Zipfian Whitening

Sho Yokoi
Tohoku University / RIKEN
yokoi@tohoku.ac.jp

Han Bao
Kyoto University
bao@i.kyoto-u.ac.jp

Hiroto Kurita
Tohoku University
hiroto.kurita@dc.tohoku.ac.jp

Hidetoshi Shimodaira
Kyoto University / RIKEN
shimo@i.kyoto-u.ac.jp

Abstract

The word embedding space in neural models is skewed, and correcting this can improve task performance. We point out that most approaches for modeling, correcting, and measuring the symmetry of an embedding space implicitly assume that the word frequencies are *uniform*; in reality, word frequencies follow a highly non-uniform distribution, known as *Zipf's law*. Surprisingly, simply performing PCA whitening weighted by the empirical word frequency that follows Zipf's law significantly improves task performance, surpassing established baselines. From a theoretical perspective, both our approach and existing methods can be clearly categorized: word representations are distributed according to an exponential family with either uniform or Zipfian base measures. By adopting the latter approach, we can naturally emphasize informative low-frequency words in terms of their vector norm, which becomes evident from the information-geometric perspective [42], and in terms of the loss functions for imbalanced classification [36]. Additionally, our theory corroborates that popular natural language processing methods, such as skip-gram negative sampling [37], WhiteningBERT [26], and headless language models [23], work well just because their word embeddings encode the empirical word frequency into the underlying probabilistic model.

https://github.com/cl-tohoku/zipfian-whitening

1 Introduction

Representing discrete words by continuous vectors is a fundamental and powerful framework of modern deep-learning-based natural language processing (NLP). Static word embeddings [43, 37], dynamic word embeddings [18, 33], and causal language models [45, 12, 54] have caused a paradigm shift—they have greatly improved the performance of virtually all kinds of NLP applications and have been actively used in relevant areas as well. While the embedded units may be characters or subwords instead of words, we simply refer to them collectively as *word*.

Recently, the machine learning and NLP communities have discovered that the word embedding space is "skewed" and that correcting this can lead to better performance in downstream tasks [39, 21, 16, 56]. The isotropy of the embedding space would be one factor: vectors dispersing more evenly should be more discriminative than those clustered in the same direction [38, 21, 51]. Typically, such spatial symmetry in the embedding space is enhanced through centering/whitening [39, 16, 26].

Nevertheless, we would like to point out that most existing approaches implicitly assume *uniform* word frequency to formalize spatial symmetry. Consider the classical centering operation as an example: we first calculate the mean of the word vectors, and then subtract it to ensure they are zero-meaned. This method, however, has an unexpected pitfall. Recall that the definition of the centroid or barycenter of a random vector $\boldsymbol{x} \sim p$, assuming it has a finite set of distinct realizations,

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

is given by $\mathbb{E}_{\boldsymbol{x}\sim p}[\boldsymbol{x}] = \sum_i p(\boldsymbol{x}_i)\boldsymbol{x}_i$. The classical centering, based on the standard (unweighted) mean, implicitly assumes that all words occur uniformly $p(\boldsymbol{w}_1) = \cdots = p(\boldsymbol{w}_n)$. In reality, however, word frequencies are known to follow a highly non-uniform distribution[1], creating a significant gap between the methodology and the actual usage of words. This seemingly obvious issue does not arise when addressing classical statistical estimation problems, as data vectors in our hands are usually representations of observations or instances. In contrast, word vectors used in NLP are representations of types or classes; each of them (such as the vector for '`the`') abstracts the numerous instances (such as the tokens of '`the`') appearing in the data. This problem of *hidden* frequencies becomes apparent in the cases where the type-token distinction [58] is crucial, such as when dealing with natural language data (§ 2). The take-home message of this paper can be summarized as follows: use empirical word frequencies when calculating expected values. Following this very simple guideline leads to strong empirical outcomes (§ 3.2, § 3.3) and opens a rich theoretical landscape (§ 4, § 5).

Notation Let $\mathcal{V} = \{w_1, \ldots, w_n\}$ denote the vocabulary, i.e., the set of words in interest. Bold-face $\boldsymbol{w}_i \in \mathbb{R}^d$ denotes the row vector of each word type w_i, and $p(w_i) \in [0, 1]$ denotes its frequency.

2 Motivation: type-token distinction and expected values

Why have word frequencies been overlooked when considering the geometric properties of embedding spaces? This can be explained through the concept of *type-token distinction* [58], which is a fundamental concept in linguistics and related fields but generally not required in statistical machine learning. Here, **type** represents a class and **token** represents an instance. For example, the phrase '`perform natural language processing in a natural way`' contains eight tokens and *seven* types. The instances '`natural`' appear twice, but as a word type, it is counted only once.

With the type-token distinction in mind, let us take a fresh look at data matrices and their expected values. Typically, each row in a data matrix represents one observation, i.e., one instance **token**. If we want to centralize a set of data vectors, computing the unweighted mean is a natural way in the machine learning pipeline. On the other hand, each row of a word embedding matrix, i.e., word vector, is a **type** embedding. Each word vector abstracts the numerous instances appearing repeatedly in a corpus, though information on the frequency of instances for each word type is not encoded in it. The unweighted mean of word vectors treats type vectors as token vectors, resulting in the complete omission of word frequency information.

Let us describe the above idea formally. The data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ or the set of data vectors $\{\boldsymbol{x}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ represents a collection of instances, observations, or **tokens**; then the empirical distribution is $\mu_{\boldsymbol{X}} = \sum_{i=1}^n \frac{1}{n}\delta(\boldsymbol{x}_i)$, where δ is the Dirac delta function. Here, the unweighted mean can be seen as the expectation $\widehat{\mathbb{E}}_{\boldsymbol{x}\sim\mu_{\boldsymbol{X}}}[\boldsymbol{x}] = \sum_{i=1}^n \frac{1}{n}\boldsymbol{x}_i$ with the empirical distribution. On the other hand, the word embedding matrix $\boldsymbol{W} \in \mathbb{R}^{n\times d}$ or the set of word vectors $\{\boldsymbol{w}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ represents a collection of **types**. When describing the empirical distribution, the hidden frequency p of tokens is necessary. Given p, the empirical distribution is $\mu_{\boldsymbol{W}} = p(w_i)\delta(\boldsymbol{w}_i)$. From this perspective, the centroid of the word vectors should be written as the expectation $\widehat{\mathbb{E}}_{\boldsymbol{w}\sim\mu_{\boldsymbol{W}}}[\boldsymbol{w}] = \sum_i p(w_i)\boldsymbol{w}_i$ over p.

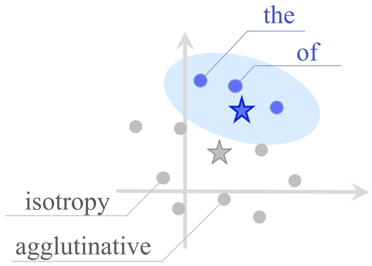

Figure 1: Low-frequent words {●} and high-frequent words {●} are unevenly distributed in the embedding space [39, 24, 44, 10]. Consequently, the "apparent" mean calculated by unweighted averaging ★ often differs from the actual centroid ★.

The distinction is not just "theoretical." First, refer to Fig. 1. Word vectors are known to cluster by frequency [39, 24, 44, 10]. In this situation, the centroid ★ weighted by the word frequencies is located near the narrow region where high-frequent words are concentrated (a region with a light blue background), and thus differs from the unweighted mean ★. Second, see Table 1, which shows

[1] As known as Zipf's law. If we count the frequencies of words in huge English corpora, we find that '`the`' has a frequency of about 5.89×10^{-2} and '`isotropy`' has a frequency of about 3.47×10^{-8}, a difference of a million times greater.

50 words sampled from each of types and tokens. Uniform sampling from types, corresponding to an unweighted mean, tends to select mostly rare words from the heavy tail. Sampling from tokens clearly captures a more natural representation of language as it typically appears in text.

Table 1: The difference between type-based sampling and token-based sampling.

Words sampled from types	Words sampled from tokens
‘scintillation’, ‘fanon’, ‘rubato’, ‘upstanding’, ‘collard’, ‘creeks’, ‘skookum’, ‘unbelievers’, ‘monocyte’, ‘nishikawa’, ‘crusher’, ‘gerwen’, ‘abrah’, ‘silverchair’, ‘hangman’, ‘unitary’, ‘klausen’, ‘arousal’, ‘heat’, ‘bridgnorth’, ‘mildred’, ‘porton’, ‘aquasox’, ‘wylie’, ‘hipaa’, ‘krimuk’, ‘hexahedron’, ‘kuei’, ‘barbera’, ‘dalvi’, ‘gilding’, ‘visakhapatnam’, ‘tatsuo’, ‘tarascon’, ‘bajram’, ‘scholes’, ‘hadad’, ‘incidental’, ‘theodosius’, ‘reichskommissariat’, ‘boeheim’, ‘amsl’, ‘buencamino’, ‘thrasyvoulos’, ‘insulated’, ‘discourtesy’, ‘nisra’, ‘ycko’, ‘luen’, ‘dooku’	‘nine’, ‘ranked’, ‘zero’, ‘the’, ‘garcia’, ‘rank’, ‘station’, ‘the’, ‘for’, ‘four’, ‘williams’, ‘drunken’, ‘a’, ‘one’, ‘eight’, ‘of’, ‘were’, ‘zero’, ‘debate’, ‘orchestra’, ‘of’, ‘wrist’, ‘points’, ‘fractured’, ‘the’, ‘to’, ‘redirect’, ‘adnan’, ‘white’, ‘car’, ‘fond’, ‘concluded’, ‘under’, ‘two’, ‘by’, ‘five’, ‘his’, ‘infection’, ‘the’, ‘the’, ‘pop’, ‘in’, ‘one’, ‘in’, ‘one’, ‘one’, ‘fram’, ‘handled’, ‘battle’, ‘mutual’

3 Embedding symmetry

3.1 Definition of embedding symmetry

In mathematical science fields, such as high-dimensional probability theory [55] and the volume of convex bodies [27], there are numerous intriguing definitions of spatial symmetry. Among them, we begin with the definition of the symmetry of *random* vectors with their frequencies [48, 55]. This is suited for dealing with word vectors because they entail word *frequencies*, unlike usual data instances.

Definition 1 (A random vector $\boldsymbol{v} \sim p$ on $\mathbb{R}^d$ has zero mean; the *1st* moment of a symmetric random vector)**.**

$$\overline{\boldsymbol{v}} := \mathbb{E}_{\boldsymbol{v}\sim p}[\boldsymbol{v}] = \boldsymbol{0} \qquad (1)$$

Definition 2 (A random vector $\boldsymbol{v} \sim p$ on $\mathbb{R}^d$ is in isotropic position around its barycenter; the *2nd* moment of a symmetric random vector)**.**

$$\mathrm{Cov}[\boldsymbol{v}] := \mathbb{E}_{\boldsymbol{v}\sim p}[(\boldsymbol{v} - \mathbb{E}_{\boldsymbol{v}\sim p}[\boldsymbol{v}])(\boldsymbol{v} - \mathbb{E}_{\boldsymbol{v}\sim p}[\boldsymbol{v}])^\top] \propto \boldsymbol{I}_d \qquad (2)$$

From these definitions, we will develop methods to *adjust* given word vectors to be symmetric in § 3.2, and to *evaluate* the symmetry of given word vectors in § 3.3.

In machine learning and NLP, the spatial symmetry of embedding spaces is a hot topic, and numerous theories and algorithms have been proposed [41, 21, 38, 56]. However, the approach in many researches implicitly treats all vectors equally, ignoring word frequency information. In the following sections, we will detail both the empirical and theoretical issues that a uniform approach can cause, especially when applied to NLP tasks. Furthermore, when embeddings correspond to tokens rather than types—such as in the internal representations of masked or causal language models—a uniform approach tends to be effective. This point will be discussed in § 5.1.

3.2 Enhancement of embedding symmetry

This section proposes **Zipfian whitening**[2], which symmetrizes a given set of word vectors with word frequency. At a glance, the most natural method to achieve Def. 1 and Def. 2 would be PCA whitening, also known as sphering. Notably, each step of whitening—centering, decorrelation, and standardization—implicitly involves calculating expected values. Our approach is simple: each time we calculate an expected value, we should weight it by the empirical word frequency. The specific algorithm is as shown in Algorithm 1. The only difference from general whitening is that it uses word frequency in the part highlighted in blue . Please refer to Appendix A for a formal explanation showing that the word vectors obtained by the proposed algorithm actually satisfy Def. 1 and Def. 2.

[2] In this paper, “Zipfian” is simply used to denote a “highly non-uniform” distribution. Our focus is on the mismatch between actual word frequencies and uniform distribution, and we have *not* constructed arguments or experiments that rely on specific properties of power laws. Refining experiments and theory based on the degree of tail heaviness is an interesting direction for future work.

Algorithm 1 Zipfian whitening; a post-processing algorithm on word embeddings. The part highlighted in blue shows the difference from the typical centering and whitening.

```
Input: Word embeddings $\{\boldsymbol{w}_i \in \mathbb{R}^d \mid w_i \in \mathcal{V}\}$, word frequency $p\colon \mathcal{V} \to [0,1]$.
Output: Processed word embeddings. $\{\overline{\boldsymbol{w}}_i \in \mathbb{R}^d\}_i$ are centered, $\{\widetilde{\boldsymbol{w}}_i \in \mathbb{R}^d\}_i$ are further whitened.
    Zipfian centering (1st moment):
    $\widehat{\boldsymbol{\mu}} \leftarrow \sum_{w_i \in \mathcal{V}} p(w_i)\boldsymbol{w}_i \in \mathbb{R}^d$
    for all $w_i \in \mathcal{V}$ do
        $\overline{\boldsymbol{w}}_i \leftarrow \boldsymbol{w}_i - \widehat{\boldsymbol{\mu}} \in \mathbb{R}^d$
    end for
    Zipfian decorrelation and standardization (2nd moment):
    $\boldsymbol{W}_p \leftarrow \left[\sqrt{p(w_1)}\,\overline{\boldsymbol{w}}_1^\top, \dots, \sqrt{p(w_{|\mathcal{V}|})}\,\overline{\boldsymbol{w}}_{|\mathcal{V}|}^\top\right]^\top \in \mathbb{R}^{|\mathcal{V}| \times d}$
    $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \leftarrow \mathrm{SVD}(\boldsymbol{W}_{\boldsymbol{p}})$
        ▷ $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \dots, \sigma_d) \in \mathbb{R}^{d \times d}$ consists of the singular values of $\boldsymbol{W}_p$.
    for all $w_i \in \mathcal{V}$ do
        $\widetilde{\boldsymbol{w}}_i \leftarrow \overline{\boldsymbol{w}_i}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}$
            ▷ $\boldsymbol{\Sigma}^{-1} := \mathrm{diag}(1/\sigma_1, \dots, 1/\sigma_d) \in \mathbb{R}^{d \times d}$.
    end for
```

Table 2: The empirical performance of Zipfian whitening, which exploits the empirical frequency of words during expectation calculations. Each cell shows the STS-B [15] score ×100. By carefully performing the simple operation of whitening, it consistently outperforms powerful baseline methods.

GloVe	46.17	
	Uniform	Zipfian
+ Centering	45.17	**52.25**
+ Whitening	52.21	**66.92**
+ ABTT [39]	54.28	
+ SIF + CCR [7]	58.70	

Word2Vec	56.98	
	Uniform	Zipfian
+ Centering	55.85	**58.84**
+ Whitening	56.03	**66.50**
+ ABTT [39]	56.98	
+ SIF + CCR [7]	63.04	

Empirical evaluation: We confirm the effectiveness of Zipfian whitening (Algorithm 1) by measuring performance on standard sentence-level downstream tasks using post-processed word vectors. We employed the most standard word embeddings—GloVe [43], word2vec [37], and fastText [11]—and utilized the widely adopted evaluation tasks, including STS-B [15] and related benchmarks. Detailed experimental settings can be found in Appendix B. Table 2 shows the results on the STS-B task. Remarkably, the proposed Zipfian whitening shows significant advantages not only over standard (uniform) centering and whitening but also over the strong baseline method [7] specifically designed to create powerful sentence vectors. Consistent results were obtained with various benchmark datasets, multiple empirical word probabilities, and a language other than English (Appendix C)[3]. In § 4.2, one reason for this remarkable performance is clarified from the perspective of information geometry.

3.3 Evaluation of embedding symmetry

The community is greatly interested not only in making word vector spaces symmetric but also in evaluating *how symmetric* or asymmetric a space is [21, 49]. Here, we return to Def. 1 and Def. 2 and describe metrics for evaluating the symmetry of word embedding spaces with word frequency.

Degree of centrality—the 1st moment of symmetry: Recall that, if the barycenter $\mathbb{E}[\boldsymbol{v}]$ is close to $\mathbf{0}$, then the random vector $\boldsymbol{v}$ can be considered symmetric in terms of the first moment (Def. 1).

[3] Notably, we observed improved scores when using word frequencies from the evaluation dataset itself as $p(w)$. In general, for NLP tasks, $p(w)$ refers to word frequencies derived from the embedding training data or from a standard large corpus. However, to optimize downstream task performance, it is preferable to base $p(w)$ on word frequencies within the evaluation dataset itself used for those tasks. This adjustment exemplifies "covariate shift" [52] in machine learning, where the distribution of training data differs from that of test data.

Thue, examining the value of $\|\mathbb{E}[\boldsymbol{v}]\| := \mathbb{E}[\boldsymbol{v}] - \mathbf{0}$ appears to be a reasonable way to measure the symmetry of the first moment. However, random vectors $\boldsymbol{v}$ and $\alpha\boldsymbol{v}$ ($\alpha \in \mathbb{R}_{>0}$) should be considered equivalent in terms of spatial symmetry. Thus, we define the scale-invariant metric (Def. 3), obtained by dividing $\|\mathbb{E}[\boldsymbol{v}]\|$ by the average length $\mathbb{E}[\|\boldsymbol{v}\|]$.

Definition 3 (Degree of centrality for the random vector $\boldsymbol{v} \sim p$; the 1st moment of symmetry)**.**

$$\mathrm{Sym}_1(\boldsymbol{v}) := 1 - \left\| \mathop{\mathbb{E}}_{\boldsymbol{v}\sim p}[\boldsymbol{v}] \right\| / \mathop{\mathbb{E}}_{\boldsymbol{v}\sim p}[\|\boldsymbol{v}\|] \tag{3}$$

By definition, $\mathrm{Sym}_1(\boldsymbol{v})$ takes values in $[0, 1]$, and $\mathrm{Sym}_1(\boldsymbol{v}) = 1$ if and only if $\boldsymbol{v}$ is zero mean.

Degree of isotropy—the 2nd moment of symmetry: If the covariance matrix $\mathbb{E}[(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])^\top]$ is a constant multiple of the identity matrix $\boldsymbol{I}_d$, i.e., if the random vector $\boldsymbol{v}$ has an equal spread in all directions, $\boldsymbol{v}$ is symmetric in terms of the second moment (Def. 2). Following convention, this degree can be confirmed by examining the flatness of the eigenspectrum.

Definition 4 (Degree of isotropy around the barycenter for the random vector $\boldsymbol{v} \sim p$; the 2nd moment of symmetry)**.**

$$\mathrm{Sym}_2(\boldsymbol{v}) := \frac{1}{\log d} H\left(\frac{\lambda_1}{\sum_j \lambda_j}, \ldots, \frac{\lambda_d}{\sum_j \lambda_j}\right) \tag{4}$$

$\{\lambda_1, \ldots, \lambda_d\}$ *are the eigenvalues of the covariance matrix* $\mathop{\mathbb{E}}_{\boldsymbol{v}\sim p}[(\boldsymbol{v} - \mathop{\mathbb{E}}_{\boldsymbol{v}\sim p}[\boldsymbol{v}])(\boldsymbol{v} - \mathop{\mathbb{E}}_{\boldsymbol{v}\sim p}[\boldsymbol{v}])^\top]$.
$H(p_1, \ldots, p_d) := -\sum_i p_i \log p_i$ *is the Shannon entropy.*

Proposition 1. $\mathrm{Sym}_2(\boldsymbol{v})$ *takes values in* $[0, 1]$*, and* $\mathrm{Sym}_2(\boldsymbol{v}) = 1$ *if and only if* $\boldsymbol{v}$ *is isotropic around its barycenter (Def. 2).* *Proof.* Please refer to Appendix D.

Note that the approach of measuring the entropy of the spectrum to evaluate the flatness of a signal can be found in many fields. For example, similar definitions are seen in probability processes [14] and signal processing [17, 47]. We also follow this standard and powerful line.

Algorithm: To compute the evaluation metrics of symmetry (Def. 3, Def. 4) for given word vectors, again, one should just use the empirical word frequency when calculating the expectations. A pseudocode for measuring symmetry is provided in Appendix E.

Empirical evaluation: To what extent does our symmetry score (an intrinsic evaluation of embedding spaces) correlate with downstream task performance (an extrinsic evaluation of those)? As baselines, we use versions of our symmetry score that do *not* account for word frequency, calculated in a uniform manner. We also compare with popular symmetry scores in NLP, the average of cosine similarity (**Ave. Cos.**) [21] and the recently proposed **IsoScore** [49]. Note that all these baselines implicitly assume uniform word frequency. Additional experimental settings can be found in Appendix B. Fig. 2 shows the results. The right side of Fig. 2 demonstrates the superiority of the Zipfian approach. Moving from the bottom-left to the top-right of the figure—i.e. as both the 1st (x-axis) and 2nd moments (y-axis) of the symmetry score increase—it is clearly visible that the downstream task performance increases (the color becomes more red). In contrast, in the left-hand plot, which assumes uniform word frequency, there is no observed relationship between the symmetry score (x and y-axis) and the downstream task performance (color). Table 3 lists the correlation coefficients between the symmetry scores and downstream task performance in more detail. It can be seen that the symmetry scores considering word frequency can "predict" downstream task performance with remarkably high correlation. On the other hand, the "prediction" performance of other metrics, including Ave. Cos. and IsoScore that implicitly assume uniform word frequency, is unsatisfactory. Surprisingly, when the most popular Ave. Cos. metric shows almost no correlation (0.04) with downstream task performance (STS-B), Zipfian symmetry metric has a strong positive correlation (0.83) with it.

4 Why is Zipfian whitening better than uniform whitening?

A natural question is why the Zipfian approach empirically dramatically outperforms the uniform approach. We provide a theoretical explanation using Table 4. In a nutshell, a significant difference arises depending on whether the base measure of an exponential family is uniform or Zipfian.

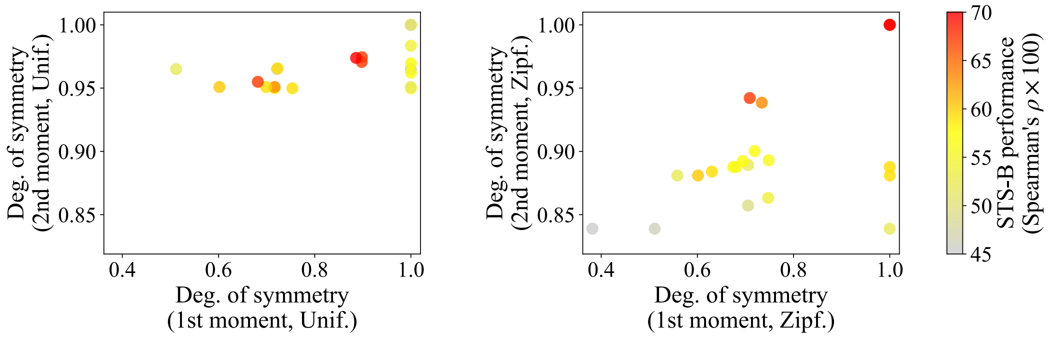

Figure 2: The relationship between the 1st-order symmetry (Def. 3, x-axis), the 2nd-order symmetry (Def. 4, y-axis), and task performance (color). Each point represents either pre-trained or post-processed word embeddings (GloVe, word2Vec, and fastText). The Zipfian measure well captures the downstream task performance (**right**), while the uniform isotropic measure cannot (**left**).

Table 3: Spearman's $\rho \times 100$ (each cell) between the symmetry scores (each column) and downstream STS-B performance (each row), on pre-trained and post-processed embeddings (GloVe, word2Vec, and fastText). The scores based on the Zipfian prior show a significantly higher correlation with task performance compared to those based on the uniform prior including Ave. Cos. and IsoScore.

	Ave. Cos. [21]	IsoScore [49]	Uniform		Zipfian	
			1st moment	2nd moment	1st moment	2nd moment
STS-B	-6.95	0.07	-21.91	-21.21	62.13	**89.55**
SICK-R	20.09	18.41	13.26	-8.71	60.04	**64.60**

4.1 Characterization through generative model, partition function, and whitening

Exponential families: Hereafter, we interpret two salient generative models from the viewpoint of *exponential families*: one given by Arora et al. [6] and the other generalizing the Levy–Goldberg formula [32, Eq. (7)]. Details of these models will be provided shortly. An exponential family is a class of probability distributions of a random variable $\boldsymbol{x}$ parametrized by a parameter $\boldsymbol{\theta}$, written in the following (canonical) form:

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \pi(\boldsymbol{x}) \exp\left(\langle \boldsymbol{x}, \boldsymbol{\theta} \rangle - \psi(\boldsymbol{\theta})\right), \quad \psi(\boldsymbol{\theta}) := \log Z(\boldsymbol{\theta}) = \log\left(\sum_{\boldsymbol{x}} \pi(\boldsymbol{x}) \exp(\langle \boldsymbol{x}, \boldsymbol{\theta} \rangle)\right), \tag{18}$$

where $\boldsymbol{x}$ is a sufficient statistic, $\boldsymbol{\theta}$ is called a natural parameter, π is the *base measure* (or "prior"), and ψ is the log-partition function. Once we specify the base measure π and the canonical pair $(\boldsymbol{x}, \boldsymbol{\theta})$, the log-partition function is determined. That being said, the base measure π is the design choice of an exponential family left for us. In the following, we specifically examine an exponential family of distributions in the form $p(w \mid c)$, where word w is predicted given context c. Specifically, the *context* represents a co-occurring word (in static word embeddings), a cloze sentence (in masked language models), or a sentence prefix (in causal language models). In all of these cases, we predict a word with the logit $\langle \boldsymbol{w}, \boldsymbol{c} \rangle$, making the exponential family a natural probabilistic model. Here, the vector $\boldsymbol{c}$ represents the vector expression of the context c, known as the "context vector." Note that, even for the same word t, the predicted word vector $\boldsymbol{w}(t)$ and the predicting context vector $\boldsymbol{c}(t)$ are distinct.

Uniform prior: Arora et al. firstly considered a log-linear generative model of word embeddings given a context (6) and demonstrated that when the generative model is adopted with normalized context vectors and a huge vocabulary, the partition function asymptotically becomes constant (8) [6, Lemma 2.1]. Here, we can regard that this model belongs to the exponential family with the uniform base measure $\pi(w) = \pi(c) = 1/|\mathcal{V}|$ [4].

[4] Although Arora et al. [6]'s generative model treats a context vector $\boldsymbol{c}$ as a model parameter drifting by a random walk, we can cast their model into an exponential family because they did not specify how the initial context vector is generated. Hence, by regarding c as an observed token with the uniform prior $\pi(c) = 1/|\mathcal{V}|$,

Table 4: Through the differences in the underlying generative models, the empirical superiority of Zipfian whitening over uniform whitening can be understood.

Generative models behind the (whitened) embeddings	
$p(w \mid c) = \dfrac{\pi(w)\exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle)}{Z(c)}, \quad Z(c) = \sum_w \pi(w)\exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle); \quad p(w, c) = p(w \mid c)\pi(c) \qquad (5)$	
with **Uniform prior**	with **Zipfian prior**
$p_{\textcircled{u}}(w \mid c) = \dfrac{1\exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle)}{Z_{\textcircled{u}}(c)}, \pi(c) \propto 1 \quad (6)$	$p_{\textcircled{z}}(w \mid c) = \dfrac{p(w)\exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle)}{Z_{\textcircled{z}}(c)}, \pi(c) = p(c) \quad (7)$
Partition functions become constant under certain conditions	
Assume $\lVert \boldsymbol{c} \rVert \equiv 1$, $\lvert\mathcal{V}\rvert \to \infty$, then $Z_{\textcircled{u}}(c) := \sum_w \exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle) = \text{const.} \quad (8)$ [6]	At the optimal solution of the corresponding loss, $Z_{\textcircled{z}}(c) := \sum_w p(w)\exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle) = \text{const.} \quad (9)$ [32]
Whitening coarsely achieves a constant partition function	
$Z_{\textcircled{u}}(c)$ $= \lvert\mathcal{V}\rvert + \utilde{\Big(\sum_w \boldsymbol{w}\Big)}^{\top} \boldsymbol{c} + \frac{1}{2}\boldsymbol{c}^{\top} \utilde{\Big(\sum_w \boldsymbol{w}\boldsymbol{w}^{\top}\Big)} \boldsymbol{c} + \dots$ $\rightsquigarrow \lvert\mathcal{V}\rvert + \utilde{\boldsymbol{0}}^{\top}\boldsymbol{c} + \frac{1}{2}\boldsymbol{c}^{\top}\utilde{\boldsymbol{I}}\boldsymbol{c} \dots \approx \text{const.} \; (10)$[39]	$Z_{\textcircled{z}}(c)$ $= \lvert\mathcal{V}\rvert + \utilde{\Big(\sum_w p(w)\boldsymbol{w}\Big)}^{\top} \boldsymbol{c} + \frac{1}{2}\boldsymbol{c}^{\top} \utilde{\Big(\sum_w p(w)\boldsymbol{w}\boldsymbol{w}^{\top}\Big)} \boldsymbol{c} + \dots$ $\rightsquigarrow \lvert\mathcal{V}\rvert + \utilde{\boldsymbol{0}}^{\top}\boldsymbol{c} + \frac{1}{2}\boldsymbol{c}^{\top}\utilde{\boldsymbol{I}}\boldsymbol{c} \dots \approx \text{const.} \; (11)$[ours]
Vector norm under generative models	
$\lVert \boldsymbol{w} \rVert_2^2 \approx 2d \log p(w) - 2Z \qquad (12)$ [6] long vector ↔ frequent (*un*informative) word	$\lVert \boldsymbol{w} \rVert_{\boldsymbol{G}(w)}^2 \approx 2\mathrm{KL}(p(\cdot) \vert\vert p(\cdot \mid w)) \quad (13)$ [42] [Thm. 1] long vector ↔ informative word
Loss and **error** corresponding to generative models $p(w \mid c)$	
softmax cross-entropy loss $\underset{(w,c)}{\mathbb{E}} - \log \dfrac{\exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle)}{Z_{\textcircled{u}}(c)} \qquad (14)$	logit-adjusted softmax cross-entropy loss $\underset{(w,c)}{\mathbb{E}} - \log \dfrac{p(w)\exp(\langle \boldsymbol{w}, \boldsymbol{c}\rangle)}{Z_{\textcircled{z}}(c)} \qquad (15)$ [36]
misclassification error $\mathbb{P}_{(w,c)}[w \notin \arg\max_{w'} \langle \boldsymbol{w}', \boldsymbol{c}\rangle] \qquad (16)$	balanced error $\frac{1}{\lvert\mathcal{V}\rvert} \sum_{w \in \mathcal{V}} \mathbb{P}_{c \mid w}[w \notin \arg\max_{w'} \langle \boldsymbol{w}', \boldsymbol{c}\rangle] \qquad (17)$ [36]

Zipfian prior: An exponential family adopted with the Zipfian measure can be written as (7). This generative model can be naturally derived from the skip-gram model with negative sampling (SGNS) [37]. By assuming that the linear model $c \mapsto \langle \boldsymbol{w}, \boldsymbol{c}\rangle$ is sufficiently capable of discriminating cooccurring words and negative samples (as in the realizable case), we can see that the generative model of the word embeddings must comply with the following formula:

$$\log \frac{p(w, c)}{p(w)p(c)} - \log k = \langle \boldsymbol{w}, \boldsymbol{c}\rangle, \tag{19}$$

where k is the number of negative samples. This optimality formula owes to Levy and Goldberg [32], and we call (19) the *Levy–Goldberg formula*. A more concise derivation is later given by Oyama et al. [42]. We can regard the Levy–Goldberg formula as an exponential family with the Zipfian base measure, $\pi(w) = p(w)$ and $\pi(c) = p(c)$, and the constant log-partition function $Z_{\textcircled{z}}(c) \equiv k^{-1}$. The generative model (7) is a relaxation of the Levy–Goldberg formula since we do not impose the realizability assumption necessary for the derivation of (19).

their model is reduced to (6). The static context prior does not contradict Arora et al. [6]'s model with sufficiently large d, where the random walk drifts extremely slowly.

What does whitening do? Mu and Viswanath [39] proposed a method to approximately make the partition function of the uniform prior model constant by centering the word vectors and removing the top principal components (10). Our Zipfian whitening corresponds to Mu and Viswanath's post-processing method, in the sense that ours and theirs make the partition function constant up to the second moment (11) and (10), respectively. In summary, Zipfian whitening (11) transforms a probabilistic model into an exponential family adopted with the Zipfian base measure (7), making it closer to the Levy–Goldberg formula (19).

4.2 Emphasis on rare words by Zipfian prior

Let us explore further why the Zipfian prior results in good performance in downstream tasks (§ 3.2). In summary, the Zipfian prior approach emphasizes low-frequency words, while the uniform prior approach emphasizes high-frequency words, both from perspectives of vector norms and errors/losses. So far in this paper, we have repeatedly discussed weighting each word according to frequency, so it may seem contradictory that Zipfian approach emphasizes low-frequency words as a result. To illustrate, let us reconsider centering. In centering, the mean vector is *subtracted* from each vector. Weighting each word vector by frequency when constructing the mean vector means that signals corresponding to high-frequency words are removed more substantially from each vector. The emphasis on low-frequency words has been repeatedly supported throughout the history of NLP and information retrieval, such as Luhn's hypothesis [34], inverse document frequency (IDF) [53], and smooth inverse frequency (SIF) [7]. For instance, it is reasonable to emphasize the word '`isotropy`' when creating a sentence embedding containing both words '`the`' and '`isotropy`'.

From the perspective of vector norm

Under the Zipfian prior model, *words with larger information content have longer (emphasized) vector representations*. Conversely, under the uniform prior model, words with smaller information content have longer (emphasized) vector representations.

As a representative example of **uniform prior** models, the norms of word vectors learned by random walk language models are theoretically and empirically proportional to word frequency (12) (see Eq. (2.4) and Fig. 2 in Arora et al. [6]). That is, in such embedding space, words with *less* information (e.g., '`the`') are emphasized. This tendency is consistently observed in dynamic language models and causal language models that adopt the softmax cross-entropy loss, another typical example of the uniform prior family [28]. By contrast, when training word embeddings with skip-gram negative sampling [37], the word embeddings follow the **Zipfian prior** family, and their norms become larger with greater information, which we show subsequently [50, 60, 42]. Based on the formulation of the exponential family and following Eq. (12) of Oyama et al. [42], we formally describe the norm properties of the word vectors obtained from the Zipfian prior model.

Theorem 1 (The norm of a word vector learned with empirical Zipfian prior models reflect the information amount of the word; a refined version of [42] Eq. (12))**.** *Assume that word embeddings $\{\boldsymbol{w}_i\}_i$ follow the Zipfian prior model (7), For the same word t, the vector $\boldsymbol{w}$ on the predicted side and the vector $\boldsymbol{c}$ on the predicting side are shared: $\boldsymbol{w}(t) \equiv \boldsymbol{c}(t)$ (weight tying), and $\sum_{t \in \mathcal{V}} p(t)\boldsymbol{w}(t) = \boldsymbol{0}$ (centered w.r.t. Zipfian prior), then each word vector $\boldsymbol{w}(t)$ satisfy*

$$\|\boldsymbol{w}(t)\|^2_{\boldsymbol{G}(t)} \approx 2\mathrm{KL}(p(\cdot)\|p(\cdot \mid t)), \quad \boldsymbol{G}(t) := \sum_{t' \in \mathcal{V}} p(t' \mid t)\boldsymbol{c}(t')\boldsymbol{c}(t')^\top, \tag{20}$$

where $\|\boldsymbol{w}\|_{\boldsymbol{A}}$ with a positive definite matrix $\boldsymbol{A}$ denote a norm based on a quadratic form $\sqrt{\boldsymbol{w}^\top \boldsymbol{A}\boldsymbol{w}}$[5]*.*
Proof. Refer to Appendix F.

In Fig. 3, we experimentally confirmed that the norms of informative words become larger with Zipfian whitening (shown from center to the right in Fig. 3), bringing them closer to the ideal Zipfian prior model[6].

[5] The matrix $\boldsymbol{G}(w)$ takes the form $\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^\top$ is indeed positive definite, similar to a covariance matrix.

[6] Given these results, some readers may be interested in the experimental outcomes for a baseline where uniform whitening is applied, followed by rescaling norms based on information content through Zipfian whitening. For these results, please refer to Appendix G.

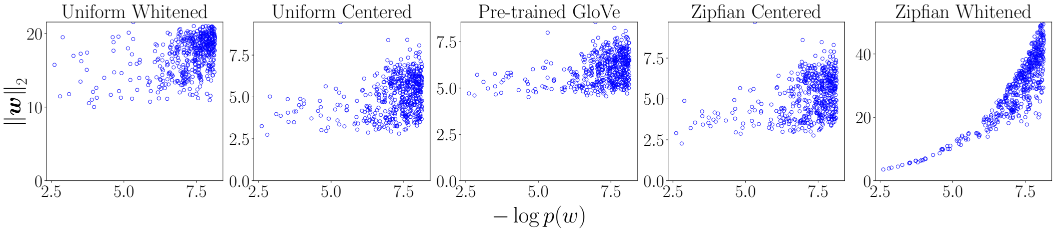

Figure 3: Relationships between the information content $-\log p(w)$ and the vector norms $\|\boldsymbol{w}\|_2$ for top 500 frequent words w. The figure in the center represents the pre-trained GloVe model. By using **Zipfian** whitening, the information content gets encoded in the norm (center to right). Conversely, with **uniform** whitening, this phenomenon does not occur (center to left).

From the perspective of error and loss

The error and loss functions associated with the Zipfian prior model emphasize low-frequency words. In contrast, the error and loss functions of the uniform prior model focus on the average loss across the entire dataset, resulting in a greater emphasis on high-frequency words.

The standard classification loss is the softmax cross-entropy loss (14). By taking its expectation over the dataset $\{(w, c)\}$, embeddings associated with higher-frequency words receive more updates because the softmax is the uniform inverse link, corresponding to the **uniform prior** model. By contrast, the logit-adjusted loss (15) has been proposed to tackle class imbalance [36]. From our viewpoint, the logit adjustment term $p(w)$ makes the inverse link belong to the **Zipfian prior** model. The softmax and logit-adjusted losses are Fisher consistent to the misclassification (16) and balanced (17) error rates, respectively. As the latter tends to stress minor classes, the logit-adjusted loss and Zipfian prior model are suitable for emphasizing low-frequency words during the learning process.

Another prominent loss function for representation learning is contrastive loss, with the SGNS loss (word2vec) [37] as a representative example in the context of word representation learning. This loss similarly uses a loss aligned with the **Zipfian prior**:

$$-\mathop{\mathbb{E}}_{(w,c)}\left[\log\sigma(\langle\boldsymbol{w},\boldsymbol{c}\rangle) + \sum_{w'_i\sim p(w)}^{i=1,\dots,k}\log\sigma(-\langle\boldsymbol{w}'_i,\boldsymbol{c}\rangle)\right], \tag{21}$$

where σ is sigmoid function, and k is the number of negative samples. Since high-frequency words are more likely to be sampled as negative examples, the loss has less impact on high-frequency words in positive examples. Consequently, low-frequency positive words are relatively emphasized in representation learning. The Levy–Goldberg formula in the previous section describes the properties of an ideally trained word2vec model, which are essentially the properties of Zipfian prior models.

5 Unified explanation of the efficacy of existing methods

Distinguishing the distribution that the base measure follows helps us understand why some existing NLP methods are effective.

5.1 Uniform whitening of token embeddings ≈ Zipfian whitening of type embeddings

Masked language models like BERT [18] and RoBERTa [33] produce dynamic (contextualized) *token* embeddings. Adding up such token embeddings of constituent tokens to create sentence embeddings often leads to poor empirical performance [46]. However, symmetrizing significantly improves their performance; such methods including "batch *centering*," "*Whitening*BERT," and contrastive learning methods [16, 46, 59, 22, 26, 57]. This improvement can also be explained from the perspective of the Zipfian prior. A dataset or corpus is first fed into the model to obtain *token* embeddings[7]. Centering/whitening is then applied to this entire set of embeddings. As this token embedding (multi)set has the multiplicity asymptotically proportional to the word frequency,

[7] Here, the computation of the additive composition $\boldsymbol{s} := 1/|s|\sum_{w\in s}\boldsymbol{w}$ can be ignored without major issues in formal discussions of spatial symmetry. This is because the words in a sentence are generated based on word

Table 5: The empirical performance difference between “uniform”—enforced centering and whitening with a uniform prior for dynamic embeddings, and “Zipfian”—conventional uniform centering and whitening over tokens with an implicit Zipfian prior over types. Each cell shows the STS-B [15] score $\times 100$. This comparison reveals that token-level uniform centering/whitening, corresponding to type-level Zipfian centering/whitening, leads to empirically better performance.

BERT-base	63.75	
	“Uniform”	“Zipfian”
+ Centering	64.04	**64.82**
+ Whitening	60.53	**64.91**

RoBERTa-base	60.75	
	“Uniform”	“Zipfian”
+ Centering	60.34	**61.30**
+ Whitening	61.31	**65.59**

this *uniform* centering/whitening of *token* embeddings corresponds to the word-frequency-weighted (*Zipfian*) centering/whitening of *type* embeddings. For a more formal description of the above explanations, please refer to Appendix H. Additionally, recent work has found that contrastive additive sentence encoders implicitly weight words by their information content [30]. This finding is consistent with the previous discussion on vector norms (§ 4.2), and can be seen as indirect evidence supporting the idea that these models belong to the Zipfian prior family.

This idea can also be supported by empirical evidence. This idea is also supported by empirical evidence. To establish a baseline for centering and whitening token embeddings under a uniform prior, we scale each embedding by the reciprocal of its type frequency, ensuring uniform treatment across types. Refer to the Appendix H for the detailed computation of this *pseudo uniform* approach and a formal explanation of how it achieves type uniformity. Table 5 shows the results. Comparing the pseudo-uniform centering/whitening (which assumes a *uniform* prior over types) with the conventional token-level uniform centering/whitening (which implicitly assumes a *Zipfian* prior over types) reveals that the latter approach based on a Zipfian prior empirically achieves better performance. Additional experimental settings and results can be found in Appendix B and Appendix I.

5.2 Headless causal language model roughly belongs to Zipfian prior family

The recently proposed headless language model [23] uses only words within the same batch to predict next tokens with a pseudo-softmax function. This method originally aimed to reduce the computational cost of the softmax function in the $|\mathcal{V}|$ direction, but an interesting side effect is the improvement in the performance. This success can also be explained from the perspective of Zipfian priors. If we repeatedly sample small batches, the sampling frequency of each word will increasingly reflect its true frequency as the batch size approaches 1.

6 Conclusion

Standard methods for adjusting and measuring symmetries in word embedding spaces—such as centering and whitening—implicitly assume *uniformly* distributed word frequencies, which is unrealistic. We hypothesize that, based on the type-token distinction, using empirical *Zipfian* word frequencies is essential when calculating the expectation (§ 2). Based on the idea and the definitions of first- and second-order symmetry in random vectors, we derived Zipfian whitening, which *enhances the symmetry* of the word embedding space. Even though it is nearly identical to standard PCA whitening, Zipfian whitening significantly outperforms existing methods (§ 3.2). Similarly, we derived a metric to *evaluate the symmetry* of word embedding spaces. Our intrinsic metrics showed a strong correlation with extrinsic task performance, even when popular metrics show almost none (§ 3.3). We then presented a framework explaining the differences in effect between whitening based on uniform and Zipfian approaches, by attributing them to differences in the base measure of the exponential family (§ 4.1). By further exploring this viewpoint through information geometry and loss functions, we showed how the Zipfian approach emphasizes the informativeness of low-frequency words (§ 4.2). Lastly, through our proposed viewpoint, we found that popular NLP methods perform well because their word embeddings end up encoding a Zipfian prior; such models include word2vec [37] (Fig. 4.2), WhiteningBERT [26] (§ 5.1), and headless language models [23] (§ 5.2).

frequency distribution, resulting in the first and second moments (Def. 1, Def. 2) of sentence vectors closely matching those of word vectors.

Acknowledgements

This work is supported by JST ACT-X Grant Number JPMJAX200S and JSPS KAKENHI Grant Number 22H05106. We received numerous constructive and valuable comments from the anonymous reviewers of NeurIPS 2024, which have significantly contributed to the quality improvements from the submission version to the camera-ready version. We would also like to thank Hayato Tsukagoshi of Nagoya University for his insightful comments on the handling of dynamic embeddings and on the experimental setup of the SimCSE paper [22], including minor discrepancies between the paper's description and its actual implementation. We also extend our gratitude to the organizers and participants of MLSS 2024, the Tohoku NLP group, the Shimodaira lab at Kyoto University, and many members of the Japanese NLP and machine learning community, for their constructive feedback and motivating encouragement throughout our research discussions.

Limitations

How these assumptions might be violated in practice

In our theoretical analysis concerning norms, and in the discussion on the relationship between whitening and normalization constants, we have proceeded by ignoring the residual terms beyond the second order. Empirically, focusing only on the first and second order has yielded significant results. However, to accurately identify cases where the proposed method might fail, a detailed theoretical and empirical examination of the asymptotic behavior of higher-order moments might be crucial. This remains an important future work.

The condition that the partition function is constant is only a necessary condition from the perspective of both the generative model's optimal solution and whitening. The true logical relationship between whitening and the generative model has not been clarified. In particular, verifying whether the projection through whitening allows us to transition between the two model families (the uniform family and the Zipfian family) is an intriguing and valuable direction for both theoretical exploration and practical application.

The scope of the empirical claims made

Our experiments primarily focused on static and dynamic word embeddings, as many of their theoretical properties have been understood and they have been central to the rise of isotropization. Admittedly, this paper also advances our understanding of causal language models. However, to make a more significant practical impact in the era of large language models, employing the proposed method as a regularization term for next-token prediction holds great promise for future work.

The experiments utilized typical downstream NLP tasks, particularly popular datasets for sentence-level semantic tasks. By scaling up the task set to include word-level tasks or leveraging a broader range of multilingual data, we can more robustly demonstrate the practical utility of the proposed framework.

The factors that influence the performance of our approach

The proposed method inherently involves numerically unstable calculations, such as multiplying by the inverse of small singular values. Embeddings for low-frequency words are often far from converged even after extensive pre-training, and the eigenvalues of the embedding space are known to decay. Given these situations, the adverse effects of small singular values are plausible. Considering recent advancements in whitening techniques, developing a more numerically stable algorithm is an important direction for future work.

Broader Impacts

Potential impacts to AI alignment Dohmatob et al. [19] reported that repeated sampling from generative AIs may shift word frequency distributions toward lighter-tailed distributions. This may reduce linguistic diversity and lead to cultural homogenization by diminishing region-specific or culturally unique expressions. Our Zipfian whitening and similar regularization methods could enhance output diversity, enriching the linguistic landscape.

Potential negative societal impacts The sentence similarity tasks used in our evaluation experiments are now considered core technologies for RAG (retrieval-augmented generation), which is essential when large language models leverage external resources. If chatbots generate responses tailored to user ideologies or preferred information sources, it may result in negative societal impacts, including political agitation.

References

[1] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In E. Agirre, J. Bos, M. Diab, S. Manandhar, Y. Marton, and D. Yuret, editors, **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1051.

[2] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. *SEM 2013 shared task: Semantic Textual Similarity. In M. Diab, T. Baldwin, and M. Baroni, editors, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/S13-1004.

[3] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In P. Nakov and T. Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, Aug. 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2010. URL https://aclanthology.org/S14-2010.

[4] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, L. Uria, and J. Wiebe. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In P. Nakov, T. Zesch, D. Cer, and D. Jurgens, editors, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.18653/v1/S15-2045. URL https://aclanthology.org/S15-2045.

[5] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In S. Bethard, M. Carpuat, D. Cer, D. Jurgens, P. Nakov, and T. Zesch, editors, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1081. URL https://aclanthology.org/S16-1081.

[6] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. A Latent Variable Model Approach to PMI-based Word Embeddings. *TACL*, 4:385–399, 12 2016. ISSN 2307-387X. doi: 10.1162/tacl{_}a{_}00106. URL https://aclweb.org/anthology/papers/Q/Q16/Q16-1028/.

[7] S. Arora, Y. Liang, and T. Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *ICLR*, 2017. URL https://openreview.net/forum?id=SyK00v5xx.

[8] A. Bakarov. A survey of word embeddings evaluation methods. *arXiv [cs.CL]*, Jan. 2018.

[9] S. Bird and E. Loper. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, 7 2004. URL https://www.aclweb.org/anthology/P04-3031.

[10] D. Biś, M. Podkorytov, and X. Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. pages 5117–5130, June 2021.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *TACL*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051. URL https://www.aclweb.org/anthology/papers/Q/Q17/Q17-1010/.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In *NeurIPS*, volume 33, pages 1877–1901, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[13] E. Bruni, G. Boleda, M. Baroni, and N.-K. Tran. Distributional Semantics in Technicolor. In *ACL*, pages 136–145, 7 2012. URL https://www.aclweb.org/anthology/P12-1015.

[14] L. L. Campbell. Minimum coefficient rate for stationary random processes. *Information and Control*, 3(4): 360–371, Dec. 1960.

[15] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *SemEval*, pages 1–14, 8 2017. doi: 10.18653/v1/S17-2001. URL https://www.aclweb.org/anthology/papers/S/S17/S17-2001/.

[16] X. Chen, N. Ding, T. Levinboim, and R. Soricut. Improving Text Generation Evaluation with Batch Centering and Tempered Word Mover Distance. In *First Workshop on Evaluation and Comparison of NLP Systems*, pages 51–59, Online, 11 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.eval4nlp-1.6. URL https://www.aclweb.org/anthology/2020.eval4nlp-1.6.

[17] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Trans. Inf. Theory*, 38(2):713–718, Mar. 1992.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, pages 4171–4186, 2019. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423/.

[19] E. Dohmatob, Y. Feng, P. Yang, F. Charton, and J. Kempe. A tale of tails: Model collapse as a change of scaling laws. *arXiv [cs.LG]*, Feb. 2024.

[20] K. Ethayarajh. Unsupervised Random Walk Sentence Embeddings: A Strong but Simple Baseline. In *Rep4NLP*, pages 91–100, 7 2018. URL https://aclweb.org/anthology/papers/W/W18/W18-3012/.

[21] K. Ethayarajh. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *EMNLP*, pages 55–65, 2019. URL http://arxiv.org/abs/1909.00512.

[22] T. Gao, X. Yao, and D. Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[23] N. Godey, É. V. de La Clergerie, and B. Sagot. Headless language models: Learning without predicting with contrastive weight tying. *arXiv [cs.CL]*, Sept. 2023.

[24] C. Gong, D. He, X. Tan, T. Qin, L. Wang, and T.-Y. Liu. FRAGE: Frequency-agnostic word representation. *Adv. Neural Inf. Process. Syst.*, 31, 2018.

[25] P. He, X. Liu, J. Gao, and W. Chen. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *ICLR*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

[26] J. Huang, D. Tang, W. Zhong, S. Lu, L. Shou, M. Gong, D. Jiang, and N. Duan. WhiteningBERT: An easy unsupervised sentence embedding approach. In M.-F. Moens, X. Huang, L. Specia, and S. W.-T. Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[27] R. Kannan, L. Lovász, and M. Simonovits. Random walks and an o*(n5) volume algorithm for convex bodies. *Random Struct. Algorithms*, 11(1):1–50, Aug. 1997.

[28] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. Transformer language models handle word frequency in prediction head. pages 4523–4535, July 2023.

[29] K. Kurihara, D. Kawahara, and T. Shibata. JGLUE: Japanese general language understanding evaluation. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, editors, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.317.

[30] H. Kurita, G. Kobayashi, S. Yokoi, and K. Inui. Contrastive learning-based sentence encoders implicitly weight informative words. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10932–10947, Singapore, Dec. 2023. Association for Computational Linguistics.

[31] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, Eytan Ruppin. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst. Secur.*, 20(1):116–131, Jan. 2002.

[32] O. Levy and Y. Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *NIPS*, pages 2177–2185, 2014. URL http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization.

[33] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv [cs.CL]*, July 2019.

[34] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, Apr. 1958.

[35] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *LREC*, pages 216–223, Reykjavik, Iceland, 5 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf.

[36] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar. Long-tail learning via logit adjustment. Sept. 2020.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *NIPS*, pages 3111–3119, 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

[38] D. Mimno and L. Thompson. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2873–2878, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[39] J. Mu and P. Viswanath. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. In *ICLR*, 2018. URL https://openreview.net/forum?id=HkuGJ3kCb.

[40] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. MTEB: Massive text embedding benchmark. In *EACL*, pages 2014–2037, Dubrovnik, Croatia, 5 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148.

[41] K. Oono and T. Suzuki. Graph neural networks exponentially lose expressive power for node classification. Sept. 2019.

[42] M. Oyama, S. Yokoi, and H. Shimodaira. Norm of word embedding encodes information gain. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2108–2130, Singapore, Dec. 2023. Association for Computational Linguistics.

[43] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *EMNLP*, pages 1532–1543, 2014. doi: 10.3115/v1/D14-1162. URL https://aclweb.org/anthology/papers/D/D14/D14-1162/.

[44] I. Provilkov, D. Emelianenko, and E. Voita. BPE-dropout: Simple and effective subword regularization. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online, July 2020. Association for Computational Linguistics.

[45] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[46] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP*, pages 3980–3990, 11 2019. doi: 10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410.

[47] O. Roy and M. Vetterli. The effective rank: A measure of effective dimensionality. pages 606–610, Sept. 2007.

[48] M. Rudelson. Random Vectors in the Isotropic Position. *Journal of Functional Analysis*, 164(1):60–72, 1999. ISSN 0022-1236. doi: https://doi.org/10.1006/jfan.1998.3384. URL http://www.sciencedirect.com/science/article/pii/S0022123698933845.

[49] W. Rudman, N. Gillman, T. Rayne, and C. Eickhoff. IsoScore: Measuring the uniformity of embedding space utilization. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3325–3339, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[50] A. M. J. Schakel and B. J. Wilson. Measuring word significance using distributed representations of words. *arXiv [cs.CL]*, Aug. 2015.

[51] H. Shi, J. Gao, H. Xu, X. Liang, Z. Li, L. Kong, S. M. S. Lee, and J. Kwok. Revisiting over-smoothing in BERT from the perspective of graph. Sept. 2021.

[52] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[53] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. In *Document retrieval systems*, pages 132–142. Taylor Graham Publishing, GBR, Dec. 1988.

[54] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971v1*, 2023. URL https://arxiv.org/abs/2302.13971v1.

[55] R. Vershynin. *High-Dimensional Probability*. Cambridge University Press, 2018.

[56] L. Wang, J. Huang, K. Huang, Z. Hu, G. Wang, and Q. Gu. Improving Neural Language Generation with Spectrum Control. In *ICLR*, 2020. URL https://openreview.net/forum?id=ByxY8CNtvr.

[57] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv:2212.03533v2*, 2024. URL https://arxiv.org/abs/2212.03533v2.

[58] L. Wetzel. Types and tokens. *The Stanford Encyclopedia of Philosophy (Fall 2018 Edition)*, Apr. 2006.

[59] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *ACL-IJCNLP*, pages 5065–5075, Online, 8 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.393. URL https://aclanthology.org/2021.acl-long.393.

[60] S. Yokoi, R. Takahashi, R. Akama, J. Suzuki, and K. Inui. Word Rotator's Distance. In *EMNLP*, pages 2944–2960, Online, 11 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.236. URL https://www.aclweb.org/anthology/2020.emnlp-main.236.

A Explanation fo the Zipfian whitened word vectors will have a zero mean and be in an isotropic position in terms of expectation

The second step of PCA whitening involves decorrelating the dimensions and then normalizing them, which is achieved by transforming the centered random vector $\overline{\boldsymbol{w}}$ as follows:

$$\widetilde{\boldsymbol{w}} := \mathbb{E}[\overline{\boldsymbol{w}}\,\overline{\boldsymbol{w}}^\top]^{-1/2}\overline{\boldsymbol{w}}. \tag{22}$$

Actually, $\mathbb{E}[\widetilde{\boldsymbol{w}}\widetilde{\boldsymbol{w}}^\top] = \boldsymbol{I}_d$ holds; $\widetilde{\boldsymbol{w}}$ satisfies Def. 2. Computationally, the estimation of $\mathbb{E}[\overline{\boldsymbol{w}}\,\overline{\boldsymbol{w}}^\top]^{-1/2}$ can be performed efficiently via singular value decomposition (SVD) of the centered "data" matrix $\overline{\boldsymbol{W}} := [\overline{\boldsymbol{w}}_1^\top, \dots, \overline{\boldsymbol{w}}_n^\top]^\top$. Note again that, this standard method assumes that the frequency of each word (i.e., each row) is uniform, which presents the issues discussed in § 2. To account for word frequency, SVD should be performed on the matrix

$$\overline{\boldsymbol{W}}_p := \left[\sqrt{p(w_1)}\overline{\boldsymbol{w}}_1^\top, \dots, \sqrt{p(w_{|\mathcal{V}|})}\overline{\boldsymbol{w}}_{|\mathcal{V}|}^\top\right]^\top \in \mathbb{R}^{|\mathcal{V}|\times d}, \tag{23}$$

where each word frequency is multiplied by its *square root*. In fact, $\overline{\boldsymbol{W}}_p^\top\overline{\boldsymbol{W}}_p$ serves as an estimator for $\mathop{\mathbb{E}}_{\boldsymbol{w}\sim p}[\overline{\boldsymbol{w}}\,\overline{\boldsymbol{w}}^\top]$. This can be confirmed by comparing the (j,k)'th elements of each matrix (or matrix-valued random variable):

$$\mathop{\mathbb{E}}_{\boldsymbol{w}\sim p}[\overline{\boldsymbol{w}}\,\overline{\boldsymbol{w}}^\top][j,k] = \mathop{\mathbb{E}}_{\boldsymbol{w}\sim p}[\overline{\boldsymbol{w}}[j]\,\overline{\boldsymbol{w}}[k]], \tag{24}$$

$$(\overline{\boldsymbol{W}}_p^\top\overline{\boldsymbol{W}}_p)[j,k] = \sum_i p(w_i)\overline{\boldsymbol{w}}_i[j]\,\overline{\boldsymbol{w}}_i[k], \tag{25}$$

where $\boldsymbol{A}[j,k]$ denotes the (j,k)'th element of $\boldsymbol{A}$, and $\boldsymbol{v}[j]$ denotes the j'th element of $\boldsymbol{v}$. Finally, the estimator for the desired $\mathbb{E}[\overline{\boldsymbol{w}}\,\overline{\boldsymbol{w}}^\top]^{-1/2}$ can be computed as

$$\overline{\boldsymbol{W}}_p = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \quad \text{(via SVD)} \tag{26}$$

$$\mathop{\widehat{\mathbb{E}}}_{\boldsymbol{w}\sim p}[\overline{\boldsymbol{w}}\,\overline{\boldsymbol{w}}^\top]^{-1/2} = (\overline{\boldsymbol{W}}_p^\top\overline{\boldsymbol{W}}_p)^{-1/2} = ((\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top))^{-1/2} = (\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top)^{-1/2} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}. \tag{27}$$

Table 6 shows the correspondence between uniform (normal) whitening and Zipfian whitening. This may be useful for readers familiar with matrix notation.

B Experimental settings

To ensure the reproducibility of the experiments conducted in this paper, we provide detailed configurations below. Additionally, the source code has been made publicly available at https://github.com/cl-tohoku/zipfian-whitening.

B.1 Word embeddings

For static embeddings, we used the most standard ones, 300-dim **GloVe**[43] model trained on Common Crawl[8], 300-dim **word2vec**[37] model trained on Google News[9], and 300-dim **fastText**[11] subword/non-subword models trained on Common Crawl[10]. For the multilingual experiment, we used **fastText-ja** [11], a fastText model trained on Japanese Wikipedia and Common Crawl [11].

[8] https://huggingface.co/sentence-transformers/average_word_embeddings_glove.6B.300d from Sentence-Transformer's implementation [46].

[9] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/ from https://code.google.com/archive/p/word2vec/

[10] https://fasttext.cc/docs/en/english-vectors.html

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

Table 6: The correspondence between uniform (normal) whitening and Zipfian whitening.

	Uniform whitening for general table data	Zipfian whitening for word embeddings	
data matrix, original vectors	$\boldsymbol{X} = \begin{bmatrix} \vdots \\ \boldsymbol{x}_i \\ \vdots \end{bmatrix}$	$\boldsymbol{W} = \begin{bmatrix} \vdots \\ \boldsymbol{w}_i \\ \vdots \end{bmatrix}$	$\in \mathbb{R}^{n \times d}$
probability	$\boldsymbol{x} \sim$ Uniform dist. $p(\boldsymbol{x}_i) = \frac{1}{n}$	$\boldsymbol{w} \sim$ Power-law dist. $p(\boldsymbol{w}_i) = p(w_i)$ (word freq.)	
mean vector	$\widehat{\boldsymbol{\mu}}_x = \sum_i \frac{1}{n} \boldsymbol{x}_i$	$\widehat{\boldsymbol{\mu}}_w = \sum_i p(w_i) \boldsymbol{w}_i$	$\in \mathbb{R}^{d}$
centered matrix, centered vectors	$\overline{\boldsymbol{X}} = \begin{bmatrix} \vdots \\ \overline{\boldsymbol{x}}_i = \boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_x \\ \vdots \end{bmatrix}$	$\overline{\boldsymbol{W}} = \begin{bmatrix} \vdots \\ \overline{\boldsymbol{w}}_i = \boldsymbol{w}_i - \widehat{\boldsymbol{\mu}}_w \\ \vdots \end{bmatrix}$	$\in \mathbb{R}^{n \times d}$
centered matrix to create cov. mat.	$\overline{\boldsymbol{X}}_p = \begin{bmatrix} \vdots \\ \sqrt{1/(n-1)}\, \overline{\boldsymbol{x}}_i \\ \vdots \end{bmatrix}$	$\overline{\boldsymbol{W}}_p = \begin{bmatrix} \vdots \\ \sqrt{p(w_i)}\, \overline{\boldsymbol{w}}_i \\ \vdots \end{bmatrix}$	$\in \mathbb{R}^{n \times d}$
covariance matrix	$\widehat{\boldsymbol{S}} = \overline{\boldsymbol{X}}_p^\top \overline{\boldsymbol{X}}_p$ $= \frac{\overline{\boldsymbol{X}}^\top \overline{\boldsymbol{X}}}{n-1}$	$\widehat{\boldsymbol{S}} = \overline{\boldsymbol{W}}_p^\top \overline{\boldsymbol{W}}_p$	$\in \mathbb{R}^{d \times d}$
SVD of centered matrix = eigendecomposition of covariance matrix	$\overline{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_d)$ $\widehat{\boldsymbol{S}} = \frac{\overline{\boldsymbol{X}}^\top \overline{\boldsymbol{X}}}{n-1} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$ $:= \frac{\boldsymbol{\Sigma}^2}{n-1} = \operatorname{diag}\left(\frac{\sigma_1^2}{n-1}, \dots\right)$	$\overline{\boldsymbol{W}}_p = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_d)$ $\widehat{\boldsymbol{S}} = \overline{\boldsymbol{W}}_p^\top \overline{\boldsymbol{W}}_p = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$ $:= \boldsymbol{\Sigma}^2 = \operatorname{diag}\left(\sigma_1^2, \dots\right)$	
	$\boldsymbol{\Lambda}^{-1/2} := \operatorname{diag}\left(\frac{1}{\sqrt{\lambda_1}}, \dots\right)$ $= \operatorname{diag}\left(\sqrt{\frac{n-1}{\sigma_1^2}}, \dots\right)$	$\boldsymbol{\Lambda}^{-1/2} := \operatorname{diag}\left(\frac{1}{\sqrt{\lambda_1}}, \dots\right)$ $= \operatorname{diag}\left(\frac{1}{\sigma_1}, \dots\right)$	
whitened matrix	$\widetilde{\boldsymbol{X}} = \overline{\boldsymbol{X}}\boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}$	$\widetilde{\boldsymbol{W}} = \overline{\boldsymbol{W}}\boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}$	
whitened vector	$= \begin{bmatrix} \vdots \\ \widetilde{\boldsymbol{x}}_i = \overline{\boldsymbol{x}}_i \boldsymbol{V}\boldsymbol{\Lambda}^{-1/2} \\ \vdots \end{bmatrix}$	$= \begin{bmatrix} \vdots \\ \widetilde{\boldsymbol{w}}_i = \overline{\boldsymbol{w}}_i \boldsymbol{V}\boldsymbol{\Lambda}^{-1/2} \\ \vdots \end{bmatrix}$	

For dynamic embeddings, we used three most standard masked language models, **BERT**[18][12], **RoBERTa**[33][13], and **DeBERTa** [25][14]. All three models are base size. To aggregate the dynamic word embeddings to create sentence embeddings, we follow the first-last average pooling from the prior work [22]. In this setting, we first average the hidden states of first and last dynamic layer of

[12] `https://huggingface.co/google-bert/bert-base-uncased`

[13] `https://huggingface.co/FacebookAI/roberta-base`

[14] `https://huggingface.co/microsoft/deberta-base`

the model to get the averaged token embeddings[15], then average the token embeddings to get final sentence embeddings[16].

B.2 Empirical word frequency and vocabulary

As the empirical word probability $p(w)$ of English words, we used the **enwiki** dataset preprocessed by Arora et al. [7][17]. For the Japanese word probability, we used Japanese Wikipedia word frequency from Wiktionary, denoted as **jawiki** [18]. Furthermore, we also used the frequency of words in the evaluation data itself (**test set probability**). The word frequency in the test set is implicitly utilized in [7]'s sentence embedding method and is also a natural approach in the context of "covariate shift" [52].

As vocabulary $\mathcal{V}$, we used the overlapping entries between the word frequency list and the pre-trained word embedding model's vocabulary across all settings, including baseline methods.

B.3 Baseline methods

As baselines for post-processing of word vectors, we used **ABTT** (all-but-the-top) [39], which established the trend of post-processing word vectors; and the strong baseline method by [7], the combination of **SIF** (smoothed inverse frequency) weighting and **CCR** (common component removal)[19]. We followed the hyperparameter choices of the original papers, with the dimensionality reduction parameter for ABTT set to $D \coloneqq 3$, and the weighting parameter for SIF set to $a \coloneqq 10^{-3}$.

B.4 Extrinsic tasks

As downstream tasks, we used the most commonly utilized ones in the community, **STS12-16** [1, 2, 3, 4, 5], **STS-B** [15] and **SICK-R** [35]. For the multilingual experiments, we used **JSTS** (Japanese version of the STS) from JGLUE benchmark [29]. They are *sentence*-level similarity tasks and are standard for empirically evaluating the performance of *word* vectors[20] [21]. These datasets consist of pairs of sentences and their semantic similarity rated by annotators. We first tokenized the dataset

[15] Note that, we apply centering/whitening operations to such *token* embeddings, not to the final *sentence* embeddings, in order to match the setting in the theoretical analysis and the static word embedding experiments.

[16] Though we followed the experimental setting from the prior work [22], there is a slight discrepancy in the experimental results of the baseline setting. We found that this was due to prior work inadvertently taking the average of the hidden states of the zero-th layer (i.e., *static* word embedding layer) and the final dynamic layer. See the discussion at `https://github.com/princeton-nlp/SimCSE/issues/285` for more details.

[17] `https://github.com/PrincetonML/SIF/raw/master/auxiliary_data/enwiki_vocab_min200.txt`

[18] `https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/Japanese2015_10000` and `https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/Japanese2015_10001-20000`

[19] CCR is a process applied to *sentence* vectors, but due to its linearity, it can be adapted to *word* vectors. For more details, please refer to Yokoi et al. [60].

[20] For those outside the field of NLP research or practice, the question, "Why not run word-level evaluation metrics?" is a natural and valid one. Our language has a property known as compositionality, allowing infinite semantic content to be conveyed through a finite vocabulary as building blocks. This principle underlies models like word2vec [37], BERT [18], and the GPT series [12], where the fundamental unit of representation is the word; these models are then applied to solve tasks with larger components, such as sentences. Our research adheres to this foundational principle of NLP. Also, existing word-level similarity datasets have significant issues that make them less suitable for our work (see Bakarov [8, Section 4.1.1]). Given that whitening reflects word information content in vector norms, classic tasks like keyword extraction—which selects words with high information content—could be good candidates; results from a prior study using methods similar to ours would also be informative [42, Section 7.1].

[21] Setting aside the criticisms from previous studies for now, we conducted an evaluation using the two most well-known lexical similarity datasets. Table 7 shows the results. We found that the process of raw → Zipfian centering → Zipfian whitening consistently improves lexical properties. However, note that the finding "*direction: uniform whitening* $>$ *direction: Zipfian whitening*" contradicts the experimental results in Appendix G, which showed "*direction: uniform whitening*, norm: Zipfian whitening $<$ *direction: Zipfian whitening*, norm: Zipfian whitening." Here, lexical similarity tasks rely solely on vector direction and do not reference vector norms, as only the cosine similarity between word vectors is used to predict similarity. This discrepancy likely arises because these datasets are not representative of natural language, as discussed in Bakarov [8, Section 4.1.1]. For example, the widely used dataset WordSim353 [31] includes only about 200 subjective ratings on

by NLTK [9] with some post-processing following [20][22], then lowercased all tokens. The typical experimental protocol we followed is to sum the word vectors to form a "sentence vector" and then check if the angles (cosine similarity) between them correlate well with the gold scores. We reported Spearman's rank correlation between the predictions (cosine scores) and human-annotated gold scores[23].

B.5 Computational resources for experiments

We conducted all experiments using a single NVIDIA RTX 6000 Ada GPU with 48GB VRAM. Each STS task required 10 seconds per model and whitening method, totaling approximately 10 minutes for the entire experiment, excluding the embedding loading time to the GPU.

For the calculation of the symmetry scores, each setting took one minute, resulting in a total of 5 minutes, again excluding the embedding loading time and the average cosine similarity (Ave. Cos.) setting. The Ave. Cos. score computation took 10 minutes per model, totaling 20 minutes for the two models.

C Experimental results on all benchmark datasets to evaluate the effects of Zipfian whitening

In § 3.2, we evaluated the empirical performance of Zipfian whitening on the STS-B dataset. In this section, we present experimental results using more comprehensive datasets. Detailed experimental settings can be found in Appendix B. Table 8, Table 9 and Table 10 show the results. Across all datasets, the method incorporating a Zipfian prior consistently outperforms the method employing a uniform prior.

D Proof of Prop. 1

Proof. We will show the following.

$$\mathbb{E}[(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])^\top] \propto \boldsymbol{I}_d \tag{28}$$

$$\overset{\textcircled{1}}{\iff} \lambda_1 = \lambda_2 = \cdots = \lambda_d \tag{29}$$

$$\overset{\textcircled{2}}{\iff} \mathrm{Sym}_2(\boldsymbol{v}) = \frac{1}{\log d} H\left(\frac{\lambda_1}{\sum_j \lambda_j}, \ldots, \frac{\lambda_d}{\sum_j \lambda_j}\right) = 1 \tag{30}$$

($\overset{\textcircled{1}}{\Longrightarrow}$) When $\mathbb{E}[(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])^\top] = k\boldsymbol{I}_d$ holds, its eigenvalues are $\{k, \ldots, k\}$.

($\overset{\textcircled{1}}{\Longleftarrow}$) Since $\mathbb{E}[(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])^\top]$ is symmetric positive definite, it can be represented as $\boldsymbol{U\Lambda U}^\top$ using a diagonal matrix with eigenvalues $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d) \in \mathbb{R}^{d \times d}$ and an orthogonal

Table 7: Each cell shows the correlation coefficients ×100 between the cosine similarity of (corrected) GloVe embeddings and the human-annotated gold score on lexical similarity tasks.

WordSim353 [31]		
GloVe	78.70	
	Uniform	Zipfian
+ Centering	75.39	**79.66**
+ Whitening	**82.31**	80.90

MEN [13]		
GloVe	80.49	
	Uniform	Zipfian
+ Centering	78.07	**80.55**
+ Whitening	**84.35**	83.97

common nouns, such as (tiger, cat, 7.35) or (king, cabbage, 0.23), which may or may not co-occur in the same document.

[22] https://github.com/kawine/usif/blob/71ffef5b6d7295c36354136bfc6728a10bd25d32/usif.py#L113-L126

[23] We used the MTEB [40] implementation: https://github.com/embeddings-benchmark/mteb, for the evaluation of the static word embeddings in Table 2,Table 8, and Table 9. For the evaluation of the dynamic word embeddings in Table 5 and Table 12, we used the implementation in SimCSE paper [22]: https://github.com/princeton-nlp/SimCSE, to match the experimental setting.

matrix $\boldsymbol{U}$. Now we have $\lambda_1 = \lambda_2 = \dots \lambda_d =: k$, then $\mathbb{E}[(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])^\top] = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top = \boldsymbol{U}k\boldsymbol{I}_d\boldsymbol{U}^\top = k\boldsymbol{I}_d$.

($\overset{\textcircled{2}}{\iff}$) The Shannon entropy $H(p)$ of a random variable p taking d possible values attains its maximum value $\log d$ if and only if p follows uniform distribution. □

E Pseudocode for the evaluation metrics of symmetry

See Algorithm 2 to measure the degree of symmetry of word embeddings.

Algorithm 2 Measure the degree of symmetry of word embeddings

```
Input: Word embeddings $\{\boldsymbol{w}_i\}$, word frequency $p\colon \mathcal{V} \to [0, 1]$.
Output: Degree of centrality (the 1st moment) $\widehat{\mathrm{Sym}}_1(\{\boldsymbol{w}_i\}, p)$ and isotropy (the 2nd moment) $\widehat{\mathrm{Sym}}_2(\{\boldsymbol{w}_i\}, p)$.
    Measure the degree of centrality (the 1st moment of symmetry):
    $\widehat{\boldsymbol{\mu}} \leftarrow \sum_{w_i \in \mathcal{V}} p(w_i)\boldsymbol{w}_i \in \mathbb{R}^d$
    $\ell \leftarrow \sum_{w_i \in \mathcal{V}} p(w_i)\|\boldsymbol{w}_i\| \in \mathbb{R}$
    $\widehat{\mathrm{Sym}}_1(\{\boldsymbol{w}_i\}, p) \leftarrow 1 - \|\widehat{\boldsymbol{\mu}}\|/\ell$
    Measure the degree of isotropy (the 2nd moment of symmetry):
    $\boldsymbol{W} \leftarrow \left[\sqrt{p(w_1)}(\boldsymbol{w}_1 - \widehat{\boldsymbol{\mu}})^\top, \dots, \sqrt{p(w_{|\mathcal{V}|})}(\boldsymbol{w}_{|\mathcal{V}|} - \widehat{\boldsymbol{\mu}})^\top\right]^\top$
    $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \leftarrow \mathrm{SVD}(\boldsymbol{\mathcal{W}})$
        ▷ $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \dots, \sigma_d) \in \mathbb{R}^{d \times d}$ consists the singular values of $\boldsymbol{W}$.
    $(\lambda_1, \dots, \lambda_d) \leftarrow (\sigma_1^2, \dots, \sigma_d^2)$
    $\widehat{\mathrm{Sym}}_2(\{\boldsymbol{w}_i\}, p) \leftarrow -\frac{1}{\log d} \sum_i \frac{\lambda_i}{\sum_i \lambda_i} \log \frac{\lambda_i}{\sum_i \lambda_i}$
```

F Proof of Thm. 1

Proof. By the assumption, the word and context vectors for the same word $t \in \mathcal{V}$, $\boldsymbol{w}(t)$ and $\boldsymbol{c}(t)$, are obtained through the linear embedding layer with weight tying, namely, $\boldsymbol{w}(t) = \boldsymbol{U}\boldsymbol{1}_t$ and $\boldsymbol{c}(t) = \boldsymbol{U}\boldsymbol{1}_t$, where $\boldsymbol{U} \in \mathbb{R}^{d \times |\mathcal{V}|}$ is the embedding matrix and $\boldsymbol{1}_t \in \mathbb{R}^{|\mathcal{V}|}$ is the one-hot vector indicating the token t. To derive the KL divergence for the model $p(c \mid w)$, we need to begin with the generative model $p(w \mid c)$ (7) and confirm that $p(c \mid w)$ belongs to an exponential family.

$$p(c \mid w) = \frac{p(w \mid c)p(c)}{p(w)} = \frac{\exp(\langle \boldsymbol{w}, \boldsymbol{c} \rangle)}{Z_{\textcircled{z}}(c)} p(c) = \frac{p(c)\exp(\langle \boldsymbol{U}^\top \boldsymbol{w}, \boldsymbol{1}_c \rangle)}{Z_{\textcircled{z}}},$$

where we used the constancy of the partition function ($\equiv Z_{\textcircled{z}}$) from (9) at the last identity. Hence, $p(c \mid w)$ is an exponential family parametrized by $\boldsymbol{U}^\top \boldsymbol{w} (:= \boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{V}|})$, and its log-partition function is given as follows:

$$\psi_{c|w}(\boldsymbol{\theta}) = \log\left(\sum_{c \in \mathcal{V}} p(c)\exp(\langle \boldsymbol{\theta}, \boldsymbol{1}_c \rangle)\right).$$

The second-order expansion of the KL divergence can be derived based on the second moment of $p(c \mid w)$, which is given by the Hessian of $\psi_{c|w}$ in the case of exponential families. First, let us derive the first moment.

$$\frac{\partial \psi_{c|w}}{\partial \boldsymbol{\theta}} = \frac{\sum_c \frac{p(c)}{Z_{\textcircled{z}}} \exp(\langle \boldsymbol{\theta}, \mathbf{1}_c \rangle) \mathbf{1}_c}{\sum_{c'} \frac{p(c')}{Z_{\textcircled{z}}} \exp(\langle \boldsymbol{\theta}, \mathbf{1}_{c'} \rangle)} = \sum_{c \in \mathcal{V}} \frac{p(c) \exp(\langle \boldsymbol{\theta}, \mathbf{1}_c \rangle)}{Z_{\textcircled{z}}} \mathbf{1}_c = \sum_{c \in \mathcal{V}} p(c \mid w) \mathbf{1}_c = \begin{bmatrix} \vdots \\ p(c \mid w) \\ \vdots \end{bmatrix}.$$

Then, the second moment is derived.

$$\begin{aligned} \frac{\partial \psi_{c|w}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} &= \frac{1}{Z_{\textcircled{z}}} \sum_{c \in \mathcal{V}} p(c) \left\{ \frac{\partial}{\partial \boldsymbol{\theta}} \exp(\langle \boldsymbol{\theta}, \mathbf{1}_c \rangle) \right\} \mathbf{1}_c^\top = \frac{1}{Z_{\textcircled{z}}} \sum_{c \in \mathcal{V}} p(c) \exp(\langle \boldsymbol{\theta}, \mathbf{1}_c \rangle) \mathbf{1}_c \mathbf{1}_c^\top \\ &= \sum_{c \in \mathcal{V}} p(c \mid w) \mathbf{1}_c \mathbf{1}_c^\top = \operatorname{diag}[\dots\ p(c \mid w)\ \dots], \end{aligned}$$

which is the $|\mathcal{V}| \times |\mathcal{V}|$ diagonal matrix with $p(c \mid w)$ being the (c, c)-th diagonal entry. Now, we are ready to derive the KL divergence. For two tokens $w, w' \in |\mathcal{V}|$, if we write $\boldsymbol{\theta} := \boldsymbol{U} \mathbf{1}_w$ and $\boldsymbol{\theta}' := \boldsymbol{U} \mathbf{1}_{w'}$, the KL divergence of the exponential family can be expanded as the following quadratic form in their parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$:

$$\begin{aligned} 2\mathrm{KL}(p(\cdot \mid w') \| p(\cdot \mid w)) &\approx (\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \left(\frac{\partial \psi_{c|w}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right) (\boldsymbol{\theta}' - \boldsymbol{\theta}) \\ &= (\boldsymbol{w}' - \boldsymbol{w})^\top \boldsymbol{U} \left(\frac{\partial \psi_{c|w}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \right) \boldsymbol{U}^\top (\boldsymbol{w}' - \boldsymbol{w}) \\ &= (\boldsymbol{w}' - \boldsymbol{w})^\top \left\{ \sum_{c \in |\mathcal{V}|} p(c \mid w) (\boldsymbol{U} \mathbf{1}_c) (\boldsymbol{U} \mathbf{1}_c)^\top \right\} (\boldsymbol{w}' - \boldsymbol{w}) \\ &= (\boldsymbol{w}' - \boldsymbol{w})^\top \left\{ \sum_{c \in \mathcal{V}} p(c \mid w) \boldsymbol{c} \boldsymbol{c}^\top \right\} (\boldsymbol{w}' - \boldsymbol{w}) \\ &= (\boldsymbol{w}' - \boldsymbol{w})^\top \boldsymbol{G}(w) (\boldsymbol{w}' - \boldsymbol{w}) \\ &= \| \boldsymbol{w}' - \boldsymbol{w} \|_{\boldsymbol{G}(w)}^2. \end{aligned}$$

We can consider a word w_0 such that $p(\cdot) = p(\cdot \mid w_0)$, that is, an uninformative word w_0 whose presence does not change the marginal distribution at all. Noting from Equation (22) of Oyama et al. [42] that $\overline{\boldsymbol{w}} := \sum_{w \in \mathcal{V}} p(w) \boldsymbol{w} \approx \boldsymbol{w}_0$, we have

$$\begin{aligned} \mathrm{KL}(p(\cdot) \| p(\cdot \mid w)) &= \mathrm{KL}(p(\cdot \mid w_0) \| p(\cdot \mid w)) && (31) \\ &= \| \boldsymbol{w}_0 - \boldsymbol{w} \|_{\boldsymbol{G}(w)}^2 && (32) \\ &\approx \| \overline{\boldsymbol{w}} - \boldsymbol{w} \|_{\boldsymbol{G}(w)}^2 && (33) \\ &\overset{\text{Assump.}}{=} \| \boldsymbol{w} \|_{\boldsymbol{G}(w)}^2. && (34) \end{aligned}$$

□

G Experiments with a mix of uniform and Zipfian settings

Based on the findings that Zipfian whitening positively impacts word vector norms (§ 4.2), we present experimental results for a baseline: first, uniform whitening is applied, followed by rescaling norms according to information content through Zipfian whitening.

Table 11 presents the results, with the basic settings identical to those in Table 2, but uses Pearson's r as the evaluation metric. Here, "Uniform $+\alpha$" refers to the process of "correcting word vectors using a uniform prior, then replacing only the norm with that obtained from Zipfian whitening." We found that appropriately weighting by norm has a critical effect on task performance. Notably, pure Zipfian centering/whitening performs even better, suggesting that Zipfian correction has two effects: (i) the norm becomes representative of information content (§ 4.2), and (ii) vectors are more evenly dispersed (isotropic), resulting in appropriate positioning in terms of direction as well.

H Formal Explanation of "Uniform whitening of token embeddings ≈ Zipfian whitening of type embeddings"

In this section, we provide a more formal explanation of "Uniform whitening of token embeddings ≈ Zipfian whitening of type embeddings," as described in § 5.1. For intuitive explanations and related discussions, please refer to § 5.1.

H.1 Uniform whitening of token embeddings ≈ Zipfian whitening of type embeddings

Assume that, when the type word of a token t is w, the token embedding $\boldsymbol{t}$ aligns with the shared type embedding $\boldsymbol{w}$.

Assumption 1. *If type*$(t) = w$*, then* $\boldsymbol{t} = \boldsymbol{w}$*.*

Note that this is a rough approximation, as token embeddings are dynamic and vary with context. Under this assumption, the unweighted mean of token embeddings $\widehat{\mathbb{E}}_{\textcircled{u}}[\boldsymbol{t}]$ obtained from a dataset $\mathcal{D}$ is asymptotically equivalent to a word-frequency-weighted (Zipfian) average of type embeddings $\mathbb{E}_{\textcircled{z}}[\boldsymbol{w}]$, as $|\mathcal{D}| \to \infty$:

$$\widehat{\mathbb{E}}_{\textcircled{u}}[\boldsymbol{t}] := \frac{1}{|\mathcal{D}|}\sum_{t\in\mathcal{D}} \boldsymbol{t} \overset{\text{Assump. 1}}{\approx} \frac{1}{|\mathcal{D}|}\sum_{w\in\mathcal{V}} c_{\mathcal{D}}(w)\boldsymbol{w} \xrightarrow{|\mathcal{D}|\to\infty} \sum_{w\in\mathcal{V}} p(w)\boldsymbol{w} =: \mathbb{E}_{\textcircled{z}}[\boldsymbol{w}], \tag{35}$$

where $c_{\mathcal{D}}$ denotes the count of type w in $\mathcal{D}$: $c_{\mathcal{D}}(w) := \#\{t \in \mathcal{D} : \mathrm{type}(t) = w\}$.

H.2 Pseudo-uniform whitening of token embeddings ≈ uniform whitening of type embeddings

To establish a baseline for centering/whitening token embeddings under uniform prior, we can apply a coefficient $1/|\mathcal{V}_{\mathcal{D}}| \cdot 1/c_{\mathcal{D}}(\mathrm{type}(t))$ to each token embedding t, for removing type frequencies that are implicitly referenced. Here, $\mathcal{V}_{\mathcal{D}}$ denotes the vocabulary contained in $\mathcal{D}$: $\mathcal{V}_{\mathcal{D}} := \{w \in \mathcal{V} : \exists t \in \mathcal{D}, \mathrm{type}(t) = w\}$. The "pseudo-uniform" average $\widehat{\mathbb{E}}_{\textcircled{\hat{u}}}[\boldsymbol{t}]$ calculated in this way is asymptotically equivalent to the uniform average of type embeddings $\mathbb{E}_{\textcircled{u}}[\boldsymbol{w}]$, under the previous assumption (Assump. 1) that ignores the dynamic nature of token embeddings:

$$\widehat{\mathbb{E}}_{\textcircled{\hat{u}}}[\boldsymbol{t}] := \sum_{t\in\mathcal{D}} \frac{1}{|\mathcal{V}_{\mathcal{D}}|}\frac{1}{c_{\mathcal{D}}(\mathrm{type}(t))}\boldsymbol{t} \overset{\text{Assump. 1}}{\approx} \sum_{w\in\mathcal{V}_{\mathcal{D}}} \frac{1}{|\mathcal{V}_{\mathcal{D}}|}\frac{\cancel{c_{\mathcal{D}}(w)}}{\cancel{c_{\mathcal{D}}(w)}}\boldsymbol{w} \xrightarrow{|\mathcal{D}|\to\infty} \sum_{w\in\mathcal{V}} \frac{1}{|\mathcal{V}|}\boldsymbol{w} =: \mathbb{E}_{\textcircled{u}}[\boldsymbol{w}]. \tag{36}$$

I Experimental results on all benchmark datasets to evaluate the effects of uniform whitening on token embeddings

In § 5.1, we evaluated the empirical performance of *uniform* whitening of dynamic *token* embeddings on the STS-B dataset. In this section, we present experimental results using more comprehensive datasets. Table 12 shows the results. Across all datasets, the methods implicitly incorporating a Zipfian prior consistently outperforms the method employing a uniform prior.

Table 8: Full results of the empirical performance of Zipfian whitening. Each cell shows the STS score $\times 100$. As empirical word frequency $p(w)$, we used **enwiki**. Across all models and tasks, Zipfian whitening outperforms powerful baseline methods.

Method		**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**SICK-R**	**STS-B**	**Avg.**
GloVe									
Averaging		56.46	50.41	51.13	58.60	49.03	57.01	46.17	52.69
Uniform	+ Centering	55.54	46.32	49.67	56.03	46.90	56.44	45.17	50.87
	+ Whitening	53.31	62.45	57.93	68.68	58.69	57.92	52.21	58.74
Zipfian	+ Centering	54.52	69.20	60.87	69.82	62.61	58.01	52.25	61.04
	+ Whitening	57.76	**72.22**	**67.04**	**76.80**	**71.72**	**61.80**	**66.92**	**67.75**
ABTT		52.67	67.38	59.40	69.53	60.71	58.56	54.28	60.36
SIF + CCR		**60.23**	68.78	62.39	67.26	61.85	56.91	58.70	62.30
Word2Vec									
Averaging		58.57	68.64	63.65	71.73	61.79	61.77	56.98	63.30
Uniform	+ Centering	58.17	67.34	62.19	70.15	59.60	61.39	55.85	62.10
	+ Whitening	56.53	66.95	62.77	72.42	61.05	62.74	56.03	62.64
Zipfian	+ Centering	56.89	69.95	65.08	73.91	65.71	62.18	58.84	64.65
	+ Whitening	56.16	70.33	**67.20**	**76.60**	**70.99**	**62.52**	**66.50**	**67.19**
ABTT		55.53	69.32	63.13	72.25	60.98	62.02	56.98	62.89
SIF + CCR		**60.05**	**73.26**	66.87	74.32	67.64	59.22	63.04	66.34
fastText									
Averaging		57.94	68.97	62.37	72.26	63.59	59.99	59.82	63.56
Uniform	+ Centering	59.73	55.02	55.16	64.22	53.39	58.85	52.46	56.98
	+ Whitening	52.47	59.01	53.90	65.33	52.61	58.34	48.60	55.75
Zipfian	+ Centering	58.30	71.69	64.57	74.10	67.59	60.75	59.40	65.20
	+ Whitening	58.86	73.85	**68.43**	**78.07**	**74.00**	**62.85**	**69.55**	**69.37**
ABTT		58.35	69.09	60.82	71.99	60.76	60.34	57.02	62.62
SIF + CCR		**61.54**	**76.95**	68.39	76.98	70.27	59.52	67.08	68.67
fastText-subword									
Averaging		49.10	47.34	51.94	61.99	51.54	53.60	50.43	52.28
Uniform	+ Centering	49.21	43.13	49.89	62.03	49.70	54.56	46.91	50.78
	+ Whitening	45.12	41.00	47.30	62.08	48.85	54.80	43.55	48.96
Zipfian	+ Centering	48.68	55.03	54.07	60.23	58.41	54.64	50.38	54.49
	+ Whitening	**61.22**	**60.68**	**63.18**	**73.59**	**69.87**	**59.82**	**68.20**	**65.22**
ABTT		49.64	41.79	48.81	60.84	47.57	55.09	44.23	49.71
SIF + CCR		57.28	54.50	60.77	68.82	61.63	56.83	60.36	60.03

Table 9: Full results of the empirical performance of Zipfian whitening, test set frequency setting. Each cell shows the STS score ×100. As empirical word frequency $p(w)$, we used **test set frequency**. Across all models and tasks, Zipfian whitening outperforms powerful baseline methods. Besides, the test set frequency setting consistently outperforms the enwiki setting in Table 8, demonstrating that the models benefit from using task-specific statistics in line with a covariate shift approach [52].

Method		**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**SICK-R**	**STS-B**	**Avg.**
				GloVe					
Averaging		57.71	50.29	50.61	58.38	48.76	56.76	46.22	52.67
Uniform	+ Centering	56.32	61.17	52.68	64.80	55.80	57.98	47.94	56.67
	+ Whitening	51.67	60.94	57.14	70.09	63.08	55.14	53.16	58.74
Zipfian	+ Centering	50.69	70.66	61.59	70.19	68.25	60.03	56.64	62.58
	+ Whitening	**61.63**	**78.36**	**69.48**	**76.83**	**74.08**	**60.11**	**71.60**	**70.30**
ABTT		52.93	66.93	60.10	71.93	63.12	58.23	53.72	60.99
				Word2Vec					
Averaging		59.00	68.92	63.99	72.51	62.25	61.87	57.15	63.67
Uniform	+ Centering	57.88	70.34	64.24	74.71	65.57	62.47	58.09	64.76
	+ Whitening	58.45	69.42	65.46	76.43	67.78	62.87	60.85	65.89
Zipfian	+ Centering	55.02	71.47	65.81	74.36	69.52	62.92	61.02	65.73
	+ Whitening	**59.37**	**76.92**	**69.48**	**76.42**	**73.56**	**60.07**	**70.42**	**69.46**
ABTT		56.33	70.42	64.71	74.74	65.19	62.55	58.21	64.59
				fastText					
Averaging		58.23	69.36	62.89	73.09	64.25	60.22	60.27	64.04
Uniform	+ Centering	60.60	69.51	61.09	73.92	64.49	61.14	57.42	64.02
	+ Whitening	55.56	63.51	57.73	70.68	62.40	57.93	54.65	60.35
Zipfian	+ Centering	55.92	73.36	65.72	74.12	72.18	62.30	62.95	66.65
	+ Whitening	**62.20**	**79.35**	**71.03**	**77.95**	**76.28**	**60.66**	**73.56**	**71.58**
ABTT		59.13	71.00	63.30	74.80	65.96	61.69	58.23	64.87
				fastText-subword					
Averaging		51.37	51.49	54.57	62.75	52.97	53.53	52.41	54.16
Uniform	+ Centering	51.31	44.80	49.66	62.27	47.43	54.86	43.12	50.49
	+ Whitening	51.52	49.33	53.51	68.28	58.34	56.94	51.69	55.66
Zipfian	+ Centering	43.15	53.40	53.67	63.05	59.09	56.57	47.16	53.73
	+ Whitening	**60.87**	**72.21**	**67.79**	**75.86**	**73.88**	**60.52**	**70.99**	**68.87**
ABTT		49.06	45.16	49.57	62.14	50.75	55.49	44.53	50.96

Table 10: Evaluation results using Japanese fastText. Each cell shows the JSTS [29] score ×100. Even in the multilingual setting, Zipfian whitening outperforms powerful baseline methods.

(a) With **jawiki** as $p(w)$

fastText-ja	55.81	
	Uniform	Zipfian
+ Centering	56.05	**57.55**
+ Whitening	55.53	**65.56**
+ ABTT	57.14	
+ SIF + CCR	61.03	

(b) With **test set frequency** as $p(w)$

fastText-ja	59.94	
	Uniform	Zipfian
+ Centering	59.89	**63.05**
+ Whitening	61.75	**69.86**
+ ABTT	63.02	

Table 11: Each cell shows the STS-B [15] score ×100.

GloVe	43.65		
	Uniform	Zipfian $+\alpha$	Zipfian
+ Centering	41.27	53.66	**55.15**
+ Whitening	53.22	64.83	**70.22**

Table 12: Full results of the whitening on dynamic embeddings. Each cell shows the STS score ×100. Token-level uniform centering/whitening ("Zipfian" settings), which corresponds to centering/whitening at the word type level under a Zipfian prior, consistently outperforms the "Uniform" setting across all STS tasks.

Method		**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**SICK-R**	**STS-B**	**Avg.**
BERT-base uncased									
First-last avg.		45.09	64.30	54.56	70.52	67.87	59.05	63.75	60.73
"Uniform"	+ Centering	47.51	64.53	54.68	72.19	69.28	59.77	64.04	61.71
	+ Whitening	40.31	56.11	47.02	68.35	64.53	48.59	60.53	55.06
"Zipfian"	+ Centering	47.58	66.26	57.32	73.18	71.09	63.27	64.82	63.36
	+ Whitening	**53.75**	**74.07**	**64.21**	**73.88**	**72.83**	**69.71**	**64.91**	**67.62**
RoBERTa-base									
First-last avg.		44.00	59.02	49.31	66.63	59.62	57.56	60.75	56.70
"Uniform"	+ Centering	46.07	55.50	46.27	66.06	60.06	51.33	60.34	55.09
	+ Whitening	37.67	54.64	47.71	66.31	62.85	50.13	61.31	54.37
"Zipfian"	+ Centering	44.97	61.19	53.73	69.57	67.88	58.60	61.30	59.61
	+ Whitening	**52.80**	**73.39**	**64.18**	**72.64**	**72.02**	**71.07**	**65.69**	**67.40**
DeBERTa-base									
First-last avg.		45.03	61.94	52.39	68.90	64.83	56.54	61.66	58.76
"Uniform"	+ Centering	45.20	61.25	50.84	68.56	63.87	53.18	62.01	57.84
	+ Whitening	38.12	50.46	45.30	63.52	62.29	46.99	58.19	52.12
"Zipfian"	+ Centering	45.87	63.24	55.07	70.53	68.88	58.50	63.18	60.75
	+ Whitening	**52.97**	**73.54**	**63.25**	**72.60**	**71.97**	**69.79**	**64.63**	**66.96**

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:

   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: Details are provided in Limitation (§ 6).

   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: [Yes]

Justification: See e.g. Appendix D, Appendix F, and Appendix H.

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: In (Appendix B), we have comprehensively described the experimental settings necessary for reproducing the empirical validations.

   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: Source code has been made publicly available at https://github.com/cl-tohoku/zipfian-whitening.

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: In (Appendix B), we have comprehensively described the experimental settings necessary for reproducing the empirical validations.

   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: [NA]

   Justification: We used pre-trained word embeddings and applied PCA-like transformations in all experiments, eliminating the need for training hyperparameters such as seed values. Additionally, we utilized existing data splits for downstream task evaluations using the standard sentence embedding evaluation tool [40]. Consequently, our experiments did not include error bars or statistical significance tests.

   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: The details on the compute resources used on the experiments are shown in Appendix B.

   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:

   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: Details are provided in Broader Impacts section (§ 6)

    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:

    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:

    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: We provided sufficient documents for using the implementations for the proposed method in the README file in the supplemental material.

    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:

    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:

    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.