FROM WALK-DERIVED DISTANCE TO EUCLIDEAN GRAPH GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Shortest path distances are strong signals to capture graph geometry in a Euclidean embedding space. Random walks determine an upper bound on graph distance between visited node pairs and repeated walks only tightens the upper bound. In addition, random walks provide a computationally efficient method to select a subset of interacting node pairs for the graph embedding. We propose Fodiwalk, a two-stage framework to efficiently extract graph-related parameters such as shortest path distance estimates in the first stage and use them in the second stage to reflect graph geometry in the generated embedding. We employed a guaranteed-to-converge force model, which directly updates the embedding gradients using the Euclidean and the estimated graph distances. Experiments on different datasets show enhanced results on various prediction tasks and geometric metrics, while retaining robustness in different dimension sizes.

1 INTRODUCTION

Graph embeddings are routinely consumed as coordinates. Yet the properties needed to classify edges, retrieve adjacent vertices, cluster communities, and rank shortest-path distances are different. In particular, a strong edge-reconstruction classifier does not imply that Euclidean separation tracks graph distance. We make the latter question explicit. Most random-walk methods learn co-occurrence objectives.

We present Fodiwalk framework uses observed graph distances in a direct force-directed embedding. A sparse augmentation stage selects pairs and records their minimum observed gap; an embedding stage combines local attraction with distance-decaying nonlocal repulsion scaled by the selected coefficient. Supplemental sampled pairs provide repulsion without pretending to measure graph distance.

Our empirical question is whether this mechanism yields useful *Euclidean* geometry. We compare it with walk, spectral, factorization, projection, and BFS-informed reference embeddings in 128-dimensional spaces. We then run experiments on dimension sizes and the interacting node pairs sets to examine the amount of retained support. Across the broad comparison, Fodiwalk has the largest measured distance–hop rank correlation compare to other learned methods on. Some methods perform better on some dataset and task pairs.

The contribution is a sparse incorporation of graph distance estimates into functions of a prescribed force model, an analysis of what the walk observations actually certify, and a multimetric evaluation with explicit protocol boundaries.

2 RELATED WORK

2.1 WALK, PROXIMITY, AND FACTORIZATION EMBEDDINGS

DeepWalk learns node representations from truncated random-walk contexts, and node2vec varies those contexts with a second-order walk bias (Perozzi et al., 2014; Grover & Leskovec, 2016). LINE models first- and second-order proximity (Tang et al., 2015); NetMF makes explicit the matrix factorization underlying several walk-context objectives (Qiu et al., 2018); and GraRep extends

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

factorization to higher-order transition structure (Cao et al., 2015). HOPE preserves selected high-order proximity matrices (Ou et al., 2016), RandNE uses random projection for scalable network embedding (Zhang et al., 2018), and ProNE combines sparse factorization with spectral propagation (Zhang et al., 2019). These methods make different choices about the structural signal and about how it is read at evaluation. In particular, several standard implementations are optimized for inner-product scores, whereas the geometric measurements in this paper read Euclidean distance. That objective–metric mismatch qualifies interpretation; it does not remove those embeddings from the common evaluation.

2.2 FORCE-DIRECTED EMBEDDINGS

Force2Vec (Rahman et al., 2020) maps force-directed embedding to a deep learning model. Their loss function essentially considers attraction between neighboring nodes and repulsive between sampled non-neighbor nodes. Its published implementation exposes a quadratic base configuration and scalable tForce2Vec and rForce2Vec options. All three are therefore reported separately rather than treated as one baseline. In contract, the force-directed embedding in FODIWALK is a single prescribed force model that directly updates embedding gradients using the force functions.

The closest lineage to FODIWALK is (Lotfalizadeh & Hasan, 2023; 2024; Lotfalizadeh, 2024). They assume attractive and repulsive forces between all-pairs, mass-normalized motion, and exact hop-distance parameter in force functions. In contrast, FODIWALK uses a simpler force model on a subset of pairs discovered through finite walk observations.

2.3 DISTANCE-AWARE AND SPECTRAL EMBEDDING METHOD

Laplacian Eigenmaps gives a spectral local-smoothness baseline (Belkin & Niyogi, 2001). Sparse Stress approximates an explicit stress objective using representative interactions (Ortmann et al., 2016). Landmark shortest-path MDS is included here as an oracle-informed reference: it uses exact shortest-path information from selected landmarks, whereas FODIWALK uses walk observations (de Silva & Tenenbaum, 2004). It is a distance-informed reference, not an upper bound on the score achieved by every embedding procedure.

3 FODIWALK FRAMEWORK

FODIWALK is a walk-based force-directed graph embedding framework. The embedding kernel function in Fodiwalk is based on a force model with guaranteed convergence properties, derived from kinematics equations and Newton's second law Lotfalizadeh & Hasan (2023; 2024); Lotfalizadeh (2024). Each graph node is treated as a massive object in the d-dimensional embedding space, where its position evolves under forces exerted by other nodes.

The framework comprises graph augmentation and force-directed embedding stages. During augmentation, graph exploration identifies interacting node pairs and extracts topological information used to parameterize their forces. Random walks are employed to construct a sparse weighted interaction graph, incorporating both local connectivity and information about longer-range graph separation. The embedding stage subsequently uses this augmented graph to compute the resulting force on each node and directly determine its displacement. Iterative updates move the embeddings toward a configuration in which the aggregate forces approach equilibrium.

3.1 GRAPH AUGMENTATION USING RANDOM WALKS

In graph augmentation, some nodes are paired up and connected by weighted edges. The neighboring nodes in the augmented graph build the set of interacting node pairs. Let $\mathcal{P}(u) = \mathcal{N}_G(u) \cup \mathcal{W}(u) \cup \mathcal{F}(u)$ be the neighbors of u in the augmented graph, comprised of $\mathcal{N}_G(u)$ the neighbors of u in the original graph, $\mathcal{W}(u)$ the nodes discovered during random walks through u, and $\mathcal{F}(u)$ a sampled set of distant nodes not in the other sets.

Random-walk sampling and pair discovery. Fodiwalk employs random walks to discover pairwise topological relationships and construct a sparse weighted interaction graph. The augmentation

Algorithm 1 Fodiwalk graph augmentation

Require: graph $G = (V, E)$; walks per node R; walk length L; window w; per-node cap m; supplemental count Q; far weight h_{far}
Ensure: augmented interaction matrix D

for $u \in V$ **do**
  $\mathcal{N}_G(u) \leftarrow$ neighbors of u in G
  $\mathcal{G}_{uv} \leftarrow \emptyset$ for all v ▷ multiset of positional gaps, per pair
  for $r = 1, \ldots, R$ **do**
    draw walk $\mathcal{W}_u^{(r)} = (X_0, \ldots, X_{L-1})$, $X_0 = u$, with $\Pr(X_{t+1} = v \mid X_t = u) = 1/\deg(u)$ ▷ first-order uniform transitions
    for each pair of positions $i < j$ with $j - i \le w$ **do**
      append gap $g = j - i$ to $\mathcal{G}_{X_i X_j}$
    end for
  end for
  $c_{uv} \leftarrow |\mathcal{G}_{uv}|$ and $h_{uv} \leftarrow \min \mathcal{G}_{uv}$ for every observed pair v
  $\mathcal{W}(u) \leftarrow$ the m pairs v with largest c_{uv}
  $\mathcal{F}(u) \leftarrow Q$ nodes sampled uniformly from $V \setminus (\{u\} \cup \mathcal{N}_G(u) \cup \mathcal{W}(u))$
  $\mathcal{P}(u) \leftarrow \mathcal{N}_G(u) \cup \mathcal{W}(u) \cup \mathcal{F}(u)$
end for
assemble D from Eq. equation 3: $D_{uv} = 1$ for $v \in \mathcal{N}_G(u)$, $D_{uv} = h_{uv}$ for $v \in \mathcal{W}(u)$, $D_{uv} = h_{\text{far}}$ for $v \in \mathcal{F}(u)$, else 0
return D

Algorithm 2 Fodiwalk force-directed embedding

Require: augmented matrix D; dimension d; degree normalization $d_u = \deg(u)$ (or explicit override); step sizes η_t; optional row-drop probability p; tolerance ε; maximum epochs T
Ensure: embedding $Z = \{z_u\}_{u \in V} \subset \mathbb{R}^d$

initialize $z_u^{(0)}$ for all $u \in V$
$\mathcal{P}(u) = \{v : D_{uv} > 0\}$ for all $u \in V$ ▷ from D
for $t = 0, 1, \ldots, T-1$ **do**
  for $u \in V$ **do** ▷ compute forces from the fixed configuration $Z^{(t)}$
    for $v \in \mathcal{P}(u)$ **do**
      $x_{uv}^{(t)} \leftarrow \|z_v^{(t)} - z_u^{(t)}\|_2, \quad \widehat{e}_{uv}^{(t)} \leftarrow (z_v^{(t)} - z_u^{(t)})/x_{uv}^{(t)}$
      $f_{uv}^{(t)} \leftarrow f(x_{uv}^{(t)}, D_{uv})$ ▷ Eq. equation 8
    end for
    $F_u^{(t)} \leftarrow \sum_{v \in \mathcal{P}(u)} f_{uv}^{(t)} \widehat{e}_{uv}^{(t)}$ ▷ Eq. equation 11
    $\Delta z_u^{(t)} \leftarrow F_u^{(t)}/d_u$
  end for
  for $u \in V$ **do** ▷ apply updates only after every force is computed
    $z_u^{(t+1)} \leftarrow z_u^{(t)} + \leftarrow$ OPTIMIZER$(\Delta z_u^{(t)}, u, t)$ ▷ plain update Eq. equation 15, momentum, Nesterov, EMA velocity, Adam, quasi-Newton-style, or adaptive rate;
  end for
  if $\frac{1}{n} \sum_{u \in V} \|\Delta z_u^{(t)}\|_2 \le \varepsilon$ **then**
    break
  end if
end for
return $Z^{(t+1)}$

procedure adopts the sliding-window principle of DeepWalk Perozzi et al. (2014), and records positional gap between nodes in Walklets Perozzi et al. (2017).

although Fodiwalk aggregates the observed separations into a single weighted interaction graph rather than learning separate representations at different walk lengths.

For each node $u \in V$, the algorithm generates R random walks of L positions,

$$\mathcal{W}_u^{(r)} = (X_0^{(r)}, X_1^{(r)}, \ldots, X_{L-1}^{(r)}), \qquad X_0^{(r)} = u. \tag{1}$$

For first-order uniform walks, the transition probability for nodes with nonzero degree is

$$\Pr(X_{t+1} = v \mid X_t = u) = \frac{1}{\deg(u)}, \tag{2}$$

Given a walk $\mathcal{W}$ and a window size w, every pair of distinct nodes appearing at positions $i < j$ with $j - i \leq w$ contributes a candidate interaction. Candidate pairs are collected from all eligible positions along the walk trajectory, regardless of the starting node. Each occurrence contributes an unordered pair $\{X_i, X_j\}$ and its positional gap $g = |j-i|$. Repeated occurrences are retained during aggregation, allowing both their frequencies and observed separations to be measured.

Let $\mathcal{G}_{uv}$ denote the multiset of positional gaps associated with pair $\{u, v\}$ across the sampled walks. Fodiwalk records its encounter frequency $c_{uv} = |\mathcal{G}_{uv}|$ and assigns the pair the minimum observed gap $h_{uv} = \min_{g \in \mathcal{G}_{uv}} g$ as a proxy for the shortest-path distance. For a walk-derived pair, the observed gap satisfies $d_G(u, v) \leq h_{uv}$, where $d_G(u, v)$ denotes shortest-path distance in the original graph. Thus, the minimum observed gap provides an upper-bound estimate of graph separation without explicitly computing shortest paths. Repeated observations provide higher confidence an estimate shortest-path.

The walk budget controls the number of candidate walk-derived pairs. A walk of length L generates at most $\sum_{g=1}^{\min(w,L-1)}(L-g)$ positional pair occurrences . For $w < L$, this becomes $wL - w(w+1)/2$, yielding a total candidate-generation cost of $O(nRLw)$ before aggregation and selection.

Interaction selection and graph construction. To control the size of the augmented graph, each node selects up to m candidates with the highest encounter frequencies, c_{uv}. An unordered pair is retained if either endpoint selects it. This union-based selection produces symmetric interaction support while retaining relationships selected from either endpoint's sampled neighborhood, an approach related to sparse neighborhood-graph construction in LargeVis and UMAP Tang et al. (2016); McInnes et al. (2018).

The resulting set contains at most nm distinct selected walk-derived pairs, although individual nodes may participate in more than m retained interactions. For large graphs, intermediate frequency-based pruning additionally limits the candidate accumulator; when activated, this procedure approximates the statistics of the complete sampled walk corpus.

Distant nodes. The standard augmentation further introduces supplemental interaction pairs, following the general negative-sampling principle employed in neighborhood embedding methods. Candidate endpoints are sampled uniformly from V, and self-pairs, existing interactions, and duplicate accepted pairs are rejected. Each accepted pair is inserted in both directions with a prescribed large h_{far} value.

Unlike walk-derived coefficients, supplemental weights do not represent observed graph distances. Their purpose is to introduce additional repulsive interactions beyond the relationships retained by window-based sampling. The selected pairs need not be topologically distant and may connect nodes belonging to different connected components.

The resulting weighted interaction matrix D is

$$D_{uv} = \begin{cases} 1, & v \in \mathcal{N}_G(u), \\ h_{uv}, & v \in \mathcal{W}(u), \\ h_{\text{far}}, & v \in V \setminus (\{u\} \cup \mathcal{N}_G(u) \cup \mathcal{W}(u)), \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

A zero entry indicates the absence of an interaction, rather than zero graph distance.

3.2 Force-directed Graph Embedding

Given the augmented graph from Section 3.1, the embedding stage places every node in $\mathbb{R}^d$ and repeatedly displaces it according to a prescribed force law until the configuration approaches equilibrium. This subsection states the physical analogy that motivates the force law, defines the scalar

force function and the criterion under which it is admissible to an inherited convergence framework, aggregates per-node forces into an update, and surveys the numerical schemes used to apply that update.

Kinematic background. Fodiwalk treats each node u as a point mass in the embedding space, with effective mass $\mu_u > 0$ ordinarily set to its degree. Each node u perceives a force contributed by every $v \in \mathcal{P}(u)$, the set of nodes exerting a force on u (Section 3.1); this force depends only on the current Euclidean separation between u and v and on their augmented coefficient h_{uv}. There is no persistent velocity state carried between epochs. Under Newton's second law, the acceleration of u at epoch t is

$$a_u^{(t)} = \frac{F_u^{(t)}}{\mu_u}, \tag{4}$$

where $F_u^{(t)}$ is the net force described below. Assuming the node starts each relocation interval from rest and that the interval's acceleration is constant, the kinematic equation of motion $z = \frac{1}{2}at^2 + v_0 t + z_0$ reduces, with $v_0 = 0$, to a displacement proportional to $a_u^{(t)}$

$$\Delta z_u^{(t)} \;\propto\; a_u^{(t)}, \tag{5}$$

with the quadratic time factor absorbed into a scalar step size η_t. Equations equation 4–equation 5 are the same kinematic derivation used by the force-directed embedding line this work extends (Lotfalizadeh & Hasan, 2023; 2024; Lotfalizadeh, 2024); Fodiwalk's contribution at this stage is the augmented, walk-derived coefficient h_{uv} that parameterizes the force perceived from each $v \in \mathcal{P}(u)$, not the kinematic derivation itself.

Force model. For each $v \in \mathcal{P}(u)$, write the Euclidean separation between u and v and the unit direction from u toward v as

$$x_{uv} = \|z_v - z_u\|_2, \qquad e_{uv} = \frac{z_v - z_u}{x_{uv}} \quad (x_{uv} > 0). \tag{6}$$

The force perceived by u from v is collinear with e_{uv} and scales with a signed scalar factor that depends on the current separation and the augmented coefficient of the pair:

$$F_{uv} = f(x_{uv}, h_{uv})\, e_{uv}. \tag{7}$$

A positive and negative factors indicate attractive and repulsive forces exerted from v to u. At $x_{uv} = 0$ the contribution of v is set to zero by convention, avoiding division by zero at the cost of a discontinuous direction field at collisions.

Force function and convergence criteria. Fodiwalk splits the scalar factor by the augmented coefficient, treating original edges ($h = 1$) and walk-witnessed or supplemental nonedges ($h \geq 2$) differently:

$$f(x,h) = \begin{cases} k_1 x - k_r, & h = 1, \\ -k_r h \exp(-k_4 x), & h \geq 2, \end{cases} \qquad k_1, k_r, k_4 > 0. \tag{8}$$

For $h = 1$, attraction grows linearly with the Euclidean distance while repulsion has constant magnitude k_r; the two balance at the isolated-pair equilibrium $x^\star = k_r/k_1$. For $h \geq 2$, the coefficient only modulates a purely repulsive term whose magnitude grows linearly in h and decays exponentially in x.

This function is the object that must satisfy the two conditions the earlier kinematic line identifies as sufficient for its convergence argument (Lotfalizadeh & Hasan, 2024): a uniform lower bound and an asymptotic slope strictly below one. Both are simple to check for Eq. equation 8 at the level of a single pair. On any finite fixed support with maximum coefficient $H = \max_{v \in \mathcal{P}(u),\, u \in V} h_{uv}$,

$$f(x,h) \;\geq\; -k_r H \qquad \text{for all } x \geq 0, \tag{9}$$

so $b = k_r H$ is a uniform lower bound. The per-pair asymptotic growth rate satisfies

$$\lim_{x \to \infty} \frac{f(x,h)}{x} = \begin{cases} k_1, & h = 1, \\ 0, & h \geq 2, \end{cases} \tag{10}$$

so choosing $k_1 < 1$ meets the asymptotic-slope condition for every individual $v \in \mathcal{P}(u)$, regardless of its coefficient. These checks establish that f is an admissible force factor for the inherited framework at the pairwise level. Section 3.2 below shows that an additional normalization is needed once these contributions are summed over $\mathcal{P}(u)$. The proof of convergence relies on Brouwer's fixed-point argument for the existence of an equilibrium configuration (Lotfalizadeh & Hasan, 2024; Lotfalizadeh, 2024).

Net force, degree normalization, and the embedding gradient. The net force on u shown in equation 11 is the vector sum of the contributions perceived from every node exerting a force on it.

$$F_u^{(t)} = \sum_{v \in \mathcal{P}(u)} f\big(x_{uv}^{(t)}, h_{uv}\big)\, e_{uv}^{(t)}. \tag{11}$$

The per-pair slope condition in Eq. equation 10 does not carry over unchanged to this sum. As every x_{uv} grows to infinity, only neighboring nodes ($h_{uv} = 1$) contribute a nonzero force factor. Writing $d_u = |\{v \in \mathcal{P}(u) : h_{uv} = 1\}|$ for the number of such neighbors (i.e. the degree of u), the net force therefore grows as

$$\lim_{x \to \infty} \frac{F_u^{(t)}}{x}\bigg|_{x_{uv} \equiv x\ \forall v} = k_1 d_u, \tag{12}$$

which exceeds one whenever $d_u > 1/k_1$. The quantity that must satisfy the asymptotic-slope condition is therefore the mass-normalized displacement field, not $F_u^{(t)}$ itself. Dividing by μ_u and setting $\mu_u = d_u$ cancels exactly this factor:

$$\frac{|F_u^{(t)}|/\mu_u}{x} = \frac{\sum_{v \in \mathcal{P}(u)} f(x, h_{uv})/\mu_u}{x} \xrightarrow[x \to \infty]{} \frac{k_1 d_u}{d_u} = k_1 < 1. \tag{13}$$

Degree normalization is thus not an incidental modeling choice, but required for the convergence condition. With this normalization the displacement field is

$$g_u^{(t)} = \frac{F_u^{(t)}}{\mu_u}, \tag{14}$$

which is informally referred as an embedding gradient even though it is evaluated directly from Eq. equation 8 rather than by differentiating a separately defined loss. The name is not purely informal, however: when the weight-one interaction subgraph equals the original graph and $\mu_u = d_u$, the aggregate contribution from edge partners to g_u reduces to $k_1\big(\frac{1}{d_u}\sum_{v \in N(u)} z_v - z_u\big)$, a degree-normalized neighborhood-smoothing step, and on fixed symmetric support the full field is exactly $-M^{-1}\nabla E(Z)$ for a pairwise potential E whose derivative recovers f, a mass-preconditioned gradient of an explicit energy. Equilibrium requires only that each node's *net* displacement vanish, which can occur when the netfor in the sum equation 11 does.

Optimizers. The plain update applies the displacement field directly, with a scalar step size η_t and, in the stochastic variant used for the reported runs, an independent Bernoulli row mask $\xi_u^{(t)}$ that freezes a node's position for the epoch with no inverse-probability rescaling of the retained rows:

$$z_u^{(t+1)} = z_u^{(t)} + \eta_t\, \xi_u^{(t)}\, g_u^{(t)}. \tag{15}$$

Beyond this plain rule, the same displacement field $g_u^{(t)}$ can be routed through standard first-order optimizers that only change how the computed displacement is applied, not the underlying force function: momentum, Nesterov-style lookahead updates, exponentially averaged velocity, Adam, quasi-Newton-style updates, and adaptive per-node step sizes are all supported, together with a learning-rate schedule for η_t. Iteration terminates after a prescribed epoch budget or once the mean displacement norm falls below a threshold ε,

$$\frac{1}{n}\sum_{u \in V} \big\|g_u^{(t)}\big\|_2 \;\le\; \varepsilon. \tag{16}$$

The choice of optimizer and of the stochastic row-masking rate are treated as configuration variables rather than fixed parts of the force model, and their effect on the reported correlations is examined empirically in Section 5 rather than assumed from the convergence checks above.

Table 1: Euclidean distance-rank correlation ρ at 128 . Values are mean $\pm$ standard deviation over 3 repeats. "–" marks unrun PubMed cells due to resource constraints. Base Force2Vec has quadratic all-pairs cost, while GraRep, NetMF, and HOPE require dense $n \times n$ representations.

Method	Cora	CiteSeer	PubMed
Fodiwalk–SQN	**.8656 ± .0016**	.8280 ± .0136	**0.7933 ± .0125**
Landmark MDS	.8172 ± .0111	**0.9545 ± .0054**	.7690 ± .0193
DeepWalk	.7879 ± .0078	.6860 ± .0006	.6990 ± .0045
node2vec	.7564 ± .0099	.6871 ± .0096	.6788 ± .0098
tForce2Vec	.7168 ± .0208	.7677 ± .0083	.5928 ± .0151
Force2Vec	.5442 ± .0093	.4461 ± .0101	– ($O(n^2)$)
Laplacian Eigenmaps	.4800 ± .0116	.4742 ± .0173	.4289 ± .0164
RandNE	.4326 ± .0259	.3760 ± .0082	.4800 ± .0256
rForce2Vec	.4179 ± .0119	.2813 ± .0041	.3621 ± .0071
ProNE	.0564 ± .0030	.0260 ± .0102	.1243 ± .0104
LINE	-.1400 ± .0173	-.1433 ± .0146	-.2978 ± .0091
GraRep	-.2123 ± .0134	-.2744 ± .0156	– (dense $O(n^2)$)
NetMF	-.2715 ± .0151	-.2984 ± .0268	– (dense $O(n^2)$)
HOPE	-.3827 ± .0104	-.3728 ± .0172	– (dense $O(n^2)$)

4 Experimental Setup

Broad evaluation. Our primary score is Spearman ρ between Euclidean embedding distances and true BFS hops for sampled reachable pairs with hop at least two. Its reported sample comprises 20,000 pairs from 200 BFS sources. The neighbor score, reported as recall@10, is a micro average over non-isolated vertices that counts true neighbors in the ten closest vertices and divides by $\min(d_u, 10)$. It evaluates observed neighbor recovery. Hop R^2 uses a random forest on one Euclidean distance feature with an 80/20 pair split. Reconstruction AUC uses a random forest on Hadamard endpoint features, balanced negatives, and an 80/20 split on the embeddings with all positive edges included. Community NMI/ARI compare k-means with a Louvain partition of the input graph, using the Louvain community count as k.

Method settings. The broad Fodiwalk row uses a 10-by-20 walk budget, window five, cap 16, 200 epochs, and SQN with reported rate 0.999. The historical plain row uses the same nominal walk budget and cap, uniform walks, force constants $(k_1, k_r, k_4) = (0.999, 1, 1)$, a constant step 0.999, and a row-drop probability 0.5. DeepWalk uses 80-by-40 walks with window ten and node2vec uses 10-by-80 with window ten in their historical headline rows. The broad reference uses 256 landmarks and exact BFS distances before MDS; it has access to information unavailable to the walk-derived methods and is no mathematical upper bound on achieved correlation.

5 Results

Fodiwalk produces the highest distance-rank correlation of any tested method on Cora and PubMed at 128 dimensions, and its walk-witnessed gap weighting is the reason: replacing that weighting with a flat weight costs 0.187 and 0.232 ρ on those two graphs. Both results hold under qualification, developed below: an exact-distance oracle beats Fodiwalk on CiteSeer, the gap-weighting advantage reverses on the hierarchical WordNet graph, and no single metric orders every method the same way. The rest of this section separates what is well supported from what is graph- or metric-specific, starting with the headline geometry comparison and moving to the analyses that qualify it.

5.1 Sparse walk separation produces useful Euclidean rankings

The central empirical question is whether walk-witnessed coefficients h_{uv} improve on treating augmented pairs as one undifferentiated set: Table 1 compares Fodiwalk against thirteen baselines on distance-rank correlation ρ at 128 dimensions across three citation graphs.

Fodiwalk–SQN has the largest ρ on Cora (0.8656 ± 0.0016) and PubMed (0.7933 ± 0.0125), ahead of every other tested method including DeepWalk, node2vec, and the Force2Vec family. On CiteSeer, landmark MDS is substantially higher (0.9545 ± 0.0054 versus 0.8280 ± 0.0136); because

Table 2: Selected community agreement in the broad campaign. NMI/ARI compare Louvain partitions of the input graph with k-means partitions of each embedding; they do not measure ground-truth community accuracy.

	Cora		CiteSeer		PubMed	
Method	NMI	ARI	NMI	ARI	NMI	ARI
Fodiwalk–SQN	.6992	.3409	.8671	.3390	.5829	.3514
DeepWalk	.7057	.3180	.8117	.1451	.6276	.4431
node2vec	.6935	.2492	.8060	.1429	.6148	.3738
Force2Vec	**.7834**	.5006	.8629	.4153	–	–
tForce2Vec	.7814	**.5174**	**.9090**	**.5869**	.6435	.4428
Landmark MDS	.6289	.3002	.6565	.0015	.4710	.2317

landmark MDS consumes exact BFS distances as input, this is an exact-distance reference rather than a competing embedding method, and its lower Cora and PubMed values confirm it is not an empirical upper bound there. FODIWALK also has the highest broad-campaign hop R^2 on Cora and PubMed, with landmark MDS again leading on CiteSeer. Methods designed around inner products, such as ProNE and LINE, can retain reasonable edge-reconstruction accuracy while showing weak or negative Euclidean rank correlation on the same embeddings; the comparison above is specific to the distance-geometry use case the paper targets, not a general-purpose embedding ranking.

5.2 METRIC GEOMETRY DOES NOT SUBSUME LOCAL OR COMMUNITY STRUCTURE

Distance-rank correlation and hop R^2 test nonadjacent geometry. Local retrieval and community detection test different properties of the same coordinates, and the three do not agree on a winner. Table 2 reports Louvain-versus-k-means agreement (NMI/ARI) alongside the geometry results above.

Node2vec is marginally above FODIWALK on Cora neighbor score@10 (0.9579 ± 0.0008 versus 0.9573 ± 0.0020) and clearly above it on PubMed (0.9184 ± 0.0003 versus 0.7763 ± 0.0027); FODIWALK is highest on CiteSeer (0.9629 ± 0.0014). Community agreement is mixed in the same way: base Force2Vec has the largest Cora NMI and tForce2Vec the largest Cora ARI, tForce2Vec leads both CiteSeer cells, and PubMed splits between tForce2Vec (NMI) and DeepWalk (ARI, 0.4431 ± 0.0187, against tForce2Vec estimate of 0.4428). Edge reconstruction AUC is near saturation across the entire broad table (every method above 0.97) and is not discriminating here, because every method sees the full graph rather than a held-out split. FODIWALK's advantage is specific to nonadjacent distance geometry: it does not carry over to local retrieval or to community structure, where several baselines are competitive or better.

5.3 EMBEDDING DIMENSION CHANGES WHICH METHOD IS PREFERABLE

Table 3 sweeps embedding dimension from 16 to 128 for FODIWALK and DeepWalk on the three graphs.

Table 3: Distance-rank correlation ρ against embedding dimension.

	Cora		PubMed		WordNet	
Dim	FODIWALK	DeepWalk	FODIWALK	DeepWalk	FODIWALK	DeepWalk
16	$.767 \pm .011$	$.785 \pm .003$	$.700 \pm .005$	$.670 \pm .004$	$.239 \pm .008$	$.462 \pm .011$
32	$.809 \pm .007$	$.793 \pm .003$	$.746 \pm .005$	$.673 \pm .003$	$.284 \pm .007$	$.401 \pm .008$
64	$.831 \pm .007$	$.779 \pm .002$	$.769 \pm .004$	$.670 \pm .002$	$.320 \pm .009$	$.295 \pm .007$
128	$.843 \pm .006$	$.742 \pm .003$	$.779 \pm .003$	$.664 \pm .002$	$.348 \pm .010$	$.163 \pm .007$

FODIWALK's correlation rises with dimension on every graph shown, and it overtakes DeepWalk on Cora and PubMed by dimension 32. This order reverses on WordNet with DeepWalk leading at 16 dimensions (0.462 versus 0.239), the two cross between 32 and 64 dimensions, and FODIWALK leads by 128 (0.348 versus 0.163).

5.4 GAP WEIGHTING IS THE MECHANISM ON CITATION GRAPHS, AND IT REVERSES ON WORDNET

Table 4 isolates the coefficient rule directly, holding the walk policy and all other settings fixed and only replacing walk-witnessed minimum-gap weights h_{uv} with a flat weight of 1.

Table 4: Minimum-gap against flat weights at fixed support, distance-rank ρ.

Graph	Minimum-gap ρ	Flat-weight ρ	Difference
Cora	0.843 ± 0.006	0.656 ± 0.005	$+0.187$
PubMed	0.779 ± 0.003	0.547 ± 0.002	$+0.232$
WordNet	0.348 ± 0.010	0.393 ± 0.010	-0.045

Gap weighting is a large, consistent gain on the two citation graphs and a small loss on WordNet. Both citation-graph gaps are many times the experiments run spread. The WordNet gap is roughly $3\times$ the spread of the difference and should be treated as a smaller but still directionally supported effect, not as noise.

6 DISCUSSION AND LIMITATIONS

The broad comparison identifies a consistent *measured* advantage for Fodiwalk in Euclidean distance ordering among the tested learned methods on three citation networks. Node2vec and tForce2Vec are stronger on other diagnostics, and WordNet reverses its Fodiwalk–DeepWalk ordering as dimension changes. We therefore interpret the method as a useful inductive bias for sampled hop geometry, with consistently competitive performance on all other metrics.

The cap experiment suggests that restricting the pair table may prevent abundant, redundant walk observations from dominating the nonlocal geometry, even as extra pairs help retrieve immediate neighbors.

Several boundaries matter. The broad Fodiwalk cell was selected in a prior sweep; the alternatives do not have an equally documented tuning budget. Force2Vec permutations probe sensitivity to ordering, whereas other seeds probe their respective randomized implementations. Fixed evaluation draws make the measured seed spread useful for run variability but do not supply uncertainty over pair sampling or graphs. Hop R^2 and reconstruction also use embedding-time evaluation seeds in the broad report. More general claims need fresh graph splits or graphs, preregistered tuning, and paired intervals over independent evaluation draws. End-to-end efficiency comparisons must include walk generation, support construction, compilation, and scoring with identical timing boundaries.

Finally, a witnessed gap upper-bounds a reachable graph distance; supplemental coefficients do not. The implementation may leave some true edges at a gap above one and may derive mass from the number of weight-one entries rather than graph degree. These are material reproducibility conditions, not innocuous notation. Theoretical statements about reciprocal, degree-normalized plain updates do not establish convergence for the directed construction, the stochastic mask, or the SQN configuration used in the broad comparison.

7 CONCLUSION

Fodiwalk converts sampled walk separation into a sparse force field for Euclidean graph embedding. The comparisons shows strong distance–hop ordering on Cora and PubMed and a clear BFS-informed exception on CiteSeer. Neighbor retrieval, community agreement, dimension, and the cap sweep delimit that result. Controlled gap assignments and independent evaluation samples are the next tests of whether the observed geometric advantage specifically comes from walk-derived separation.

REFERENCES

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, 2001.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pp. 891–900, 2015. doi: 10.1145/2806416.2806512.

Vin de Silva and Joshua B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford University, 2004.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.

Hamidreza Lotfalizadeh. *Kinematics-based Force-Directed Graph Embedding*. PhD thesis, Purdue University, 2024.

Hamidreza Lotfalizadeh and Mohammad Al Hasan. Force-directed graph embedding with hops distance. In *IEEE International Conference on Big Data (BigData)*, pp. 2946–2953, 2023.

Hamidreza Lotfalizadeh and Mohammad Al Hasan. Kinematic-based force-directed graph embedding. In *Complex Networks XV*, pp. 137–149. Springer, 2024. doi: 10.1007/978-3-031-57515-0_11.

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

Mark Ortmann, Mirza Klimenta, and Ulrik Brandes. A sparse stress model. In *Graph Drawing and Network Visualization (GD 2016)*, 2016.

Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. doi: 10.1145/2939672.2939751.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. Don't walk, skip! online learning of multi-scale network embeddings. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pp. 258–265, 2017.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467, 2018.

Md. Khaledur Rahman, Majedul Haque Sujon, and Ariful Azad. Force2vec: Parallel force-directed graph embedding. In *IEEE International Conference on Data Mining (ICDM)*, 2020.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, 2015.

Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th international conference on world wide web*, pp. 287–297, 2016.

Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. Prone: Fast and scalable network representation learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 4278–4284, 2019. doi: 10.24963/ijcai.2019/594.

Ziwei Zhang, Peng Cui, Haoyang Li, Xiao Wang, and Wenwu Zhu. Billion-scale network embedding with iterative random projection. In *2018 IEEE International Conference on Data Mining*, pp. 787–796, 2018. doi: 10.1109/ICDM.2018.00094.