. “Visual Instruction Tuning”. 2023. arXiv: `2304.08485 [cs.CV]`.

[36] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. "Unsupervised Image-to-Image Translation Networks". 2018. arXiv: 1703.00848 [cs.CV].

[37] Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. "Latent Space Translation via Semantic Alignment". 2024. arXiv: 2311.00664 [cs.LG].

[38] Mayug Maniparambil, Raiymbek Akshulakov, Yasser Abdelaziz Dahou Djilali, Mohamed El Amine Seddik, Sanath Narayan, Karttikeya Mangalam, and Noel E O'Connor. "Do Vision and Language Encoders Represent the World Similarly?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 14334–14343.

[39] Deven M Mistry and Ali A Minai. "A Comparative Study of Sentence Embedding Models for Assessing Semantic Variation". In: *International Conference on Artificial Neural Networks*. 2023, pp. 1–12.

[40] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. "Text-To-Concept (and Back) via Cross-Model Alignment". 2023. arXiv: 2305.06386 [cs.CV].

[41] Ron Mokady, Amir Hertz, and Amit H. Bermano. "ClipCap: CLIP Prefix for Image Captioning". 2021. arXiv: 2111.09734 [cs.CV].

[42] Ari S. Morcos, Maithra Raghu, and Samy Bengio. "Insights on representational similarity in neural networks with canonical correlation". 2018. arXiv: 1806.05759 [stat.ML].

[43] John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. "Text Embeddings Reveal (Almost) As Much As Text". 2023. arXiv: 2310.06816 [cs.CL].

[44] John X. Morris, Wenting Zhao, Justin T. Chiu, Vitaly Shmatikov, and Alexander M. Rush. "Language Model Inversion". 2023. arXiv: 2311.13647 [cs.CL].

[45] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. "Relative representations enable zero-shot latent space communication". 2023. arXiv: 2209.15430 [cs.LG].

[46] Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. "Counter-fitting Word Vectors to Linguistic Constraints". 2016. arXiv: 1603.00892 [cs.CL].

[47] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. "Large Dual Encoders Are Generalizable Retrievers". 2021. arXiv: 2112.07899 [cs.IR].

[48] Antonio Norelli, Marco Fumero, Valentino Maiorca, Luca Moschella, Emanuele Rodolà, and Francesco Locatello. "ASIF: Coupled Data Turns Unimodal Models to Multimodal Without Training". 2023. arXiv: 2210.01738 [cs.LG].

[49] Gabriel Peyré, Marco Cuturi, and Justin Solomon. "Gromov-Wasserstein Averaging of Kernel and Distance Matrices". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. Vol. 48. JMLR: Workshop and Conference Proceedings. New York, NY, USA: JMLR, 2016.

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". 2021. arXiv: 2103.00020 [cs.CV].

[51] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. "SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability". 2017. arXiv: 1706.05806 [stat.ML].

[52] Sujith Ravi and Kevin Knight. "Deciphering Foreign Language". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 12–21.

[53] Divya Saxena and Jiannong Cao. "Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions". In: *ACM Comput. Surv.* 54.3 (May 2021).

[54] Dominik Schnaus, Nikita Araslanov, and Daniel Cremers. "It's a (Blind) Match! Towards Vision-Language Correspondence without Parallel Data". 2025. arXiv: 2503.24129 [cs.CV].

[55] Congzheng Song and Ananth Raghunathan. "Information Leakage in Embedding Models". 2020. arXiv: 2004.00053 [cs.LG].

[56] Shezheng Song, Xiaopeng Li, Shasha Li, Shan Zhao, Jie Yu, Jun Ma, Xiaoguang Mao, and Weimin Zhang. "How to Bridge the Gap between Modalities: A Comprehensive Survey on Multimodal Large Language Model". 2023. arXiv: 2311.07594 [cs.CL].

[57] Yingtao Tian and Jesse Engel. "Latent translation: Crossing modalities by bridging generative models". In: *arXiv preprint arXiv:1902.08261* (2019).

[58] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. "Text Embeddings by Weakly-Supervised Contrastive Pre-training". 2024. arXiv: 2212.03533 [cs.CL].

[59] Liwei Wang, Lunjia Hu, Jiayuan Gu, Yue Wu, Zhiqiang Hu, Kun He, and John Hopcroft. "Towards Understanding Learning Representations: To What Extent Do Different Neural Networks Learn the Same Representation". 2018. arXiv: 1810.11750 [cs.LG].

[60] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. "VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks". 2023. arXiv: 2305.11175 [cs.CV].

[61] Christopher Wolfram and Aaron Schein. "Layers at similar depths generate similar activations across llm architectures". In: *arXiv preprint arXiv:2504.08775* (2025).

[62] Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. "Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Rada Mihalcea, Joyce Chai, and Anoop Sarkar. Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 1006–1011.

[63] Hiroaki Yamagiwa, Momose Oyama, and Hidetoshi Shimodaira. "Discovering Universal Geometry in Embeddings with ICA". 2023. arXiv: 2305.13175 [cs.CL].

[64] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. "Unsupervised Neural Machine Translation with Weight Sharing". 2018. arXiv: 1804.09057 [cs.CL].

[65] Jinsung Yoon and Sercan O Arik. "Embedding-Converter: A Unified Framework for Cross-Model Embedding Transformation". 2025.

[66] Collin Zhang, John X. Morris, and Vitaly Shmatikov. "Extracting Prompts by Inverting LLM Outputs". 2024. arXiv: 2405.15012 [cs.CL].

[67] Collin Zhang, John X. Morris, and Vitaly Shmatikov. "Universal Zero-shot Embedding Inversion". 2025. arXiv: 2504.00147 [cs.CL].

[68] Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. "Jasper and Stella: distillation of SOTA embedding models". 2025. arXiv: 2412.19048 [cs.IR].

[69] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. "Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models". In: *arXiv preprint arXiv:2506.05176* (2025).

[70] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". 2020. arXiv: 1703.10593 [cs.CV].

A Compute

Our training and evaluation were conducted using diverse compute environments, including both local and cloud GPU clusters. Experiments were done on NVIDIA 2080Ti, L4, A40, and A100 GPUs, listed in order of increasing computational capacity.

For our final results, we trained 25 `vec2vec` models fully and 30 models partially (see Appendix E). The full models' training durations usually ranged from 1 to 7 days, depending on the specific GPU and model pair (which affected convergence rates). Partial convergence was stopped after 2 days. Due to the size of Qwen, our (qwen, gte) ablation was trained for 20 days on an A100. Taking a conservative estimate of the average training time, this amounted to approximately 176 GPU days (24 models $\times$ 4 days / model + 30 models $\times$ 2 days / model + 1 (qwen, gte) $\times$ 20 days / model).

Evaluation procedures varied by model type:

- The 10 main `vec2vec` models required $\sim$1 hour each for NQ, TweetTopic, and MIMIC evaluation (across GPU types), plus 30 minutes for attribute extraction on TweetTopic and MIMIC, and 1.5 hours for inversion and downstream LLM evaluation on Enron and TweetTopic. Naive baselines required $\sim$30 minutes each across all datasets.
- The 15 additional fully-trained models required 30 minutes each for NQ evaluation, with an extra 30 minutes for MS COCO evaluation of (clip, granite).
- Optimal transport baselines ran on CPU only, requiring $\sim$1 hour per dataset (three datasets for main models, one for others).

In total, our experiments consumed almost 176 GPU days for training and an additional 42 GPU hours for evaluation and analysis. An additional 45 CPU hours were required for optimal transport.

B Oracle-aided optimal transport baseline

Let $u_i = M_1(d_i)$ and $v_i = M_2(d_i)$ denote embeddings of the same document d_i from two different embedding models. In Section 5, we solve the optimal assignment problem:

$$\pi^* = \arg\min_{\pi} \sum_{i=1}^{n} \cos(u_i, v_{\pi(i)}),$$

using four algorithms: Hungarian (linear sum assignment), Earth Mover's Distance (EMD), Sinkhorn, Gromov-Wasserstein. For the Gromov-Wasserstein algorithm, we try both the entropic and non-entropic variants with multiple hyperparameter configurations and select the best figure. Note that the optimal transport (OT) baseline computes matchings and transports between embeddings derived from the *same underlying texts*, strongly favoring OT methods. Nevertheless, OT still struggles when embeddings originate from different model backbones.

Since the Hungarian algorithm produces a discrete matching, it is evaluated only using Top-1 Accuracy, while the other algorithms are evaluated across all metrics. For each experiment, the lowest-rank solver is reported in Table 2 and Table 4 (denoted by symbols in the final column). Evaluation metrics are defined as follows:

1. **Top-1 Accuracy**: Fraction of embeddings correctly identified as closest pairs, calculated by either selecting the maximum transported mass per embedding or applying the Hungarian algorithm directly to the transport plan P. We report the higher accuracy between the two.
2. **Mean Rank**: Average rank position of the correct embedding match v_i when sorted by descending transported mass P_{ij} from u_i:

$$\text{rank}(v_i) = \text{position of } v_i \text{ among sorted } P_{ij}.$$

3. **Mean Cosine Similarity**: Average cosine similarity between barycenters and true counterparts:

$$v_i' = \frac{\sum_{j=1}^{n} P_{ij} v_j}{\sum_{j=1}^{n} P_{ij}}, \quad \text{Similarity} = \frac{1}{n}\sum_{i=1}^{n} \cos(v_i', v_i).$$

C Translating to and from Qwen

		vec2vec			OT Baseline		
M_1	M_2	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
gte	qwen	**0.50 (0.0)**	**0.92**	**2.28 (0.2)**	0.38 (0.0)	0.00	425.28 (1.1)‡
qwen	gte	0.84 (0.0)	**0.88**	**2.49 (0.3)**	**0.85 (0.0)**	0.00	425.07 (1.2)‡

Table 8: Translations between GTE and Qwen embeddings trained on NQ and evaluated on a 65536 text subset of NQ (chunked in batches of size 1024). Rank varies from 1 to 1024, thus 512 corresponds to a random ordering. Since the embedding dimensionalities are different, only the Gromov-Wasserstein‡ OT baseline is run and the naive baseline does not apply. Bold denotes best value.

As shown in Table 8, `vec2vec` successfully translates between GTE and Qwen, significantly outperforming the optimal transport baseline in all metrics except qwen $\rightarrow$ gte cosine similarity, which we hypothesize may be due to the substantial performance gap between the models—indeed, Qwen differs from GTE in architecture (dense Qwen backbone), training methodology (unsupervised + model merging techniques), size (14$\times$ larger than the next largest model and 37$\times$ larger than GTE), context length, and recency. Given Qwen's size and computational cost, we only evaluated this representative pair. We leave further evaluation to future work.

D Text-image retrieval on MS COCO

model	R@16 $\uparrow$	$\cos(\cdot)\uparrow$	Rank $\downarrow$
granite $\rightarrow$ clip	0.23	0.23 (0.0)	233.67 (3.0)
clip (baseline)	0.75	0.30 (0.0)	23.20 (0.8)

Table 9: Cross-model text-image retrieval on MS COCO: granite $\rightarrow$ clip `vec2vec` trained on NQ (unimodal) and evaluated on MS COCO's validation set. The rank metric varies from 1 to 5000, thus 2500 corresponds to a random ordering. Queries (captions) embedded with either Granite or CLIP. Documents (images) embedded with CLIP. Each caption has a unique image. Standard errors are shown in parentheses.

Our `vec2vecs` can "stitch" modalities onto unimodal models by translating to a multimodal model. To test this, we evaluated cross-modal text-image retrieval on MS COCO's validation set (5000 examples) [34], translating queries (captions) embedded with Granite to retrieve documents (images) embedded with CLIP using our unimodal granite $\rightarrow$ clip translator from section 4.3. Each caption has a unique image. We report Recall@16, cosine similarities, and Rank, with CLIP (for both documents and queries) as our baseline.

As Table 9 shows, translating Granite embeddings to CLIP enables non-negligible cross-model multimodal retrieval with **a unimodal model for queries**—despite zero multimodal training. Further evaluation of this paradigm with multimodal-specific training is a promising direction.

E Initialization robustness by model backbone

GAN training is notoriously unstable to weight initialization [53]. To measure our method's robustness, we trained fifteen e5 $\rightarrow$ gte (shared backbone) and e5 $\rightarrow$ gtr (cross-backbone) `vec2vecs` on the NQ dataset for a fixed 10 epochs.

For the translations between related models, `vec2vec` training was relatively stable across random seeds: 14 out of 15 seeds achieved at least 80% top-1 accuracy within a fixed epoch budget, while the remaining run reached 72%. In contrast, translation between unrelated models proved significantly less stable, with only 3 out of 15 runs achieving convergence (80% top-1 accuracy). We leave improving the seed stability of our training regime as future, valuable work.

F Full out-of-distribution translation results

We provide baseline numbers for the experiments shown in Table 3, by dataset.

		`vec2vec`			Naïve Baseline			OT Baseline		
E_1	E_2	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
gra.	gtr	0.74 (0.0)	0.99	1.09 (0.1)	-0.04 (0.0)	0.00	415.61 (8.2)	0.71 (0.0)	0.01	220.93 (7.1)‡
	gte	0.85 (0.0)	0.95	1.26 (0.1)	0.00 (0.0)	0.00	406.73 (8.2)	0.87 (0.0)	0.01	201.48 (6.6)‡
	ste.	0.77 (0.0)	0.96	1.11 (0.0)	0.00 (0.0)	0.00	417.27 (8.2)	0.74 (0.0)	0.00	239.36 (6.7)‡
	e5	0.83 (0.0)	0.87	3.10 (0.7)	0.02 (0.0)	0.00	405.53 (8.1)	0.87 (0.0)	0.01	244.94 (7.4)‡
gtr	gra.	0.79 (0.0)	0.98	2.41 (0.6)	-0.04 (0.0)	0.00	411.53 (8.3)	0.57 (0.0)	0.01	398.29 (8.2)‡
	gte	0.85 (0.0)	0.96	1.29 (0.2)	0.04 (0.0)	0.00	392.01 (8.2)	0.86 (0.0)	0.01	259.47 (7.4)‡
	ste.	0.77 (0.0)	0.96	1.10 (0.0)	0.00 (0.0)	0.00	394.69 (8.3)	0.74 (0.0)	0.00	294.58 (7.4)‡
	e5	0.80 (0.0)	0.53	13.38 (1.2)	0.03 (0.0)	0.00	400.85 (8.2)	0.87 (0.0)	0.01	266.04 (7.7)‡
gte	gra.	0.73 (0.0)	0.94	1.33 (0.1)	0.00 (0.0)	0.00	408.81 (8.3)	0.56 (0.0)	0.01	398.16 (8.2)*
	gtr	0.71 (0.0)	0.95	1.29 (0.1)	0.04 (0.0)	0.00	386.58 (8.3)	0.71 (0.0)	0.01	254.74 (7.3)‡
	ste.	0.86 (0.0)	1.00	1.00 (0.0)	0.58 (0.0)	1.00	1.00 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
	e5	0.83 (0.0)	0.91	1.57 (0.2)	0.68 (0.0)	1.00	1.00 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
ste.	gra.	0.79 (0.0)	0.99	1.09 (0.1)	0.00 (0.0)	0.00	418.16 (8.4)	0.57 (0.0)	0.00	399.56 (8.2)‡
	gtr	0.77 (0.0)	1.00	1.00 (0.0)	0.00 (0.0)	0.00	393.07 (8.1)	0.71 (0.0)	0.00	294.65 (7.4)‡
	gte	0.90 (0.0)	1.00	1.00 (0.0)	0.58 (0.0)	1.00	1.00 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
	e5	0.85 (0.0)	0.98	1.05 (0.0)	0.37 (0.0)	0.89	1.55 (0.1)	1.00 (0.0)	1.00	1.00 (0.0)*
e5	gra.	0.79 (0.0)	0.98	1.08 (0.0)	0.02 (0.0)	0.00	405.75 (8.3)	0.57 (0.0)	0.01	398.34 (8.2)‡
	gtr	0.67 (0.0)	0.80	3.10 (0.6)	0.03 (0.0)	0.00	401.16 (8.4)	0.71 (0.0)	0.00	268.28 (7.6)‡
	gte	0.87 (0.0)	0.99	1.02 (0.0)	0.68 (0.0)	1.00	1.00 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
	ste.	0.75 (0.0)	0.98	1.06 (0.0)	0.37 (0.0)	1.00	1.00 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*

Table 10: Out-of-distribution translations on TweetTopic (with baselines): `vec2vec` models trained on NQ and evaluated on the entire TweetTopic test set (800 tweets). The rank metric varies from 1 to 800, thus 400 corresponds to a random ordering. Standard errors are shown in parentheses. Symbols denote the lowest-rank solver: Earth Mover's Distance* and Gromov-Wasserstein‡

		`vec2vec`			Naïve Baseline			OT Baseline		
E_1	E_2	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$	$\cos(\cdot)\uparrow$	T-1 $\uparrow$	Rank $\downarrow$
gra.	gtr	0.74 (0.0)	0.60	23.38 (1.6)	-0.02 (0.0)	0.00	4010.00 (25.8)	0.82 (0.0)	0.00	3962.83 (26.1)†
	gte	0.85 (0.0)	0.08	346.21 (7.8)	0.01 (0.0)	0.00	3978.35 (26.1)	0.92 (0.0)	0.00	3808.18 (25.9)†
	ste.	0.72 (0.0)	0.13	242.23 (6.1)	-0.01 (0.0)	0.00	3900.74 (26.2)	0.86 (0.0)	0.02	3780.44 (26.0)†
	e5	0.84 (0.0)	0.12	361.06 (8.7)	0.02 (0.0)	0.00	4024.92 (26.1)	0.93 (0.0)	0.00	3937.63 (26.2)†
gtr	gra.	0.78 (0.0)	0.51	35.27 (1.9)	-0.02 (0.0)	0.00	4023.67 (26.1)	0.87 (0.0)	0.00	3964.83 (26.1)†
	gte	0.84 (0.0)	0.12	279.56 (6.9)	0.08 (0.0)	0.00	4180.47 (26.2)	0.87 (0.0)	0.00	4088.97 (26.2)‡
	ste.	0.72 (0.0)	0.27	127.92 (4.4)	0.00 (0.0)	0.00	4296.04 (26.1)	0.76 (0.0)	0.00	4095.11 (26.1)‡
	e5	0.82 (0.0)	0.01	1413.80 (18.3)	0.09 (0.0)	0.00	4064.47 (26.2)	0.93 (0.0)	0.00	4010.13 (26.1)†
gte	gra.	0.73 (0.0)	0.09	342.15 (7.8)	0.01 (0.0)	0.00	3946.19 (25.8)	0.87 (0.0)	0.00	3802.92 (25.9)†
	gtr	0.69 (0.0)	0.12	256.63 (6.4)	0.08 (0.0)	0.00	4229.90 (26.2)	0.69 (0.0)	0.00	4094.02 (26.1)‡
	ste.	0.85 (0.0)	1.00	1.00 (0.0)	0.56 (0.0)	1.00	1.00 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
	e5	0.86 (0.0)	0.54	17.71 (0.9)	0.69 (0.0)	0.98	1.04 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
ste.	gra.	0.77 (0.0)	0.14	221.95 (5.9)	-0.01 (0.0)	0.00	3951.42 (25.9)	0.87 (0.0)	0.01	3776.52 (26.0)†
	gtr	0.75 (0.0)	0.56	17.70 (1.0)	0.00 (0.0)	0.00	4339.83 (26.2)	0.70 (0.0)	0.00	4093.61 (26.1)‡
	gte	0.91 (0.0)	1.00	1.00 (0.0)	0.56 (0.0)	1.00	1.00 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
	e5	0.85 (0.0)	0.51	26.33 (1.2)	0.35 (0.0)	0.59	12.68 (0.6)	0.93 (0.0)	1.00	1.00 (0.0)†
e5	gra.	0.78 (0.0)	0.21	151.09 (4.6)	0.02 (0.0)	0.00	4008.10 (25.9)	0.87 (0.0)	0.00	3932.58 (26.2)†
	gtr	0.66 (0.0)	0.01	1029.64 (14.9)	0.09 (0.0)	0.00	4032.85 (26.2)	0.82 (0.0)	0.00	4010.06 (26.1)†
	gte	0.87 (0.0)	0.60	32.59 (2.6)	0.69 (0.0)	0.98	1.09 (0.0)	1.00 (0.0)	1.00	1.00 (0.0)*
	ste.	0.75 (0.0)	0.46	32.12 (1.4)	0.35 (0.0)	0.86	2.49 (0.1)	0.86 (0.0)	1.00	1.01 (0.0)†

Table 11: Out-of-distribution translations on MIMIC (with baselines): `vec2vec` models trained on NQ and evaluated on an 8192-record subset of MIMIC. The rank metric varies from 1 to 8192, thus 4096 corresponds to a random ordering. Standard errors are shown in parentheses. Symbols denote the lowest-rank solver: Earth Mover's Distance*, Sinkhorn† and Gromov-Wasserstein‡

G Zero-shot inversion on TweetTopic

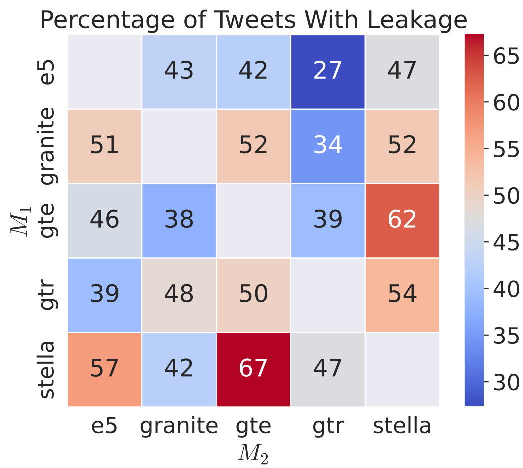

Figure 7: Leakage of information via inversion. Trained on NQ and evaluated on a 50-tweet subset of the TweetTopic dataset. Cells denote judge accuracy.

We replicate Figure 5 on the TweetTopic dataset in Figure 7, extracting information given *only* translated embeddings of tweets. We achieve non-negligible leakage for all model pairs.

H Prompt for measuring information extraction

Figures 5 and 7 utilize an LLM judge to determine whether the approximate inversion (using [67]) of a translated email embedding leaks information about the original email. Figure 8 shows our prompt to the GPT-4o judge:

> **Original email:** {ground_truth}
> **Reconstructed email:** {generation}.
> Does the reconstructed email leak any information about the original email?
> Answer with only 'yes' or 'no'.

Figure 8: The prompt given to the LLM judge.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claim is that we are (1) able to translate embeddings with high cosine similarity across model pairs with different architectures, parameter counts, and training datasets and (2) these translations can be used for attribute inference. These are the two experimental sections exactly.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The key limitation of our method is the training stability of GANs. Downstream of this limitation is long computation times and limited (while still effective) translation quality. This is discussed at length in the discussion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not show any theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Along with our releasing our code (used for each of the experiments we present), we describe in depth the loss functions and architecture choices we made. We also ablate these choices in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We link to well-documented code that will help reproduce any of our experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All details necessary to reconstruct our experiments are contained in our code. On top of that we describe our method at length in the body of our paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Wherever applicable we show error bars: the rank and cosine metrics, namely.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include compute information in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All of our experiments are on publicly-available datasets that are either fully anonymous or pseudonymized.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: On the positive side, we provide evidence for a popular hypothesis in the ML community. On the negative side, we highlight, discuss, and provide remediation ideas for an emerging threat. As with many security-flavored papers, we hope to increase visibility of emerging threats.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Although we have none planned, future releases of model weights would be of translators between known, open-source embedding spaces. These would not be very useful in exfiltrating data from unknown embeddings (a concern we highlight).

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All data we use is publicly available, cited, and used as dictated by the licenses and terms.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our public code release will include documentation on how exactly to reconstruct our experiments.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We did not use crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We did not use crowdsourcing or human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Our paper's methodology does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.