# EMPIRICAL OBSERVATIONS ON THE INSTABILITY OF ALIGNING WORD VECTOR SPACES WITH GANS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Unsupervised bilingual dictionary induction (UBDI) is useful for unsupervised machine translation and for cross-lingual transfer of models into low-resource languages. One approach to UBDI is to align word vector spaces in different languages using Generative Adversarial Networks (GANs) with linear generators, achieving state-of-the-art performance for several language pairs. For some pairs, however, GAN-based induction is unstable or completely fails to align the vector spaces. We focus on cases where linear transformations provably exist, but the performance of GAN-based UBDI depends heavily on the model initialization. We show that the instability depends on the shape and density of the vector sets, but not on noise; that it is the result of local optima, but neither over-parameterization nor changing the batch size or the learning rate consistently reduces instability. Nevertheless, we can stabilize GAN-based UBDI through best-of-N model selection, based on an unsupervised stopping criterion.

## 1 INTRODUCTION

A word vector space – also sometimes referred to as a *word embedding* – associates similar words in a vocabulary with similar vectors. Learning a projection of one word vector space into another, such that similar words – across the two word embeddings – are associated with similar vectors, is useful in many contexts, with the most prominent example being the alignment of vocabularies of different languages. This is a key step in machine translation of low-resource languages (Lample et al., 2018). An embedding of English words may associate *thoughtful*, *considerate*, and *gracious* with similar vectors, for example, but for English-Icelandic translation, it would be useful to have access to a cross-lingual word embedding space in which *hugulsamur* (lit.: 'thoughtful') was also associated with a similar vector. Such joint embeddings of words across languages can also be used to extract bilingual dictionaries.

Projections between word vector spaces have typically been learned from dictionary seeds. In seminal papers such as Mikolov et al. (2013) and Faruqui and Dyer (2014), these seeds would comprise thousands of words, but Vulić and Korhonen (2016) showed that we can learn reliable projections from as little as 50 words. Smith et al. (2017) and Hauer et al. (2017) subsequently showed that the seed can be replaced with just words that are identical across languages; and Artetxe et al. (2017) showed that numerals can also do the job, in some cases; both proposals removing the need for an actual dictionary. Even more recently, a handful of papers have proposed an entirely unsupervised approach to projecting word vector spaces onto each other, based on Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). We present the core idea behind such approaches in §3, but briefly put, GANs are used to learn a linear transformation to minimize the divergence between a target distribution (say the Icelandic embeddings) and a source distribution (the English embeddings projected into the Icelandic space).

The possibility of unsupervised bilingual dictionary induction (UBDI) has seemingly removed the data bottleneck in machine translation, evoking the idea that we can now *learn to translate without human supervision* (Lample et al., 2018). Yet, it remains an open question whether the initial, positive results extrapolate to real-world scenarios of learning translations between low-resource language pairs. Søgaard et al. (2018) recently presented results suggesting that UBDI is challenged by some language pairs exhibiting very different morphosyntactic properties, as well as when the monolingual corpora are very different. In this paper, we identify *easy*, *hard*, and *impossible* instances

of GAN-based UBDI, and apply a simple test for discriminating between them. The hard cases exhibit instability, i.e. their success depends heavily on initialization. We set up a series of experiments to investigate these hard cases.

**Our contributions** We introduce a distinction between easy, hard, and impossible alignment problems over pairs of word vector spaces and show that a simple linearity test can be used to tell these cases apart. We show that the impossible cases are caused not necessarily by linguistic differences, but rather by properties of the corpora and the embedding algorithms. We also show that in the hard cases, the likelihood of being trapped in local minima depends heavily on the shape and density of the vector sets, but not on noise. Changes in the number of parameters, batch size, and learning rate do not alleviate the instability. Yet, using an unsupervised model selection method over N different initializations to select the best generators, leads to a 6.74% average error reduction over standard MUSE.

**Structure of the paper** §2 presents MUSE (Conneau et al., 2018), an approach to GAN-based UBDI. Here we also discuss theoretical results from the GAN literature, relevant to our case, and show a relation to a common point set registration method. In §3, we use a test based on Procrustes Analysis to discriminate between easy, hard, and impossible cases, discussing its relation with tests of isomorphism and isospectrality. We then focus on the hard cases, where linear transformations provably exist, but GANs exhibit considerable instability. Through a series of experiments, we analyze what affects the instability of GAN-based UBDI. §4 presents our unsupervised best-of-N model selection method for stabilizing GAN-based UBDI.

## 2 UNSUPERVISED ALIGNMENT USING GANS

In this section, we discuss the dynamics of GAN-based UBDI and how the training behavior of GANs can help us understand their limitations as applied to UBDI. Two families of approaches to UBDI exist: using GANs (Barone, 2016; Conneau et al., 2018; Zhang et al., 2017) and using iterative closest point (Hoshen and Wolf, 2018). We focus on GAN-based UBDI, and more specifically on MUSE (Conneau et al., 2018), but at the end of this section we establish a relation between the two families of algorithms.

A GAN consists of a generator and a discriminator. The generator $G$ is trained to fool the discriminator $D$. The generator can be any differentiable function; in MUSE, it is a linear transform $\Omega$. Let $\mathbf{e} \in E$ be an English word vector, and $\mathbf{f} \in F$ a French word vector, both of dimensionality $d$. The goal of the generator is then to choose $\Omega \in \mathbb{R}^{d \times d}$ such that $\Omega E$ has a distribution close to $F$. The discriminator is a map $D_w : \mathcal{X} \rightarrow \{0, 1\}$, implemented in MUSE as a multi-layered perceptron. The objective of the discriminator is to discriminate between vector spaces $F$ and $\Omega E$. During training, the model parameters $\Omega$ and $w$ are optimized using stochastic gradient descent by alternately updating the parameters of the discriminator based on the gradient of the discriminator loss and the parameters of the generator based on the gradient of the generator loss, which, by definition, is the inverse of the discriminator loss. The loss function used in MUSE and in our experiments below is cross-entropy. In each iteration, we sample $N$ vectors $e \in E$ and $N$ vectors $f \in F$ and update the discriminator parameters $w$ according to

$$w \rightarrow w + \alpha \sum_{i=1}^{N} \nabla [\log D_w(f_i) + \log(1 - D_w(G_\Omega(e_i)]$$

Theoretically, the optimal parameters are a solution to the min-max problem: $\min_\Omega \max_w \mathbb{E}[\log(D_w(F)) + \log(1 - D_w(G_\Omega(E)))]$, which reduces to $\min_\Omega JS(P_F \mid P_\Omega)$. If a generator wins the game against an ideal discriminator on a very large number of samples, then $F$ and $\Omega E$ can be shown to be close in Jensen-Shannon divergence, and thus the model has learned the true data distribution. This result, referring to the distributions of the data, $p_{data}$, and the distribution, $p_g$, $G$ is sampling from, is from Goodfellow et al. (2014): If $G$ and $D$ have enough capacity, and at each step of training, the discriminator is allowed to reach its optimum given $G$, and $p_g$ is updated so as to improve the criterion

$$E_{\mathbf{x} \sim p_{data}}[\log D^*_G(\mathbf{x})] + E_{\mathbf{x} \sim p_g}[\log(1 - D^*_G(\mathbf{x}))]$$

then $p_g$ converges to $p_{data}$. This result relies on a number of assumptions that do not hold in practice. The generator in MUSE, which learns a linear transform $\Omega$, has very limited capacity, for example, and we are updating $\Omega$ rather than $p_g$. In practice, therefore, during training, MUSE alternates between $k$ steps of optimizing the discriminator and one step of optimizing the generator. Another common problem with training GANs is that the discriminator loss quickly drops to zero, when there is no overlap between $p_g$ and $p_{data}$ (Arjovsky et al., 2017); but note that in our case, the discriminator is initially presented with $IE$ and $F$, for which there is typically no trivial solution, since the embedding spaces are likely to overlap. We show in §4 that discriminator and generator loss are poor model selection criteria, however; instead we propose a simple criterion based on cosine similarities between nearest neighbors in the learned alignment.

From $\Omega E$ and $F$, we can extract a bilingual dictionary using nearest neighbor queries, i.e., by asking what is the nearest neighbor of $\Omega E$ in $F$, or vice versa. MUSE uses a normalized nearest neighbor retrieval method to reduce the influence of hubs (Radovanović et al., 2010; Dinu et al., 2015). The method is called *cross-domain similarity local scaling* (CSLS) and used to expand high-density areas and condense low-density ones. The mean similarity of a source language embedding $\Omega\mathbf{e}$ to its $k$ nearest neighbors in the target language ($k = 10$ suggested) is defined as $\mu_E^k(\Omega(\mathbf{e})) = \frac{1}{k}\sum_{i=1}^{k} cos(\mathbf{e}, \mathbf{f}_i)$, where $cos$ is the cosine similarity. $\mu_F(\mathbf{f}_i)$ is defined in an analogous manner for every $i$. $CSLS(\mathbf{e}, \mathbf{f}_i)$ is then calculated as $2cos(\mathbf{e}, \mathbf{f}_i) - \mu_E(\Omega(\mathbf{e})) - \mu_F(\mathbf{f}_i)$. MUSE uses an unsupervised validation criterion based on CSLS. The translations of the top 10k most frequent words in the source language are obtained with CSLS and average pairwise cosine similarity is computed over them. This metric is considered indicative of the closeness between the projected source space and the target space, and is found to correlate well with supervised evaluation metrics. After inducing a bilingual dictionary, $E_d$ and $F_d$, by querying $\Omega E$ and $F$ with CSLS, MUSE performs a refinement step based on the Procrustes algorithm (Schönemann, 1966), whereby the singular value decomposition of $F_d^T E_d$, computed as $U\Sigma V^T$, gives $\Omega = UV^T$.

The idea of minimizing nearest neighbor similarity for unsupervised model selection is also found in point set registration and lies at the core of iterative closest point (ICP) optimization (Besl and McKay, 1992). ICP typically minimizes the $\lambda_2$ distance (mean squared error) between nearest neighbor pairs. The ICP optimization algorithm works by assigning each transformed vector to its nearest neighbor and then computing the new relative transformation that minimizes the cost function with respect to this assignment. ICP can be shown to converge to local optima (Besl and McKay, 1992), in polynomial time (Ezra et al., 2006). ICP easily gets trapped in local optima, however, exact algorithms only exist for two- and three-dimensional point set registration, and these algorithms are slow (Yang et al., 2016). Generally, it holds that the optimal solution to the GAN min-max problem is also optimal for ICP. To see this, note that a GAN minimizes the Jensen-Shannon divergence between $F$ and $\Omega E$. The optimal solution to this is $F = \Omega E$. As sample size goes to infinity, this means the $\mathcal{L}_2$ loss in ICP goes to 0. In other words, ICP loss is minimal if an optimal solution to the UBDI min-max problem is found. ICP was independently proposed for UBDI in Hoshen and Wolf (2018). They report their method only works using PCA initialization. We explored PCA initialization for MUSE, but observed the opposite effect, namely that PCA initialization leads to a degradation in performance.

## 3 When and why is unsupervised alignment hard?

A function $\Omega$ from $E$ to $F$ is a linear transformation if $\Omega(f+g) = \Omega(f) + \Omega(g)$ and $\Omega(kf) = k\Omega(f)$ for all elements $f, g$ of $E$, and for all scalars $k$. An invertible linear transformation is called an *isomorphism*. The two vector spaces $E$ and $F$ are called isomorphic, if there is an isomorphism from $E$ to $F$. Equivalently, if the kernel of a linear transformation between two vector spaces of the same dimensionality contains only the zero vector, it is invertible and hence an isomorphism. Most work on supervised or unsupervised alignment of word vector spaces relies on the assumption that they are approximately isomorphic, i.e., isomorphic after removing a small set of vertices (Mikolov et al., 2013; Barone, 2016; Zhang et al., 2017; Conneau et al., 2018). In this section, show that word vector spaces are *not* necessarily approximately isomorphic. We will refer to cases of non-approximately isomorphic word vector spaces as *impossible* cases. The possible cases can be further divided into easy and hard cases; corresponding to the cases where GAN-based UBDI is stable and unstable (i.e., performance is highly dependent on initialization), respectively.

It is not difficult to see why hard cases may arise when using GANs for unsupervised alignment of vector spaces. One example of a hard (but not impossible) problem instance is the case of two smoothly populated vector spaces on unit spheres. In this case, there is an infinite set of equally good linear transformations (rotations) that achieve the same training loss. Similarly, for two binary-valued, $n$-dimensional vector spaces with one vector in each possible position. Here the number of local optima would be $2^n$, but since the loss is the same in each of them the loss landscape is highly non-convex, and the basin of convergence is therefore very small (Yang et al., 2016). The chance of aligning the two spaces using gradient descent optimization would be $\frac{1}{2^n}$. In other words, minimizing the Jensen-Shannon divergence between the word vector distributions, even in the easy case, is not always guaranteed to uncover an alignment between translation equivalents. From the above, it follows that alignments between linearly alignable vector spaces cannot always be learned using UBDI methods. In §3.1 , we test for approximate isomorphism to decide whether two vector spaces are linearly alignable.§3.2–3.3 are devoted to analyzing *when* alignments between linearly alignable vector spaces can be learned.

In our experiments in §3 and 4, Bengali and Cebuano embeddings are pretrained by FastText;[1] all others are trained using FastText on Polyglot.[2] In the experiments in §5, we use FastText embeddings pretrained on Wiki and Common Crawl data.[3] If not indicated otherwise, we use MUSE with default parameters (Conneau et al., 2018).

### 3.1 Linear alignability and graph similarity

*Procrustes fit* (Kementchedjhieva et al., 2018) is a simple linearity test, which, as we find, captures the dynamics of GAN-based UBDI well. Compared to isomorphism and isospectrality tests, *Procrustes fit* is inexpensive and can be run with bigger dictionary seeds.

**Procrustes fit** The idea behind this test is to apply a Procrustes analysis (see §2) on a sizeable dictionary seed (5000 tokens), to measure the training fit. Since $UV^TE = F$ if and only if $E$ and $F$ are isomorphic, the Procrustes fit tests the linear alignability between two embedding spaces exists. We can correlate the Procrustes fit measure with the performance of UBDI. While UBDI is motivated by cases where dictionary seeds are *not* available, and Procrustes fit relies on dictionary seeds, a strong correlation can act as a sanity check on UBDI, as well as a tool to help us understand its limitations. The relationship between Procrustes fit and UBDI performance is presented in Figure 1 and shows a very strong correlation. One immediate conclusion is that the poor UBDI performance on languages such as Bengali and Cebuano is not a result of GANs being a poor estimator of the linear transforms, but rather a result of there not being a good linear transform from English into these languages.[4]

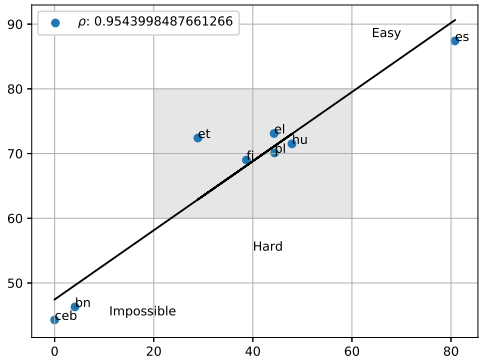



Figure 1: Correlation of UBDI performance (P@1; $x$-axis) with Procrustes fit ($y$-axis)

**Isomorphism and isospectrality** We briefly compare *Procrustes fit* to two similarity measures for nearest neighbor graphs of vector spaces, introduced in Søgaard et al. (2018). The nearest neighbor

[1] https://github.com/facebookresearch/MUSE

[2] https://polyglot.readthedocs.io/

[3] https://fasttext.cc/docs/en/crawl-vectors.html

[4] We also experimented with learning non-linear alignment using GANs with *non-linear* generators, specifically, a generator with two hidden layers connected by a non-linearity. Our results on Spanish were much poorer than with linear generators, as expected: 35.9 with *tanh*; 0.0 with *sigmoid*; 21.4 with *relu*. On Cebuano, where the linear generator failed, we observed the same result with all three non-linear approaches: P@1 = 0.0 (All numbers are averaged over 5 runs).

| | EASY | HARD | | | | IMPOSSIBLE | |
|---|---|---|---|---|---|---|---|
| | es | el | et | fi | hu | pl | bn | ceb |
| **Procrustes fit** | 87.4 | 73.1 | 72.4 | 69.0 | 71.5 | 70.1 | 46.3 | 44.3 |
| **10-Isospectrality** | 02.5 | 05.5 | 05.5 | 02.7 | 02.6 | 03.2 | 03.1 | 03.4 |
| **100-Isospectrality** | 04.2 | 42.3 | 66.9 | 15.4 | 13.5 | 06.8 | 08.8 | 16.5 |
| **Directionality** | 0.35 | 0.18 | 0.19 | 0.23 | 0.28 | 0.30 | * | * |
| **Vocab size** | 138,385 | 37,397 | 29,587 | 80,999 | 88,681 | 139,641 | 145,351 | 695,368 |

Table 1: Dataset properties. See §3.1 for **Procrustes fit**. $k$-**Isospectrality** is $k$-subgraph isospectrality (§3.1). **Directionality** is the variance in the inner products with means (Mimno and Thompson, 2017). Languages are Spanish (es), Greek (el), Estonian (et), Finnish (fi), Hungarian (hu), Polish (pl), Bengali (bn), and Cebuano (ceb). **Directionality** scores for Bengali and Cebuano are not comparable.

graph of a word vector space is obtained by adding edges between any word vertex and its nearest neighbor. Note that only cycles of length 2 are possible in a nearest neighbor graph. Two nearest neighbor graphs are *graph isomorphic* if they contain the same number of vertices connected in the same way. Two isomorphic vector spaces have isomorphic nearest neighbor graphs, but *not* vice versa. We say that the nearest neighbor graphs are $k$-subgraph isomorphic if the nearest neighbor graphs for the most frequent $k$ words (in the source language and their translations) are isomorphic. There are exact algorithms, e.g., VF2 (Cordella et al., 2001), for checking whether two nearest neighbor graphs are graph isomorphic. These algorithms do not scale easily to graphs with hundreds of thousands of nodes, however. Also, the algorithms do not identify approximate isomorphism, unless run on all subgraphs with $k$ vertices removed. Such tests are therefore impractical.

Søgaard et al. (2018) instead introduce a spectral metric based on eigenvalues of the Laplacian of the nearest neighbor graphs, similar to metrics used for graph matching problems in computer vision (Reuter et al., 2005) and biology (Lewitus and Morlon, 2016). The metric quantifies to what extent the nearest neighbor graphs are isospectral. Note that (approximately) isospectral graphs need not be (approximately) isomorphic, but (approximately) isomorphic graphs are always (approximately) isospectral. Let $A_1$ and $A_2$ be the adjacency matrices of the nearest neighbor graphs $G_1$ and $G_2$ of our two word embeddings, respectively. Let $L_1 = D_1 - A_1$ and $L_2 = D_2 - A_2$ be the Laplacians of the nearest neighbor graphs, where $D_1$ and $D_2$ are the corresponding diagonal matrices of degrees. We then compute the eigensimilarity of the Laplacians of the nearest neighbor graphs, $L_1$ and $L_2$. For each graph, we find the smallest $k$ such that the sum of the $k$ largest Laplacian eigenvalues is <90% of the Laplacian eigenvalues. We take the smallest $k$ of the two, and use the sum of the squared differences between the largest $k$ Laplacian eigenvalues $\Delta$: $\Delta = \sum_{i=1}^{k}(\lambda_{1_i} - \lambda_{2_i})^2$, where $k$ is chosen s.t. $\min_j\{\frac{\sum_{i=1}^{k}\lambda_{j_i}}{\sum_{i=1}^{n}\lambda_{ji}} > 0.9\}$. Note that $\Delta = 0$ means the graphs are isospectral, and the metric goes to infinite. Thus, the higher $\Delta$ is, the *less* similar the graphs (i.e., their Laplacian spectra). Isospectrality varies with Procrustes fit; to see this, we show that $\sum_{i=1}^{k}(\lambda_{E_i} - \lambda_{F_i})^2$ varies with $\sum_{f_i \in F_d, e_i \in UV^T(E_d)} |f_i = CSLS(e_i)|$. If $E = F$, it holds that $\sum_{i=1}^{k}(\lambda_{1_i} - \lambda_{2_i})^2 = 0$. Since $UV^T = \arg\min_\Omega ||\Omega E - F||_F^2$, in this case $\Omega = I$. Two isomorphic graphs also have the same set of sorted eigenvalues, i.e., $\sum_{i=1}^{k}(\lambda_{1_i} - \lambda_{2_i})^2 = 0$. In general, it holds that if we add an edge to a graph $G$, to form $G'$, its spectrum changes monotonically (So, 1999). Since the Procrustes fit evaluates the nearest neighbor graph, it follows that a change in the nearest neighbor graph leading to a drop in Procrustes fit will also lead to a drop in eigenvector similarity. However, isomorphism and isospectrality tests are computationally expensive, and in practice, we have to sample subgraphs and run the tests on multiple subgraphs, which leads to a poor approximation of the similarities of the two embedding graphs.

In practice, Procrustes fit, $k$-subgraph isomorphism, and $k$-subgraph isospectrality thus *all* rely on a dictionary. The tests are therefore not diagnostic tools, but means to understand the dynamics of UBDI. Procrustes fit is more discriminative (since vector space isomorphism entails nearest neighbor graph isomorphism, not vice versa) and computationally more efficient. In our experiments, it also correlates much better with UBDI performance (MUSE in Table 2; the correlation coefficient is

96%, compared to 0% for $k$-subgraph isomorphism (not listed in Table 2), and -27% for $k$-subgraph isospectrality with $k = 10$.

**Observation 1** *Impossible cases are not (solely) the result of linguistic differences, but also of corpus characteristics.* English-Bengali and English-Cebuano are not linearly alignable according to our Procrustes fit tests. There can be two explanations for such an observation: linguistic differences between the two languages or variance in the monolingual corpora for Bengali and Cebuano, i.e. noise and little support per word. We test for this by applying the Procrustes fit test to the word vector spaces of Bengali and a higher resource related language, Hindi. The Procrustes fit for Bengali-Hindi is even lower than for English-Bengali (30.01, compared to 46.25, respectively). This finding is surprising as we would expect Bengali and Hindi to align well due to their relatedness. The result thus suggests that the Bengali embeddings are of insufficient quality, which can largely explain the poor alignment found by the GAN. This is further supported by follow-up experiments we ran aligning a word vector space for English and a word vector space induced from scrambled English sentences (learned on two different 10% samples of Wikipedia), which can be thought of as a sample from a synthetic language that completely diverges from English in its syntactic properties.[5] GAN-based UBDI was able to near-perfectly recover the word identities without supervision, showing that its success is not easily impeded by linguistic differences.

**Observation 2** *Impossible cases can also be the result of the inductive biases of the underlying word embedding algorithms.* One observation made in Conneau et al. (2018) is that the performance of MUSE degrades a little when using alternative embedding algorithms, but that alignment is still possible. We, however, observe that this is not the case if using *different*, monolingual embedding algorithms, i.e., if using FastText for English and Hyperwords for Spanish. While such embeddings are still linearly alignable (as verified by computing their Procrustes fits), GAN-based UBDI consistently fails on such cases. This also holds for the case of aligning FastText for English and Hyperwords for *English*, as observed in Hartmann et al. (2018).

### 3.2 Ablation transformations

In order to better understand the dynamics of GAN-based UBDI in hard cases, i.e., when the GAN suffers from local minima, we introduce three *ablation transformations*, designed to control for properties of the word vector spaces: unit length normalization, PCA-based pruning, and noising. The results of GAN-based UBDI after applying these transforms are reported in Table 2.

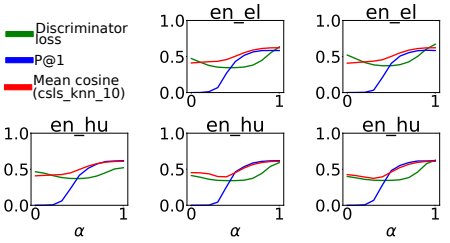
**Observation 3** *GAN-based UBDI becomes more unstable and performance deteriorates with unit length normalization.* This ablation transform performs unit length normalization (ULN) of all vectors $\mathbf{x}$, i.e., $\mathbf{x}' = \frac{\mathbf{x}}{||\mathbf{x}||^2}$, and is often used in supervised bilingual dictionary induction (Xing et al., 2015; Artetxe et al., 2017). We use this transform to project word vectors onto a sphere – to control for shape information. If vectors are distributed smoothly over two spheres, there is no way to learn an alignment in the absence of dictionary seed; in other words, if UBDI is unaffected by this transform, UBDI learns from density information alone. While supervised methods are insensitive to or benefit from ULN, we find that UBDI is very sensitive to such normalization (see Table 2, M-unit). We verify that supervised alignment is not affected by ULN by checking the Procrustes fit (§3.1), which remains unchanged under this transformation.

**Observation 4** *GAN-based UBDI becomes more unstable and performance deteriorates with PCA pruning.* In order to control for density, we apply PCA to our word vector spaces, reducing them to 25 dimensions, and prune our vocabularies to remove density clusters by keeping all but one of the nearest neighbors vectors on an integer grid. This removes about 10% of our vocabularies. We then apply UBDI to the original vectors for the remaining words. This smoothening of the embeddings results in highly unstable and reduced performance (see Table 2, M-PCA). In other words, density information, while less crucial than shape information, is important for the stability of UBDI, possibly by reducing the chance of getting stuck in local optima. This is in contrast with the results on using ICP for UBDI in Hoshen and Wolf (2018), who report significant improvements

[5]The context window for learning the embeddings is smaller than the average sentence length, which ensures that words in scrambled English are seen in different contexts than in regular English.

| | EASY | | HARD | | | | | | | | | | IMPOSSIBLE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | es | | el | | et | | fi | | hu | | pl | | bn | | ceb | |
| | P@1 | fail | P@1 | fail | P@1 | fail | P@1 | fail | P@1 | fail | P@1 | fail | P@1 | fail | P@1 | fail |
| MUSE | 80.8 | 0 | 35.4 | 2 | 23.1 | 2 | 31.0 | 2 | **47.9** | 0 | 31.1 | 3 | 00.0 | 10 | 00.0 | 10 |
| M-unit | 80.4 | 0 | 17.7 | 5 | 12.6 | 5 | 32.3 | 1 | 14.4 | 7 | 24.1 | 4 | 00.0 | 10 | 00.0 | 10 |
| M-PCA | 74.1 | 1 | 09.3 | 8 | 10.9 | 3 | 03.5 | 8 | 38.2 | 2 | 18.0 | 6 | 00.0 | 10 | 00.0 | 10 |
| M-noise | 80.1 | 0 | 33.7 | 2 | 19.4 | 2 | 34.7 | 0 | 42.2 | 1 | 40.9 | 1 | 00.0 | 10 | 00.0 | 10 |
| M-discr | 77.5 | 0 | 26.5 | 4 | 09.4 | 7 | 26.3 | 2 | 09.3 | 8 | 19.6 | 6 | 00.0 | 10 | 00.0 | 10 |
| M-cosine | **81.3** | 0 | **44.8** | 0 | **28.7** | 0 | **36.3** | 0 | 47.2 | 0 | **45.2** | 0 | 00.0 | 10 | 00.0 | 10 |

Table 2: Main experiments; average performance and stability across 10 runs. We consider a P@1 score below 1% a fail. MUSE is the MUSE system with default parameters. Ablation transforms: M-unit uses unit length normalization to evaluate the impact of shape; M-PCA uses PCA-based pruning to evaluate the impact of density; M-noise uses 25% random vectors injected in the target language space to evaluate the impact of noise. M-discr uses discriminator loss for model selection, as a baseline for M-cosine; M-cosine uses our model selection criterion. The macro-averaged error reduction of M-cosine over MUSE for the HARD languages is 7%; and 4% across all language pairs.




(a) Discriminator loss (green), P@1 (blue) and mean cosine similarity (red) for generators along the line segment from the (failed) unsupervised GAN solution to the supervised Procrustes fit. $\alpha$ is the interpolation parameter.

| | | el | | | hu | | |
|---|---|---|---|---|---|---|---|
| | | P@1 (10 runs) | P@1 (top-5) | fail | P@1 (10 runs) | P@1 (top-5) | fail |
| MUSE | | **23.68** | **33.59** | 2 | 28.24 | 42.26 | 3 |
| OP | $d$=4096 | 21.39 | 32.23 | 3 | 36.16 | 42.83 | 1 |
| | $d$=8192 | 17.56 | 30.05 | 3 | **39.42** | **43.44** | 0 |
| BS | $b$=16 | 22.09 | 32.18 | 2 | 31.33 | 39.04 | 1 |
| | $b$=128 | 00.00 | 00.00 | 10 | 00.00 | 00.00 | 10 |
| LR | $l$=0.01 | 10.43 | 20.62 | 5 | 15.43 | 30.86 | 6 |
| | $l$=0.001 | 00.00 | 00.00 | 10 | 00.00 | 00.00 | 10 |

(b) Results for overparametrization (OP), increased batch sizes (BS) and decreased discriminator learning rates (LR), compared to the MUSE default hyperparameters: $d$=2048, $b$=32, $l$=0.1: averages over 10 runs and over the top 5 runs, and the number of fails.

Figure 2: Follow-up experiments on Greek (el) and Hungarian (hu)

using PCA initialization with 50 dimensions. We ran experiments with 25, 50, and 100 dimensions, with or without pruning, observing significant drops in performance across the board.

**Observation 5** *GAN-based UBDI is largely unaffected by noise injection.* We add 25% random vectors, randomly sampled from a hypercube bounding the vector set. GAN-based UBDI results are not consistently affected by noise injection (see Table 2, M-noise). This is because the injected vectors rarely end up in the seed dictionaries used for the Procrustes analysis step.

### 3.3 OVER-PARAMETERIZATION, BATCH SIZE, AND LEARNING RATE

In follow-up experiments on Greek and Hungarian, we find that GAN-based UBDI gets stuck in local optima in hard cases, and over-parameterization, increasing batch size or decreasing learning rate does not help.

**Observation 6** *In the hard cases, GAN-based UBDI gets stuck in local optima.* In cases where linear alignment is possible, but UBDI is unstable, the model might get stuck in a local optimum, which is the result of the discriminator loss not increasing in the region around the current discriminator model. We analyze the discriminator loss in these areas by plotting it as a function of the generator parameters for the failure cases of two of the hard alignment cases, namely English-Greek and English-Hungarian. We plot the loss surface along its intersection with a line segment connecting two sets of parameters (Goodfellow et al., 2015; Li et al., 2018). In our case, we interpolate between the model induced by GAN-based UBDI and the (oracle) model obtained using supervised Procrustes

analysis. Result are shown in Figure 2a. The green loss curves represent the current discriminator's loss along all the generators between the current generator and the generator found by Procrustes analysis. In all cases, we see that while performance (P@1 and mean cosine similarity) goes up, there is an initial drop in the discriminator loss, which suggests there is no learning signal in this direction for GAN-based UBDI. This is along a line segment representing the shortest path from the failed generator to the oracle generator, of course; linear interpolation provides no guarantee there are no almost-as-short paths with plenty of signal. A more sophisticated sampling method is to sample along two random direction vectors Goodfellow et al. (2015); Li et al. (2018). We used an alternative strategy of sampling from normal distributions with fixed variance that were orthogonal to the line segment. We observed the same pattern, leading us to the conclusion that instability is caused by local optima.

**Observation 7** *Over-parameterization does not consistently help in the hard cases.* Recent work has observed that over-parameterization leads to smoother loss landscapes and makes optimization easier (Brutzkus et al., 2018). We experiment with widening our discriminators to smoothen the loss landscape, but results are inconsistent: for Hungarian, this made GAN-based UBDI more stable; for Greek, less stable (see Figure 2b).

**Observation 8** *Changing the batch size or the learning rate to hurt the discriminator also does not help.* Previous work has shown that large learning rate and small batch size contribute towards SGD finding flatter minima (Jastrzebski et al., 2018), but in our experiments, we are interested in the discriminator not ending up in flat regions, where there is no signal to update the generator. We therefore experiment with *smaller* learning rate and *larger* batch sizes. The motivation behind both is decreasing the scale of random fluctuations in the SGD dynamics (Smith and Le, 2017; Balles et al., 2017), enabling the discriminator to explore narrower regions in the loss landscape. See Figure 2b for results. Increasing the batch size or varying the learning rate (up or down) clearly comes at a cost, and it seems the MUSE default hyperparameters are close to optimal.

## 4 Unsupervised model selection

In this section, we compare two unsupervised model selection criteria. We train three models with different random seeds in parallel and use the selection criterion to select one of these models to train for the remaining epochs. The first criterion is the discriminator loss during training, which is used in Daskalakis et al. (2018), for example. In contrast, we propose to use the mean cosine similarity between all translations predicted by the $CSLS$ method (see §2), which was used as an unsupervised stopping criterion by Conneau et al. (2018).

**Observation 9** *In the hard cases, model selection with cosine similarity can stabilize GAN-based UBDI.* As we see in Table 2, the selection criterion based on discriminator loss (M-discr) *increases* the instability of UBDI, leading to 4/10 failed alignments for Greek compared to 2/10 without model selection, for example. Cosine similarity (M-cosine) in contrast leads to perfectly stable UBDI. Note that if the probability of getting stuck in a local optimum that leads to a poor alignment is $\beta$, using $n$ random restarts and oracle model selection we increase the probability of finding a good alignment to $1 - (1 - \beta)^n$. In our experiments, $n = 3$.

## 5 Conclusions

Some pairs of word vector spaces are not alignable based on distributional information alone. For other pairs, GANs *can* be used to induce such an alignment, but the degree of instability is very susceptible to the shape and density of the word vector spaces, albeit not to noise. Instability is caused by local optima, but not remedied by standard techniques such as over-parameterization, increasing the batch size or decreasing the learning rate. We propose an unsupervised model selection criterion that enables stable learning, leading to a ~7% error reduction over MUSE, and present further observations about the alignability of word vector distributions.

## REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. In *CoRR*. page abs/1701.07875.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of ACL*. pages 451–462. https://doi.org/10.18653/v1/P17-1042.

Lukas Balles, Javier Romero, and Philipp Hennig. 2017. Coupling adaptive batch sizes with learning rates. In *Proceedings of UAI*.

Antonio Valerio Miceli Barone. 2016. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. *Proceedings of the 1st Workshop on Representation Learning for NLP* pages 121–126. http://arxiv.org/pdf/1608.02996.pdf.

Paul Besl and Neil McKay. 1992. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2).

Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. 2018. SGD learns over-parameterized networks that provably generalize on linearly separable data. In *Proceedings of ICLR*.

Alexis Conneau, Guillaume Lample, Marc Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *ICLR*.

L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. 2001. An improved algorithm for matching large graphs. *Proceedings of the 3rd IAPR TC-15 Workshop on Graphbased Representations in Pattern Recognition* 17:1–35.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. 2018. Training GANs with optimism. In *ICLR*.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of ICLR (Workshop Papers)*. http://arxiv.org/abs/1412.6568.

Ester Ezra, Micha Sharir, and Alon Efrat. 2006. On the ICP algorithm. In *SGC*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*. pages 462–471. http://repository.cmu.edu/lti/31.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David WardeFarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. In *Proceedings of NIPS*.

Ian J Goodfellow, Oriol Vinyals, and Andrew Saxe. 2015. Qualitatively characterizing neural network optimization problems. In *Proceedings of ICLR*.

Mareike Hartmann, Yova Kementchedjhieva, and Anders Søgaard. 2018. Why is unsupervised alignment of english embeddings from different algorithms so hard? In *Proceedings of EMNLP*.

Bradley Hauer, Garrett Nicolai, and Grzegorz Kondrak. 2017. Bootstrapping unsupervised bilingual lexicon induction. In *Proceedings of EACL*. pages 619–624.

Yedid Hoshen and Lior Wolf. 2018. An iterative closest point method for unsupervised word translation. In *CoRR*. page 1801.06126.

Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. 2018. Finding flatter minima with SGD. In *Proceedings of ICLR*.

Yova Kementchedjhieva, Sebastian Ruder, Ryan Cotterell, and Anders Søgaard. 2018. Generalizing procrustes analysis for better bilingual dictionary induction. In *Proceedings of CoNLL*.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *Proceedings of ICLR (Conference Papers)*. http://arxiv.org/abs/1711.00043.

Eric Lewitus and Helene Morlon. 2016. Characterizing and comparing phylogenies from their laplacian spectrum. *Systematic Biology* 65:495—-507.

Hao Li, Zheng Xu, Gavin Taylor, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *ICLR*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR* abs/1309.4168. http://arxiv.org/abs/1309.4168.

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of EMNLP*.

Milos Radovanović, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research* 11:2487–2531. http://portal.acm.org/citation.cfm?id=1953015.

Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. 2005. Laplace-spectra as fingerprints for shape matching. In *Proceedings of Symposium on Solid and Physical Modeling*.

Peter Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31:1–10.

Samuel Smith and Quoc Le. 2017. A Bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451* .

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of ICLR (Conference Track)*.

Wasin So. 1999. Rank one perturbation and its application to the laplacian spectrum of a graph. *Linear and Multilinear Algebra* 46:193–198.

Ander Søgaard, Sebastian Ruder, and Ivan Vulic. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of ACL*.

Ivan Vulić and Anna Korhonen. 2016. On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of ACL*. pages 247–257.

Chao Xing, Chao Liu, Dong Wang, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of NAACL-HLT*.

Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. 2016. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(11).

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of ACL*.