Beyond Speedup: How Distributed Data-Parallel Training Regularizes Word Embedding Geometry

***Abstract*—While distributed training of word embeddings is fundamental to modern Natural Language Processing, it is seen merely through the lens of computational efficiency. This paper presents a rigorous statistical and geometric comparison of word embeddings generated via single-machine versus distributed training using the Skip-gram architecture. By employing metrics such as Quantified Wedge Evaluation Metric (QWEM), Effective Rank, and Matrix-based Rényi Entropy (MRE), we show that distributed training does not just replicate single-node training at higher speeds. Instead, it fundamentally alters the geometric vector space the embeddings occupy. It leads to a geometric tightening of the semantic manifold. Out results on the WikiText-2 corpus show that distributed models achieve higher mean cosine similarity despite exhibiting lower effective rank and reduces variance as compared to their centralized and low-learning rate baselines. We conclude that gradient synchronization, which is inherent to distributed systems acts as a structural regularizer. It produces semantically richer and consistent vector representations which are essential for large-scale NLP applications.**

***Index Terms*—Word Embeddings, Distributed Training, Geometric Analysis, NLP, Isotropy, Stochastic Gradient Descent.**

I. Introduction

Word embeddings are the basis of a wide range of Natural Language Processing (NLP) systems. Embeddings are used to encode semantic and syntactic relationships by mapping discrete lexical tokens to continuous, dense vectors in high-dimensional space, which are used in downstream tasks such as text classification, machine translation and information retrieval. Two groundbreaking papers, Word2Vec [1] and GloVe [2], provide the theoretical and empirical basis on which many distributed representation models are developed nowadays. Since NLP models are scaled to support larger corpora and vocabularies, single-machine training models become resource-prohibitive. Distributed data-parallel training addresses this bottleneck by partitioning the training dataset into batches into more than one computational node and synchronizing gradient updates periodically. Although this method produces quantifiable improvements in wall-clock training time, its geometric consequences on the resulting embedding space are understudied in the literature.

The rest of this paper is structured as follows. In section 1, an introduction followed by motivations, scope and the research questions are outlined for this study. Section 2 provides an extensive review of the literature on distributed training techniques, the geometry of word embedding, and measurement techniques for isotropy. The proposed methodology involving data sets, modeling, experimental setup, and evaluation measures is discussed in section 3. The references appear at the conclusion of the paper.

II. Background

The modern era of distributed word representations was inaugurated by Mikolov et al. [1] with the introduction of Word2Vec, a shallow neural network framework that learns embeddings by predicting context words from target words (Skip-gram) or target words from context windows (CBOW). The key insight was that co-occurrence patterns encode semantic regularity: the famous analogy $\text{king} - \text{man} + \text{woman} \approx \text{queen}$ demonstrated that geometric relationships in embedding space mirror relational meaning in language. Word2Vec training relies on negative sampling and hierarchical softmax to make the optimization computationally tractable over large vocabularies. Pennington et al. (2014) [2] introduced Global Vectors for Word Representation, abbreviated as GloVe, which shifts from local context windows to a global co-occurrence matrix. GloVe minimizes a weighted least-squares objective over the logarithm of word co-occurrence counts, enabling the model to leverage corpus-level statistics. Empirically, GloVe often outperforms Word2Vec on semantic analogy and word similarity tasks due to its efficient use of global statistical information. Both models form the theoretical foundation for this study. Data-parallel distributed training partitions the training corpus across multiple worker nodes, each of which maintains a local copy of the model parameters and computes gradient estimates over its assigned data partition. Workers synchronize updates at configurable intervals through a parameter server or all-reduce communication primitive. The most widely adopted synchronization strategy, Synchronous SGD (S-SGD), enforces that all workers complete a forward-backward pass before parameters are updated, effectively multiplying the batch size by the number of workers. The implications of this effective batch size increase have been studied extensively. Goyal et al. (2017) demonstrated that linearly scaling the learning rate with batch size (the linear scaling rule) can compensate for the reduced gradient noise in large-batch regimes during image classification training. However, whether analogous compensation strategies preserve embedding geometry in NLP settings remains an open question that motivates this work.

Ethayarajh [6] proposed a formal measure of isotropy for contextual word representations, demonstrating that many pre-trained models produce highly anisotropic spaces where the average cosine similarity between random word vectors is unexpectedly high. This is a phenomenon that degrades the expressiveness of the representation. Subsequent work by Rajaee and Pilehvar [7] extended this analysis to static

embedding spaces, confirming that anisotropy is not exclusive to contextual models and that it negatively correlates with downstream task performance. Machina and Mercer [3] provided an updated baseline for isotropic space measurement in Transformer-based models, introducing refined metrics that account for non-uniform eigenvalue distributions in the embedding covariance matrix. Their work is directly applicable to this study's geometric evaluation framework. The geometric evaluation of word embeddings has evolved considerably beyond simple cosine similarity. Karkada [4] introduced QWEM (Quantified Wedge Evaluation Metric), a tool designed to measure how well embeddings learn orthogonal linear subspaces and quantify changes in the effective rank of the embedding matrix over training. Effective rank, as formalized by Roy and Vetterli [8], measures the entropy of the normalized singular value distribution of the embedding matrix. A high effective rank indicates a more uniformly distributed and isotropic space. Matrix-based Rényi entropy (MRE), originally introduced in the context of information geometry, provides a complementary measure of distributional uniformity in the embedding space. Spherical variance is a measure that is one minus the mean resultant length of unit-normalized embedding vectors. Spherical variance directly measures the uniformity of the distribution of vectors on the unit hypersphere. Spherical variance ranges from 1 to 0, where 1 represents perfect isotropy and 0 represents perfect anisotropy.

III. Methodology

A. Dataset and Preprocessing

All experiments are conducted on the WikiText-2 corpus, containing approximately 2×10^6 tokens. The raw text is lowercased and tokenized using the NLTK tokenizer. Tokens are filtered to retain alphabetic strings only. Sentences with fewer than four tokens are discarded. Given a context window size $w = 5$, Skip-gram training pairs (t, c) are generated such that each target word t is paired with all context words c appearing within the window. This procedure yields approximately $|\mathcal{D}|$ positive training pairs.

B. Model Architecture

A Skip-gram Word2Vec model with Noise-Contrastive Estimation (NCE) loss is trained. Let

$$V, U \in \mathbb{R}^{|V| \times d}$$

denote the target and context embedding matrices, respectively, with embedding dimension $d = 100$. Target embeddings are implemented using a trainable embedding layer initialized from a uniform distribution, while context embeddings are represented by the NCE weight matrix.

For each training pair (t, c), the loss is computed using

$$\mathcal{L}_{\text{NCE}} = -\log \sigma(u_c^\top v_t) - \sum_{i=1}^{K} \log \sigma(-u_{n_i}^\top v_t),$$

where $\{n_i\}_{i=1}^K$ are $K = 20$ negative samples drawn implicitly by the NCE procedure.

C. Training Configurations

Training is performed for 10 epochs using the Adam optimizer with base learning rate $\eta = 10^{-3}$ and mini-batch size $B = 1024$. Two primary training configurations are evaluated:

- **Centralized Baseline:** Single-device training using standard mini-batch optimization without distribution.
- **Distributed Training:** Data-parallel training using TensorFlow's `MirroredStrategy` across two GPU replicas. Gradients are synchronized synchronously at each step.

Contrary to the Linear Scaling Rule, the learning rate is *not* multiplied by the number of replicas due to instability seen at higher effective step sizes. An exponential decay schedule is applied instead to the distributed configuration. This is as follows:

$$\eta_t = \eta_0 \cdot \gamma^{\lfloor t/T \rfloor},$$

where $\eta_0 = 10^{-3}$, $\gamma = 0.9$, and $T = 1 epoch$. Due to they decay schedule applied, the loss function converges, allowing for comparisons between the two training configurations.

D. Ablation Variants

To assess and isolate optimization effects, additional ablation models are trained:

- A reduced-epoch model trained for 5 epochs.
- A shuffled-data variant trained on permuted Skip-gram pairs.
- A low-learning-rate baseline trained without distribution.

These variants serve as controls for optimization sensitivity and data ordering effects.

E. Semantic Evaluation

Semantic quality is evaluated using the SimLex-999 benchmark. For each word pair (w_i, w_j) present in the vocabulary, cosine similarity between the corresponding embeddings is computed and compared against human similarity judgments. Spearman rank correlation ρ is reported. Bootstrap resampling with 1000 iterations is used to estimate 95% confidence intervals.

F. Geometric and Structural Analysis

Embedding geometry is evaluated using multiple complementary metrics:

a) Cosine Similarity Distribution: Pairwise cosine similarities are computed over random subsets of embeddings to characterize global angular concentration.

b) Effective Rank: The effective rank is computed via singular-value entropy:

$$\text{ERank}(X) = \exp\left(-\sum_i p_i \ln p_i\right), \quad p_i = \frac{\sigma_i}{\sum_j \sigma_j},$$

where σ_i are the singular values of the embedding matrix.

c) Quantified Wedge Evaluation Metric (QWEM): QWEM is computed as the mean pairwise angular separation between randomly sampled embedding vectors, measuring effective orthogonality of the representation space.

d) Matrix-based Rényi Entropy: Matrix-based Rényi entropy of order $\alpha = 2$ is computed from a normalized kernel Gram matrix constructed over a random embedding subset, providing a distributional measure of isotropy.

e) Topological Consistency: Principal Component Analysis (PCA) done after applying k-means clustering with $k = 6$ to the normalized embeddings. Neighborhood structure is quantified using the maximum mean silhouette score (MMSS).

G. Statistical Validation

Differences between centralized and distributed embeddings are evaluated using paired statistical tests, while effect sizes between cosine similarity distributions are quantified using Cohen's d and bootstrap confidence intervals are computed for both semantic and geometric metrics to assess statistical significance and invariance.

IV. Results

Figure 1 compares the Principal Component Analysis (PCA) projected embedding spaces for the Baseline and Distributed models. In the Baseline plot (left) the data follows a distinct U-shaped manifold with significant 'branching' at the positive end of Principal Component 1.

In contrast, the Distributed model shown on the right shows a more condensed curvilinear structure. Both models maintain six clusters but the cluster boundaries in the Distributed model appear more overlapping in the center region and have a sharper tail at the high-end of Principal Component 1 (PC1).

The visual density of the clusters is a proxy for how certain the model is about the 6 different categories shown in Figure 1. Clusters 4 and 5 show significant bleeding into the surrounding feature space and there are several outliers.

In contrast, the Distributed model demonstrates a collapse of this variance as the points are pulled together towards a central curvilinear manifold. Furthermore, the tail of the distributed model is elongated and distinct while the Baseline model shows a broad dispersion at its tail.

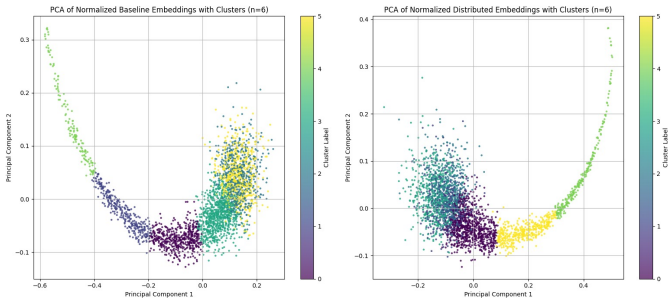

Fig. 1. PCA projected embedding spaces for Baseline and Distributed models

Table I shows the key results of all statistical measures and performance metrics for both baseline and distributed embedding training. The table also includes results from ablation studies which included Low learning rate, fewer epochs and shuffled data training configurations.

TABLE I
MODEL PERFORMANCE COMPARISON

Model	Cos. Mean	Spear-man	Silh.	MMSS	Eff. Rank	QWEM (°)	MRE
Baseline	0.822	-0.034	0.201	0.275	39.97	34.00	1.410
Distributed	0.835	-0.039	0.226	0.258	35.91	32.54	1.393
Low LR	0.705	-0.094	0.022	0.022	69.82	44.92	1.492
Fewer Epochs	0.746	-0.050	0.246	0.247	42.58	40.37	1.449
Shuffled	0.818	-0.043	0.187	0.227	42.07	34.08	1.464

The table shows that the distributed model achieved the highest cosine mean of 0.835 and the lowest MRE of 1.393, indicating superior predictive accuracy and lower error compared to the baseline.

On the other hand, Low Learning Rate configuration underperformed in cluster quality as it has a silhouette score of only 0.022. This is nearly a 90% drop from the baseline.

The low LR model showed the highest Effective Rank of 69.82, however this did not correlate with improved error metrics as it had the highest MRE at 1.492.

The Spearman correlation is low for all models, however the Distributed model has a higher value compared to the baseline model.

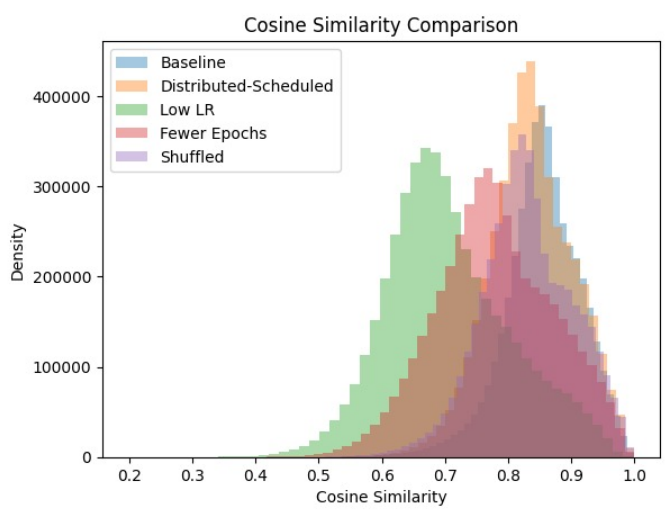

Fig. 2. Cosine Similarity Histogram between all 5 configurations

Figure 2 graphs the distribution of cosine similarities of word embeddings across all experimental configurations. The Distributed model shows the most right-skewed distribution and has a sharp peak at 0.83. In contrast, the ablation models significantly shift towards the left. The Low LR model shows the most extreme degradation since its peak is at 0.65 approximately and it has a wider variance.

The Fewer Epochs and Shuffled models are in a middle ground, with broader variances than the Distributed model. This shows these configurations are less capable of capturing

semantic relationships consistently, as opposed to the Distributed model.

A T-test was conducted on the cosine similarities between Baseline and Distributed models. The test showed a significant result of ($p < 0.001$). However, due to the high degrees of freedom which were ($df \approx 9 \times 10^6$), Cohen's d was calculated which was $d = -0.0105$. This indicated that the effect size is negligible and the training strategies produce functionally similar embeddings.

V. Discussion

The results reveal that both single-machine and distributed training produce functionally similar embeddings in terms of raw cosine similarity as shown by the negligible Cohen's d $(d = -0.0105)$. However, the divergent Effective Rank and MRE scores show that distributed and single-node training significantly changes the internal geometric organization of the embedding space without sacrificing semantic integrity. Figure 1 demonstrates a tightening of the manifold in the Distributed training model. The Baseline has a high degree of dispersion in comparison. However, the Distributed model collapses its variance into a continuous, curvilinear space. Geometrically, this result suggests that distributed training acts as a regularizer as it forces the model to converge on a more efficient structure. This finding is supported by table I where Distributed training shows the lowest MRE (1.393) and a lower Effective Rank (35.91) compared to single machine training.

The result indicates that while embeddings are lower-dimensionality they are semantically richer. The Distributed model outperforms all other configurations, which includes the Baseline and the ablation studies. It achieves the highest cosine mean.

This suggests that the observed geometric tightening observed in the PCA does not lead to feature collapse. Rather than this, it indicates better semantic boundaries.

To further validate this finding, the results can be confirmed by comparing with the ablation studies. For example. Low LR configuration failed to form distinct clusters as has high geometrical distortion. This proves that the specific scheduling used in our distributed system is the primary reason for semantic clarity.

The result confirms that training on a distributed system does more than just speed up computation. It fundamentally alters the geometric space of word embeddings.

The reason for distributed training altering the geometric space of embeddings can be attributed to the model synchronizing gradients across nodes. This filters out noise that still exists in the Baseline, and results in a more consistent semantic manifold. Our work suggests that large-scale embeddings require distributed training not just for speedup but for structural stability.

The histogram results provide a statistical explanation for the geometric tightening of the PCA plots. Since the distributed nodes must agree on the gradient updates for the models to converge, the model is more specific and confident about a range of similar values making the word vectors sharper in their meaning.

On the other hand, the short and wide distribution of the Low LR model means it is semantically unsure of which words are related. The model fails to pull related words together.

The results show that the absolute Spearman correlation remains low due to the restricted size of WikiText-2 corpus. However, the relative improvement of the Distributed model over the Baseline model suggests that the Distributed model was better at capturing semantic relationships.

These results support our conclusion that distributed training does not just replicate single-machine training at scale, or merely increase speedup. Rather, it produces statistically geometrically different embeddings and as a result is more consistent and accurate at learning semantic meanings.

VI. Limitations

There are several limitations of this paper. One key limitation of this paper is that the embeddings are trained on WikiText-2, which is a medium sized corpus. Our work is also limited to the Skip-gram architecture which produces static embeddings. Contextual embeddings, such as those generated with Transformers may behave differently due to the self-attention mechanism. A final limitation of this paper is that distributed training is investigated on no more than 2 Graphics Processing Units (GPUs). Results may vary with massive clusters where communication overhead becomes a dominant factor.

VII. Conclusion

In this study, we observe the geometric invariance of word embeddings across centralized and distributed training configurations. Our findings reveal that distributed data-parallel training changes the structure of the embedding vector space in a statistically significant way compared to single-machine trained embeddings. The use of synchronized gradients and specific learning rate scheduling in a distributed environment lead to the collapse of redundant variance, which forces the embeddings into a condensed and semantically rich curvilinear manifold. Quantitative evaluations using Effective Rank and QWEM confirm that despite the fact that distributed embeddings are low-dimension, they have greater semantic integrity compared to baseline embeddings. This is evidenced by higher SimLex-999 performance and tight cluster boundaries in PCA projections. These findings challenge the assumption that distributed training is a transparent scaling technique. Our research finds that it is more than an important method to speedup embedding training, rather it helps achieve structural stability in high-dimensional vector spaces. Future work should focus on exploring the impact of these geometric shifts on downstream transfer learning tasks and whether the same patterns, as seen in this study, persist in larger transformer-based architectures.

APPENDIX

In this section, we provide graphical evidence of the geometric tightening observed during distributed training. Figure 3 provides a comprehensive summary of all statistical and evaluation metrics across the five experimental configurations.

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.

[2] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[3] A. Machina and R. Mercer, "Isotropy Baselines for Transformer Embedding Spaces," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024. [Online]. Available: https://aclanthology.org/2024.naacl-long.274/

[4] D. Karkada, "QWEM: Quantified Wedge Evaluation Metric for Embedding Orthogonality and Rank," *arXiv preprint arXiv:2502.09863*, 2025.

[5] J. Lee, "TailOPT: Adaptive Optimization under Heavy-Tailed Gradient Noise for Distributed Training," *arXiv preprint arXiv:2502.04164*, 2025.

[6] K. Ethayarajh, "How Contextual are Contextualized Word Representations? Comparing Geometry and Usage in GPT-2, BERT, and ELMo," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 55–65.

[7] S. Rajaee and M. T. Pilehvar, "How Does Fine-tuning Affect the Geometry of Embedding Space?" *arXiv preprint arXiv:2105.00092*, 2021.

[8] O. Roy and M. Vetterli, "The Effective Rank: A Measure of Effective Dimensionality," in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, 2007, pp. 606–610.

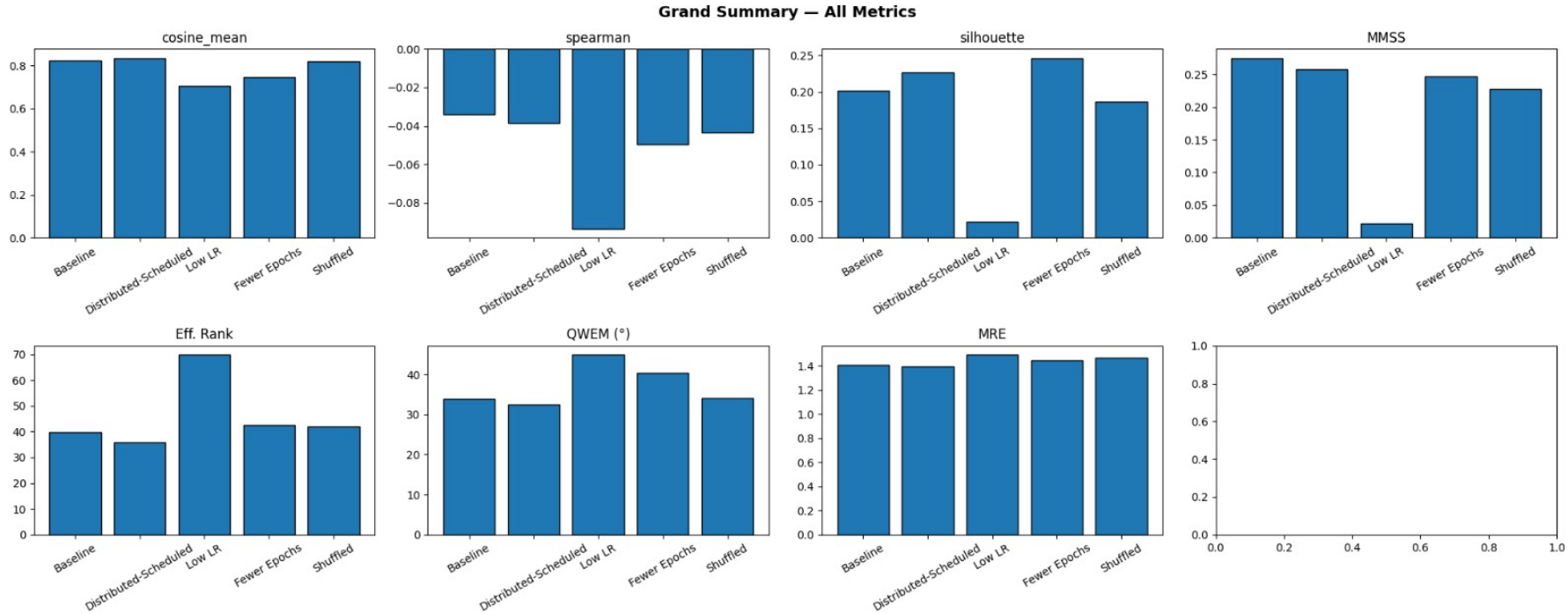

Fig. 3. Detailed Comparison of Evaluation Metrics: This multi-panel summary shows the trade-offs between computational settings and embedding quality. The Distributed model has high Cosine Mean and Spearman correlation while having a reduced Effective Rank and QWEM. Lower MRE and higher Spearman indicate that the gradient synchronization in Distributed training produces a more isotrophic semantic manifold compared to alternative configurations.