GraphWalker: Deployable Inertial-Based Indoor Localization via Trajectory to Graph Alignment

Maimoonah Al-Mashhadani
Carnegie Mellon University
maimoonm@andrew.cmu.edu

Yaqoob Ansari
Carnegie Mellon University
yansari@andrew.cmu.edu

Khaled A. Harras
Carnegie Mellon University
kharras@andrew.cmu.edu

Eduardo Feo-Flushing
Carnegie Mellon University
efeoflus@andrew.cmu.edu

Abstract

Inertial indoor localization is attractive because it requires no infrastructure and reveals no identifying signals, yet unbounded drift and the absence of a global coordinate frame prevent it from serving as a standalone solution. Existing map-constrained approaches mitigate drift using floor plans, but typically rely on dense geometric maps, calibrated probabilistic models, or offline global inference. We present GraphWalker, a real-time indoor localization system that aligns IMU-derived trajectories with coarse semantic floor plan graphs through online trajectory-to-graph alignment. GraphWalker maintains multiple candidate path hypotheses through bounded beam search and independently corrects drift for each hypothesis, requiring only a coarse semantic graph, a known starting pose, and commodity inertial sensing. On 152 real-world trajectories totaling 9 km of walking across three floor plans, GraphWalker improves localization accuracy by up to 50% over map-free and map-constrained baselines and reduces path errors by 42%, while maintaining real-time operation.

CCS Concepts

• **Information systems** → **Location based services**; • **Human-centered computing** → *Ubiquitous and mobile computing*.

Keywords

Indoor localization, inertial measurement unit, semantic graph alignment, beam search localization, infrastructure-free positioning

ACM Reference Format:
Maimoonah Al-Mashhadani, Yaqoob Ansari, Khaled A. Harras, and Eduardo Feo-Flushing. 2026. GraphWalker: Deployable Inertial-Based Indoor Localization via Trajectory to Graph Alignment. In *The 34th ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '26), November 03–06, 2026, Riverside, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3841645.3843004

1 Introduction

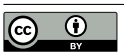
Estimating a person's location and movement through environments underpins applications spanning navigation [14], emergency response [13], and assistive technologies [3, 20]. Outdoors, this is routinely supported by global navigation satellite systems, but indoors, those signals are often unavailable or unreliable, and no comparably deployable alternative exists.

Dominant indoor localization approaches rely on dedicated infrastructure or environment-specific sensing, achieving strong accuracy at substantial deployment and maintenance cost. Wi-Fi fingerprinting, Bluetooth beacons, and ultra-wideband anchors [6, 7, 16] perform well within instrumented environments, however, they require expensive site surveys, periodic recalibration as buildings evolve [12], and remain vulnerable to signal degradation. Vision-based methods and SLAM instead exploit geometric information from cameras and dense scene reconstruction [17], but trade installed hardware for substantial offline preparation, and are sensitive to lighting, occlusion, and privacy constraints.

Among infrastructure-free approaches, inertial measurement units (IMUs) embedded in commodity smartphones and wearable devices offer perhaps the most deployable option, but suffer from unbounded drift that existing approaches only partially address. Classical pedestrian dead reckoning (PDR) estimates motion directly from inertial sensing but accumulates drift rapidly due to sensor noise and bias [5]. Learned inertial odometry systems substantially improve short-term motion estimation from raw IMU streams [9], yet still produce trajectories in an arbitrary local coordinate frame and remain susceptible to unbounded long-term drift. Learned absolute-location techniques, such as NILoc [8], instead regress position from velocity sequences, but require per-scene pretraining that does not transfer to unseen floor plans.

Prior work has attempted to constrain IMU drift through map matching using probabilistic formulations such as Hidden Markov Models [15], Conditional Random Fields [19], and particle filters [18]. While effective in structured settings, these approaches typically rely on dense geometric floor plans, carefully tuned observation models, or globally optimal inference over complete trajectories. Such assumptions limit applicability in practical deployments where only coarse structural maps are available, and localization estimates must be produced incrementally in real time.

To bridge this gap, we present GraphWalker, a real-time indoor localization system that aligns IMU trajectories with coarse semantic floor plan graphs. GraphWalker performs online trajectory-to-graph alignment through a bounded beam-search formulation that maintains multiple plausible path interpretations in parallel. By localizing incrementally as motion is observed, the system avoids global inference over complete trajectories, while remaining robust to accumulated drift and topological ambiguity. Unlike prior probabilistic map-matching approaches, GraphWalker operates directly on sparse semantic graphs rather than requiring dense geometric floor plans, calibrated observation models, or dedicated localization infrastructure. It maintains a bounded set of candidate path hypotheses online, scores each against the graph, and corrects heading drift per candidate rather than under one global orientation.



This work is licensed under a Creative Commons Attribution 4.0 International License.
SIGSPATIAL '26, Riverside, CA, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2950-8/2026/11
https://doi.org/10.1145/3841645.3843004

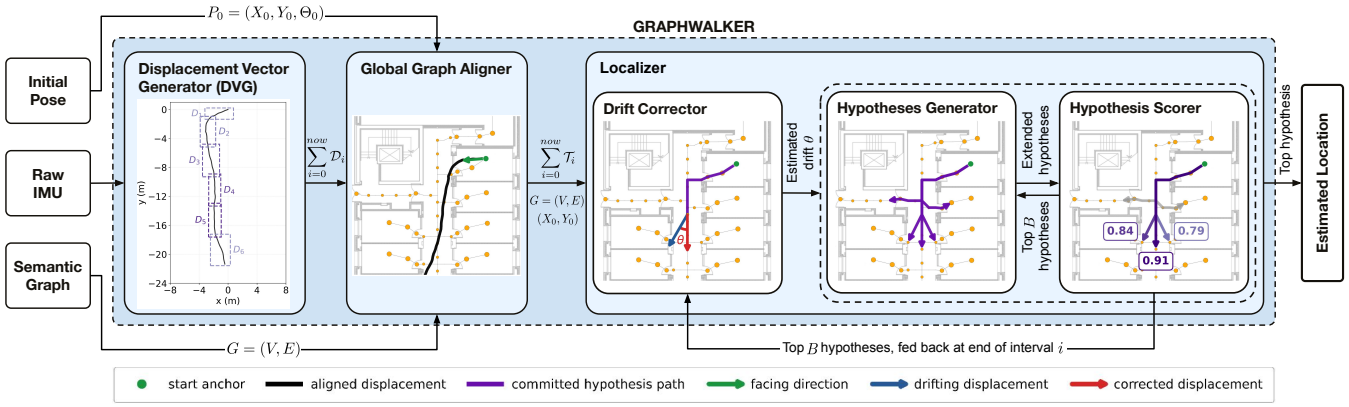


Figure 1: Architecture overview of GraphWalker. Raw incremental IMU, an initial pose P_0, and a semantic floor plan graph G are fed to the system. The DVG converts the IMU stream into displacement vectors $\mathcal{D}_i$; the Global Graph Aligner anchors them to the floor plan frame, yielding trajectory $\mathcal{T}_i$; the Localizer runs incremental beam search over G, maintaining B hypotheses and returning the highest-scoring path, whose terminal node is the estimated user location.

To evaluate GraphWalker, we collect 152 indoor walking trajectories spanning 9 km and 170 minutes across floor plans of varying structural complexity. We compare it against four baselines drawn from map-free and map-constrained paradigms. GraphWalker reduces localization error by 50% relative to the strongest online map-constrained baseline while maintaining real-time operation, and also reduces navigable-path error by 42%.

2 GraphWalker Architecture

The GraphWalker system is comprised of three stages (Fig. 1). The *Displacement Vector Generator* converts iteratively fed raw inertial samples into a sequence of displacement vectors. The *Global Graph Aligner* anchors these displacements to the floor plan coordinate frame using the known initial pose. The *Localizer*, the heart of the system, performs incremental beam search over the aligned displacement vectors as they arrive, maintaining B parallel hypotheses about the user's path through the graph. The entire pipeline ingests the IMU stream incrementally, enabling real-time online localization.

GraphWalker takes three inputs: a semantic floor plan graph, a stream of raw inertial measurements, and a known initial pose $P_0 = (X_0, Y_0, \Theta_0)$. The floor plan is formalized as a graph $G = (V, E)$, where nodes $v \in V$ denote semantic locations and edges $e \in E$ encode traversable connections with distance and bearing. Each node carries a semantic type $\tau(v) \in \{\text{room, door, corridor}\}$ reflecting its functional role within the building.

2.1 Displacement Vector Generator

The Displacement Vector Generator (DVG) converts the raw inertial stream into a sequence of body-frame displacement vectors $\mathcal{D}_i$, computed over the i-th IMU interval. Each interval is a fixed-duration chunk of samples processed by a single invocation of the pipeline. We use RoNIN [9], a learned inertial odometry model that regresses planar velocity from windowed accelerometer and gyroscope features; cumulative integration over the interval yields the displacement vectors consumed downstream.

2.2 Global Graph Aligner

The Global Graph Aligner places the displacement sequence into the input graph's coordinate frame so it can be compared against it. By construction, $\mathcal{D}_i$ carries no global pose; its origin and heading are arbitrary. The aligner resolves this using the known initial pose P_0 as anchor, yielding a trajectory $\mathcal{T}_i$. Each DVG sample is first divided by the graph's resolution, converting it from meters into the graph's coordinate units, then mapped by a single rigid transform comprising a planar rotation R and a translation $\mathbf{t}$, as $\mathbf{p}_i^{\text{map}} = R\,\mathbf{p}_i^{\text{imu}} + \mathbf{t}$, where $\mathbf{t} = (X_0, Y_0)$ places the first sample at the initial position and R rotates the DVG's initial motion direction onto the heading encoded by Θ_0. The transform is estimated once from the initial pose and then held fixed for the entire trajectory, so subsequent samples inherit it and progressively assemble into one continuous trajectory in the graph.

2.3 Localizer

The Localizer matches the anchored trajectory against the graph, seeking the semantically valid path $\mathcal{P}^* = (v_1, v_2, \ldots, v_k),\ v_j \in V$ through G that best explains $\mathcal{T}$. We match cumulative edge lengths to the odometric distances in $\mathcal{D}$ and edge headings to the trajectory's directions; the terminal node v_k is the estimated user location. It is initialized with a hypothesis rooted at the node closest to (X_0, Y_0), and runs an incremental beam search that consumes $\mathcal{T}_i$ one interval at a time. Each interval passes through three components in sequence, after which the top-B survivors form the updated beam.

Drift Corrector. Slow inertial heading drift accumulates over long trajectories [2], causing the estimated direction of motion to diverge from the true direction in the floor plan frame even though the global translation anchor remains correct. At the start of each IMU interval, every active hypothesis estimates its θ as the signed angle between the direction of its most recently committed graph edge and the direction of the corresponding aligned-trajectory interval. Because correction is per-hypothesis, parallel hypotheses with different committed paths produce different rotation estimates, and the system never commits to a single global drift estimate that could bias the beam. The estimate is held fixed for the duration of

the interval; re-estimating θ at every candidate expansion would fit the offset to each speculative move and destabilize the ranking.

Hypotheses Generator. Each hypothesis in the beam is extended to the graph nodes it could plausibly have reached, subject to the graph's traversability rules. Explicitly encoded edges must follow the semantic chain *room* ↔ *door* ↔ *corridor*, with same-type transitions permitted between room nodes (for multi-node room representation) and between corridor nodes but not between door nodes. To accommodate open hallway regions not fully captured by explicit edges, two corridor nodes may also transition directly if their Euclidean distance does not exceed a configurable radius r. Formally, the set of valid transitions is:

$$\mathcal{A} = E \cup \{(u, v) \mid \tau(u) = \tau(v) = \text{corridor},\ \|p_u - p_v\| \le r\} \quad (1)$$

For each active hypothesis, the Generator sorts each candidate move into one of two outcomes: *queued* for immediate scoring if enough odometric distance remains to cover the edge, or *deferred* to a pending pool with the outstanding deficit recorded. Pending moves are promoted once newly accumulated distance covers that deficit, letting a single graph edge span multiple intervals so that long corridor traversals need not align with interval boundaries.

Hypotheses Scorer. Each queued move is scored against two criteria: local heading alignment and global trajectory consistency. For a move from u to v, the aligned-trajectory displacement is rotated by the hypothesis's current drift estimate θ to produce a frame-corrected heading $\hat{d}_{\text{IMU}}$, which is compared against the unit edge direction $\hat{d}_{\text{graph}}$. The *alignment* score captures local heading agreement, $a = \hat{d}_{\text{IMU}} \cdot \hat{d}_{\text{graph}} \in [-1, 1]$. The *trajectory similarity* score σ captures whether the shape of the hypothesis's recent committed path is consistent with that of the recent aligned trajectory. It is computed over a look-back window applied identically to both, keeping the score responsive to current movement and preventing early path decisions from dominating later estimates. The window length is *adaptive*: it begins at a base distance ℓ and grows with the user's recent walking pace, bounded above at 2ℓ. $\sigma = 0.6\,\sigma_{\text{eff}} + 0.4\,\sigma_{\text{dir}}$ combines an *efficiency ratio* σ_{eff}, which compares how directly each trajectory moves toward its net displacement, and a *direction agreement* σ_{dir}, the cosine of the net-displacement vectors rescaled to $[0, 1]$ so both terms share a common range. For each queued move, the updated hypothesis score is:

$$s' = 0.5\,s + 0.25\,a + 0.25\,\sigma \quad (2)$$

where s is the score carried from the hypothesis's previous expansion. After scoring, hypotheses are ranked and truncated to a fixed width B, so the number carried forward stays fixed regardless of graph size or trajectory length. If odometric distance remains in the current interval, the beam re-enters the Generator; once the interval is exhausted it carries over to the Drift Corrector at the start of the next interval. When the final interval is processed, the terminal node of the top-ranked hypothesis is returned as the estimated location, and its node sequence as the estimated trajectory.

3 Experimental Setup

Dataset. The dataset comprises 152 indoor walking trajectories collected across three floor plan sections, totaling approximately 170 minutes of walking and 9,079 m, with individual trajectories ranging from 16 to 161 m. The three sections span a range of spatial layouts: FP_1 (42 trajectories) is an office environment with a regular grid-like corridor structure, FP_2 (64) is more asymmetric and open, and FP_3 (46) combines open regions with grid-like corridor segments. We manually constructed the semantic graph for each section from building blueprints, independently of the collected trajectories, with nodes at room centroids, doorways, and corridor waypoints. Five volunteers carried a Samsung Galaxy S20 5G close to the chest throughout each walk, following a navigation task from a start point A to a destination B, with a subset of walks including an intermediate stop at C. Endpoints were chosen so that natural walking would involve turns, U-turns, and pauses, the motion features that most degrade inertial measurement; beyond them we gave no guidance on route or pace. Ground truth came from manual waypoint annotation by an observer following close behind each participant, and all trajectories were initialized from a manually recorded ground-truth pose.

Metrics. Each ground-truth waypoint at time t is paired with the predicted node committed within ±3 s of t that lies spatially closest, allowing tolerance for an online system to fall behind or jump ahead. *Trajectory accuracy* is the Euclidean distance between each predicted node and its matched ground-truth coordinate, reported as mean and root-mean-square error (RMSE). *Waypoint recall* is the fraction of waypoints matched within 1 m. *Shortest-path error* is the distance along the graph between each predicted and matched ground-truth node, capturing structural error that Euclidean distance cannot. Endpoint accuracy is captured via *end error*, the distance between final predicted and ground-truth positions, and *end match*, satisfied when the predicted endpoint is a room node in the same room cluster as ground truth and reachable within 5 m along the graph. The *latency ratio* is execution time over trajectory duration, where below 1.0 is faster than real time.

Configuration and Baselines. Unless otherwise specified, experiments use the default configuration of GraphWalker: 5 s IMU intervals, beam width B=50, base look-back distance ℓ=12.6 m, and corridor transition radius r=2.2 m. We compare against four baselines spanning map-free and map-constrained, online and offline paradigms. *PDR* estimates motion through step detection and heading integration, with Kalman smoothing and zero-velocity updates [4]. *RoNIN Overlay* anchors the learned odometry trajectory [9] to the floor plan frame using the known initial pose, with no topological constraints. *Particle Filter (PF)* weights particles by geometric agreement with nearby graph structure and resamples to discard low-probability hypotheses. *CRF* is an offline sequence-level baseline solved by Viterbi decoding over a chain-structured CRF [11]. Learned absolute-location methods such as NILoc [8] are excluded, since their per-scene pretraining falls outside the deployment setting GraphWalker targets. No site-specific tuning or learned transition calibration is applied to any baseline.

4 Results

Across the three floor plans, GraphWalker achieves the lowest average mean error, the highest recall, and the highest endpoint match rate overall (Table 1). Its 2.36 m mean localization error corresponds to a 30% reduction relative to the strongest map-free baseline (RoNIN, 3.37 m), and a 50% reduction relative to the strongest online map-constrained baseline (PF, 4.77 m). On the more irregular

Table 1: Evaluation results across all 152 trajectories, with per-layout mean error below the aggregates. Latency is measured for GraphWalker only and is not comparable across systems. Bold: best per row.

	System				
Metric	GW	PDR	RoNIN	PF	CRF
Mean err. (m) ↓	**2.36**	11.71	3.37	4.77	16.01
RMSE (m) ↓	**3.17**	14.61	4.26	6.23	18.53
Recall ↑	**0.74**	0.19	0.38	0.41	0.09
Path err. (m) ↓	**3.63**	17.22	5.73	6.23	22.71
End err. (m) ↓	**6.25**	24.51	7.12	10.80	26.40
End Match ↑	**0.54**	0.01	0.07	0.12	0.03
Mean err. FP_1 (m) ↓	**3.96**	11.10	4.20	8.01	17.17
Mean err. FP_2 (m) ↓	**1.53**	10.07	2.87	4.08	12.38
Mean err. FP_3 (m) ↓	**2.02**	14.57	3.28	2.71	19.96
Latency ratio	0.19	—	—	—	—

FP_2 and FP_3 environments, GraphWalker achieves mean errors of 1.53 m and 2.02 m respectively, while on the more regular FP_1 layout the gap narrows (3.96 m for GraphWalker vs. 4.20 m for RoNIN).

The baselines exhibit different failure modes. PDR accumulates substantial drift across all layouts (10.07–14.57 m mean error) and rarely recovers the correct endpoint. RoNIN substantially reduces this drift through learned odometry, but without map awareness its endpoint localization remains limited. PF benefits from structural constraints and achieves moderate recall (0.41 average), but still reaches 4.77 m mean error. CRF, evaluated without site-specific transition calibration, performs worst of all five systems, which we attribute to the uncalibrated protocol rather than to a limitation of CRF map matching in general.

Endpoint and topological accuracy both highlight the effect of incorporating global path consistency during search. GraphWalker achieves a 54% endpoint match rate overall, compared with at most 12% for any baseline (PF), and an average shortest-path error of 3.63 m against 5.73 m for RoNIN and 6.23 m for PF, suggesting that the recovered trajectories are generally more consistent with the underlying floor plan structure. Finally, GraphWalker maintains real-time performance throughout evaluation, with latency ratios between 0.15 and 0.30 across all floor plans and an average of 0.19.

Two patterns emerge from these results. First, the aggregate table conceals one exception: on FP_1 alone, RoNIN attains marginally lower RMSE (5.22 m vs. 5.45 m) and endpoint distance (8.50 m vs. 10.67 m). This is a structural property of graph-constrained localization rather than a defect: GraphWalker must terminate at a graph node, so committing to the wrong parallel corridor lands a full corridor-spacing from the truth, whereas unconstrained drift can sit closer in Euclidean terms while still being in the wrong room. Endpoint match on the same layout resolves the apparent contradiction (0.381 vs. 0.167), and for applications that act on the user's identified location, topological correctness is the metric that matters. Second, error is consistently highest on the grid-like FP_1 and lowest on the irregular FP_2: regular layouts make candidate paths locally indistinguishable and force the beam to commit at each turn without lookahead, while irregular layouts offer asymmetries that make the same evidence more discriminating. Structural regularity of the environment, rather than trajectory length, is therefore the dominant factor in localization difficulty.

5 Conclusion

We presented GraphWalker, which localizes indoors by aligning IMU-derived trajectories to coarse semantic floor plan graphs through an incremental beam search with per-hypothesis drift correction. Across 152 real-world trajectories it outperforms every map-free and map-constrained baseline while operating in real time, showing that coarse semantic structure alone sufficiently constrains drifting inertial trajectories. Results are confined to one building and one carrying position, and assume a known initial pose whose error tolerance we do not characterize. Future work will address these limitations, extend the formalism to multi-floor environments, integrate sparse opportunistic signals for structurally regular layouts, and replace manual graph construction with automated extraction from floor plans [1] or navigational text [10].

Acknowledgments

The authors used generative AI tools to assist with language polishing, and to generate and debug supporting code for data processing, figure generation, and parts of the system implementation.

References

[1] Y. Ansari, A. Karkour, E. F. Flushing, and K. A. Harras. 2025. Tesseract: Unfolding navigable graph representations from low-semantic floor plans. In *ACM SIGSPATIAL*.
[2] J. Borenstein and L. Ojeda. 2010. Heuristic Drift Elimination for Personnel Tracking Systems. *Journal of Navigation* (2010).
[3] Aliaa Essameldin, Mohamed Nurulhoque, and Khaled A Harras. 2020. More than the sum of its things: Resource sharing across IoTs at the edge. In *ACM/IEEE SEC*.
[4] E. Foxlin. 2005. Pedestrian Tracking with Shoe-Mounted Inertial Sensors. *IEEE Computer Graphics and Applications* (2005).
[5] R. Harle. 2013. A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys & Tutorials* (2013).
[6] Omar Hashem, Khaled A Harras, and Moustafa Youssef. 2021. Accurate indoor positioning using IEEE 802.11 mc round trip time. *Pervasive and Mobile Computing* 75 (2021), 101416.
[7] O. Hashem, M. Youssef, and K. A. Harras. 2020. Winar: RTT-based sub-meter indoor localization using commercial devices. In *IEEE PerCom*.
[8] S. Herath, D. Caruso, C. Liu, Y. Chen, and Y. Furukawa. 2022. Neural Inertial Localization. In *CVPR*.
[9] S. Herath, H. Yan, and Y. Furukawa. 2020. RONIN: Robust Neural Inertial Navigation in the Wild. In *ICRA*.
[10] A. Karkour, K. A. Harras, and E. Feo-Flushing. 2024. Text2Map: From Navigational Instructions to Graph-Based Indoor Map Representations Using LLMs. In *IEEE IROS*.
[11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*.
[12] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen. 2015. A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned. In *IPSN*.
[13] S. Mostafa, K. A. Harras, and M. Youssef. 2023. Unicellular: An accurate and ubiquitous floor identification system using single cell tower information. In *ACM SIGSPATIAL*.
[14] Sherif Mostafa, Moustafa Youssef, and Khaled A Harras. 2024. ModeSense: Ubiquitous and accurate transportation mode detection using serving cell tower information. In *ACM SIGSPACIAL*. 184–195.
[15] P. Newson and J. Krumm. 2009. Hidden Markov Map Matching Through Noise and Sparseness. In *ACM SIGSPATIAL*.
[16] S. Subedi and J.-Y. Pyun. 2020. A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies. *Sensors* (2020).
[17] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. 2018. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR*.
[18] T. Willemsen, F. Keller, and H. Sternberg. 2022. Real-Time Map Matching with a Backtracking Particle Filter Using Geospatial Analysis. *Sensors* (2022).
[19] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. 2014. Lightweight Map Matching for Indoor Localisation Using Conditional Random Fields. In *IPSN*.
[20] F. Zafari, A. Gkelias, and K. K. Leung. 2019. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys & Tutorials* (2019).