RoVaR: Robust Multi-agent Tracking through Dual-layer Diversity in Visual and RF Sensing

MALLESHAM DASARI*, Carnegie Mellon University, USA
RAMANUJAN K SHESHADRI, NEC Laboratories America, USA
KARTHIKEYAN SUNDARESAN, Georgia Institute of Technology, USA
SAMIR R. DAS, Stony Brook University, USA

The plethora of sensors in our commodity devices provides a rich substrate for sensor-fused tracking. Yet, today's solutions are unable to deliver robust and high tracking accuracies across multiple agents in practical, everyday environments – a feature central to the future of immersive and collaborative applications. This can be attributed to the limited scope of diversity leveraged by these fusion solutions, preventing them from catering to the multiple dimensions of *accuracy*, *robustness* (diverse environmental conditions) and *scalability* (multiple agents) simultaneously.

In this work, we take an important step towards this goal by introducing the notion of *dual-layer diversity* to the problem of sensor fusion in multi-agent tracking. We demonstrate that the fusion of complementary tracking modalities, – passive/relative (e.g. visual odometry) and active/absolute tracking (e.g.infrastructure-assisted RF localization) offer a key first layer of diversity that brings scalability while the second layer of diversity lies in the *methodology* of fusion, where we bring together the complementary strengths of algorithmic (for robustness) and data-driven (for accuracy) approaches. RoVaR is an embodiment of such a dual-layer diversity approach that intelligently *attends* to cross-modal information using algorithmic and data-driven techniques that jointly share the burden of accurately tracking multiple agents in the wild. Extensive evaluations reveal RoVaR's multi-dimensional benefits in terms of tracking accuracy, scalability and robustness to enable practical multi-agent immersive applications in everyday environments.

CCS Concepts: • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools.*

Additional Key Words and Phrases: Localization, Tracking, Sensor Fusion, Machine Learning

ACM Reference Format:
Mallesham Dasari, Ramanujan K Sheshadri, Karthikeyan Sundaresan, and Samir R. Das. 2023. RoVaR: Robust Multi-agent Tracking through Dual-layer Diversity in Visual and RF Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 8 (March 2023), 25 pages. https://doi.org/10.1145/3580854

1 INTRODUCTION

Tracking multiple agents (humans and robots) in real life within a given space is a foundational problem in immersive and interactive applications. The growing availability of visual (cameras), inertial (IMUs), and RF (WiFi, BLE, UWB) sensors on our everyday devices provides a rich substrate for effective tracking. However, tracking solutions that are robust, accurate and cost-effective in realistic environments remain elusive.

*The work was done while the author was an intern at NEC Labs America and pursuing PhD at Stony Brook University.

Authors' addresses: Mallesham Dasari, malleshd@andrew.cmu.edu, Carnegie Mellon University, USA; Ramanujan K Sheshadri, ram@nec-labs.com, NEC Laboratories America, USA; Karthikeyan Sundaresan, karthik@ece.gatech.edu, Georgia Institute of Technology, USA; Samir R. Das, samir@cs.stonybrook.edu, Stony Brook University, USA.

2474-9567/2023/3-ART8
https://doi.org/10.1145/3580854

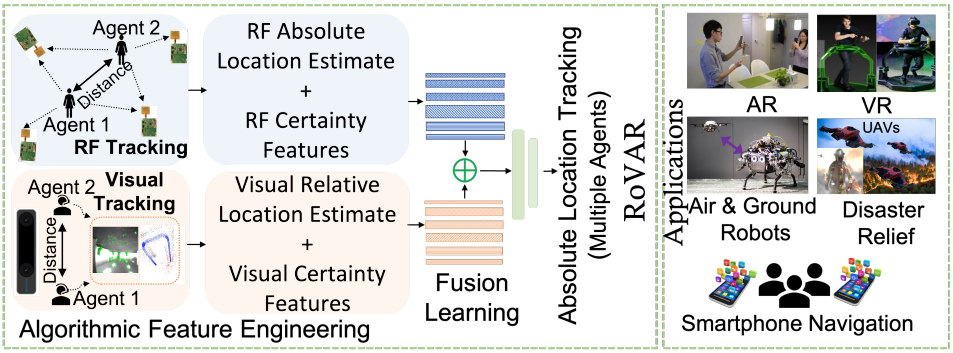


Fig. 1. Proposed tracking system and its applications.

Limitations of Today's Tracking Solutions. Existing tracking solutions can be broadly categorized as: passive and active (infrastructure-assisted) tracking. Passive solutions [13, 14, 49, 61] are largely odometry-based relative tracking methods that commonly use cameras and inertial sensors. While these passive/relative solutions are cost-effective and can track to within few tens of *cm* accuracy under favorable conditions, they are significantly vulnerable (1m+ error) when facing everyday environmental conditions like dim-light, texture-less surfaces, etc. [20, 48]. Further, their reliance on *relative* tracking prevents them from robustly recovering from unfavorable events, while also limiting the use of multi-agent tracking within one global frame of reference. On the other hand, active tracking approaches [25, 38, 40, 46] use planned deployment of one or more anchors to locate agents. They enable multi-agent tracking through *absolute* localization which also eliminates error accumulation. However, these active/absolute tracking solutions often face a tradeoff in operational range and accuracy, and typically offer lower accuracies than passive tracking. Therefore, most of the current generation commercial solutions (e.g., ARCore [31] and ARKit [9]) continue to adopt passive tracking and face aforementioned limitations when deployed in practice.

Need for Dual-layer Diversity. This work advocates the need for a hybrid tracking approach that can effectively bring together the complementary benefits of *active (absolute, multi-agent)* and *passive (relative, high resolution)* tracking to enable *scalable and accurate multi-agent* tracking.[1] While this forms the first layer that leverages diversity in the sensor tracking *modalities*, this is not sufficient. Conventional *algorithmic* sensor-fusion approaches (e.g. Kalman [19, 55, 57] and Particle filters [59]) that offer this first layer of diversity are robust to operate in various environments in the wild, but are unable to effectively sift out the erroneous artifacts of individual sensors and fail to provide effective fusion that can deliver high accuracies sustainably. On the other hand, recent *data-driven* approaches (e.g. milliEgo [35], VINET [17]) show significant promise for much better fusion, however, they have thus far targeted only passive/relative tracking in a small area, with a black-box learning approach that often face nontrivial challenges when deployed in the real-world (§2.3). Hence, the fundamental challenge in realizing our vision for a *robust, accurate, and scalable* tracking solution lies in not only leveraging the diversity offered by the two modalities of tracking (active and passive), but also the diversity offered by the methodology of fusion – in particular, figuring out when and how to leverage statistical learning in combination with closed-form algorithms.

RoVaR: We propose RoVaR (Fig. 1): an accurate and robust multi-agent tracking solution sufficiently lightweight to run in real-time on a low-end portable GPU device. RoVaR embodies effective fusion through diversity at two layers: it brings together the complementary benefits of active (RF - WiFi/UWB) and passive (visual) tracking modalities through an intelligent combination of both algorithmic and data-driven techniques. In

[1]We consider 2D position tracking in this work to focus on the fusion methodology. Details on extending it to 3D tracking can be found in §7.

RoVaR, agents carry a tracking device (e.g., headset, smart device) that houses an embedded camera (either stereo or monocular) and an RF interface. While the camera enables visual odometry-based relative tracking, the RF interface enables absolute tracking by estimating ranges (and angles, AoA) to one or more access points (anchors) in the environment. RF technologies like WiFi, and more so UWB, offer a good balance between tracking resolution (160 MHz with 802.11mc, 500 MHz with UWB) and robust NLoS operation, and are already deployed in many commodity access points and smartphones (Apple iPhone12, Samsung Galaxy S21, etc.) [6, 7]. While RoVaR employs UWB for RF tracking modality in this work, its approach is equally applicable to other RF technologies (e.g., WiFi).

RoVaR leverages algorithmic solutions to estimate the absolute location estimate from RF (e.g. multi-lateration [16, 24] or range+AoA [23] approaches) and the relative translation estimate from visual (e.g. ORB-SLAM3 [14]), while data-driven model assists in data filtering, feature composition and the fusion itself. RoVaR allows individual sensor algorithms to provide estimates along with their certainties to the fusion model. This leverages the physics and geometry inherent to these localization problems and also relieves the fusion model from trying to learn purely from raw data alone, allowing it to focus more on the effective fusion. This also reduces the compute and latency requirements enabling RoVaR to run in real-time even on resource-constrained devices, compared to compute-intensive deep learning/blackbox approaches. More importantly, these pure blackbox approaches are also limited in their ability to generalize. In contrast, RoVaR's approach contributes to a robust model that generalizes well for deployments in untrained environments.

RoVaR is built as a hand-held prototype comprising a UWB beacon and Intel T265 stereo-camera, experimented in diverse environments spanning 4500 sq.ft. area at multiple locations for a total of 232,000 tracked data points. Comprehensive evaluations over diverse settings reveal the importance of RoVaR's proposed dual-layer diversity framework, where active RF tracking eliminates the accumulation of errors, while leveraging the high resolution offered by the visual sensor, offering a median tracking accuracy of 15 cm. This is in contrast to 40 and 32 cm accuracy offered by RF and visual tracking solutions respectively in isolation. RoVaR requires small memory footprint of 5MB, 60% less training data, and a 5X latency reduction, 3X less power consumption compared to the next best alternative (blackbox solution), enabling real-time operation on mobile platforms like Jetson Nano/TX2 [5]. In summary, our contributions in this work are the following:

(1) We advocate the need for dual-layer diversity in both *tracking modality* (active + passive) and *fusion methodology* (algorithmic + data-driven) for accurate, robust, and scalable tracking in everyday environments (§2).

(2) We design and build RoVaR as an instantiation of our lightweight, yet robust and accurate framework that embodies the dual-layer diversity approach to fuse visual and RF sensing (§4).

(3) We comprehensively evaluate and showcase the benefits of RoVaR-style fusion models that enable both robustness and real-time operation[2] (§6).

2 BACKGROUND AND MOTIVATION

2.1 Related Work

2.1.1 Active/Absolute Tracking: Active tracking techniques use pre-deployed anchors at known locations to continuously localize and track a device. Popular solutions include RF multi-lateration techniques [16, 24] or recent deep learning methods [11]. There are other works using optical (e.g., IR beacons [28, 30], VLC [36]) and acoustic sensors [18, 34, 42], however, these solutions are either limited to LoS scenarios or highly sensitive to ambient noise [27], material attenuation (e.g., wood, concrete) [18, 53], and certain environmental conditions (e.g., temperature, humidity) limiting them to a room-level application. In comparison, RF signals are more robust in LoS and NLoS indoor environments. Consequently, prior works mainly leverage RF solutions like WiFi [29],

[2]We released our dataset (UWB, Camera and RFID) at https://github.com/mdasari823/RoVAR, to foster this line of research in tracking.

Table 1. Choice of Passive Tracking solutions.

	Mean	Std	Max
ARCore (Pixel2 Phone) [31]	1m	50cm	> 5m
ORBSLAM3 (KITTI Dataset) [14]	10cm	5cm	1.5m
RGBD SLAM (Intel D435 Depth Cam) [14]	50cm	20cm	2m
Lidar (Intel L515) [61]	40cm	20cm	2m

Table 2. Choice of RF Solutions - Range Vs. Accuracy.

	LTE [38]	WiFi [25]	UWB [46]	mmWave [40]
Accuracy	20m	5m	25cm	1cm
Range	1km	40m	30m	10m

UWB [37] and mmWave [39, 60] for active tracking.

2.1.2 Passive/Relative Tracking: Passive odometry-based solutions do not require pre-deployed infrastructure. Visual and inertial odometry are perhaps the most popular in this category. Visual Odometry (VO) algorithms rely on changes in texture, color and shape in successive camera images of a static environment to track the motion with high-precision (cm-scale). Popular VO solutions include ORBSLAM3 [14], and recent deep learning methods (e.g., DeepSLAM [33], VINET [17]). However, majority of these solutions suffer in environments with poor lighting, lack of texture or in conditions that enable perceptual aliasing [50, 52].

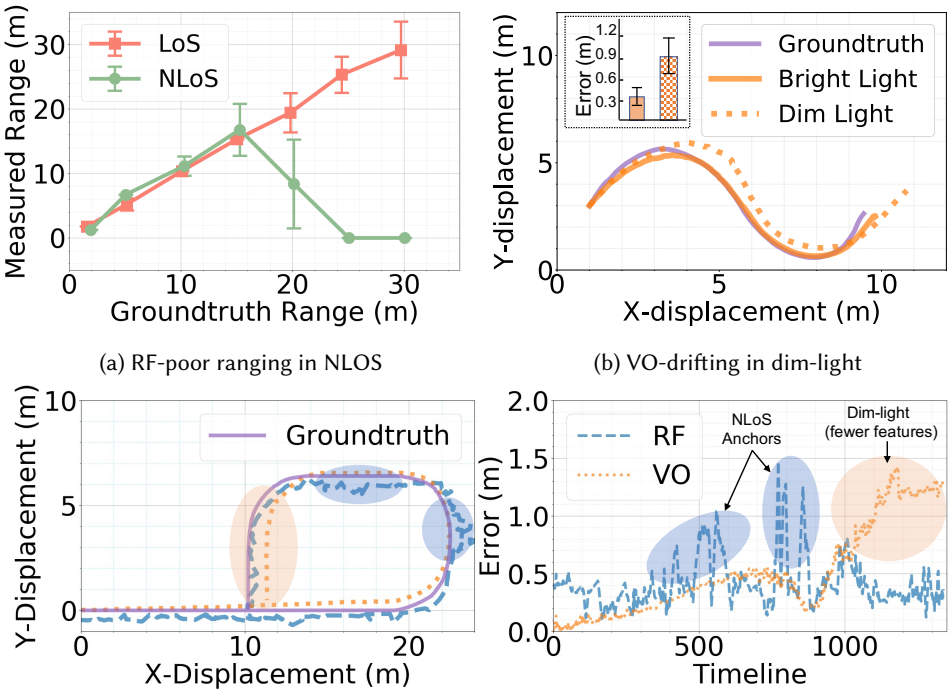
2.1.3 Sensor Fusion for Tracking: Sensor fusion has been long studied in the past to compensate for the noise originating from individual sensors. Solutions in this space include either algorithmic (e.g., WiFi+Camera [22], IMU+WiFi [56]) or deep learning methods (e.g., milliEgo [35], Camera+IMU [15, 17], DeepTIO [47]). While the algorithmic approaches are robust to diverse environments but struggle to deliver high tracking accuracies, the deep learning methods often fail to robustly generalize to untrained environments while offering high accuracy in trained environments (see §2.3).

2.2 Choice of Tracking Techniques

Although RoVaR's design is applicable to other RF and visual tracking modalities, we rationalize the following choices.

2.2.1 UWB-based Active Tracking: UWBs, compared to other RF technologies, offer a good balance between coverage (35-40m in LoS and 25-30m in NLoS) and ranging accuracy (Table 2). Consequently, there has been a spurt of consumer devices (e.g., IPhone 12, Samsung Galaxy S21 etc.) using UWB chips for indoor tracking. Future UWB chips [8] are expected to offer multi-antenna (single anchor) solutions that enable both ranging and AoA estimation. RoVaR can easily accommodate WiFi FTM [25] (in sub-6 GHz or mmWave band) as well.

2.2.2 ORB-SLAM3 for Passive Tracking: ORB-SLAM3 [14] is a feature-based SLAM algorithm, robust to motion clutter and can work with monocular, stereo or RGB-D images. The ORB-features [45] allow for fast matching across camera frames to enable real-time tracking. Previous studies show that ORB-SLAM algorithms can significantly outperform other popular SLAM algorithms in odometry detection [14, 43] (see Table 1). Nonetheless, in §4 we discuss how RoVaR can just as easily support other SLAM/VO algorithms.



(a) RF-poor ranging in NLOS

(b) VO-drifting in dim-light

(c) Complementary benefits of VO and RF

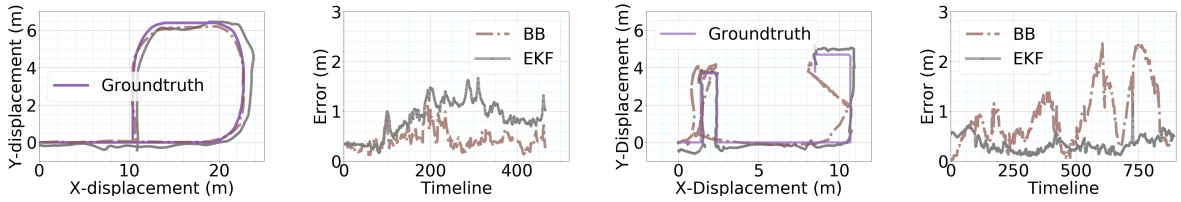
Fig. 2. Deficiencies of Active and Passive Tracking: a) RF suffering from NLoS conditions, b) VO suffering from poor lighting, c) Example trajectory with groundtruth and tracking error of RF and VO.

2.3 Limitations of Current Generation Tracking & Fusion Solutions

2.3.1 Individual Limitations of Multiple Sensors: VO algorithms' *relative* tracking – despite some of its intrinsic optimizations (e.g., loop closure detection) and high accuracy – is vulnerable to error accumulation over time. Once an error sets in, the relative measurements of the VO propagate the error forward, resulting in significant drifts in the estimated trajectories (Fig. 2b). UWB-based active tracking is immune to such drifts as each location estimation is independent of the previous ones. Errors that occur due to incorrect ranging estimates (e.g., NLoS bad anchors, see Fig. 2a) do not propagate, allowing UWB to provide better accuracy in *absolute* tracking over time. However, UWB can only provide sub-meter scale coarse location tracking compared to VO's cm-scale finer estimates. An effective fusion between the two sensors can bring complementary benefits to one-another, delivering a system capable of highly accurate absolute tracking over a sustained period of time. Fig 2c shows the potential for fusion between UWB and VO (ORB-SLAM3) for a simple trajectory (details are in §5), part of which is dimly lit (adverse to VO), while another part contains NLoS anchors (adverse to UWB)[3].

2.3.2 Scalability Across Multiple Users: Accurately tracking multiple users with respect to one another (even in NLoS) is the key to realizing multi-user collaborative applications. However, VO algorithms track users relative to their own individual starting point (needing a common origin in case of multiple users). Unless all users start at a common origin and are in visual LoS of each other with similar device/camera capabilities (unrealistic expectation), it is infeasible to estimate users' locations with respect to one another. In contrast, RF's active

[3]We note that there is limited prior work on UWB+VO fusion. This work takes a first attempt to showcase the benefits of fusing VO and UWB sensors.



(a) Trained Environment (left: trajectory, right: error) (b) Unseen Environment (left: trajectory, right: error)

Fig. 3. Limitations of Algorithmic tracking (RF+VO using EKF) and Deep learning based blackbox (BB) solutions. While BB performs better than classical solutions in trained environment, it fails in a new unseen place.

tracking provides absolute location estimates of each user with respect to a known RF anchor's physical location. Fusing VO's relative tracking with RF's absolute location estimates can enable accurate tracking in a global reference point (RF anchor) as well as relative to one-another.

2.3.3 Limitations of Today's Sensor Fusion Methodologies: Current sensor fusion methods can be broadly divided into two classes of solutions: 1) algorithmic or 2) data-driven models. Algorithmic solutions (e.g., Kalman Filter (KF) [55] and Bayesian Particle Filters [59]) aim to minimize statistical noise using time series data from individual sensors. However, we observe that these approaches consistently under-perform when fusing UWB and ORB-SLAM3. The KF requires system model to be linearized and noise distribution to be Gaussian, which often is not true for VO and RF real-life measurements, either due to sensor hardware and/or environmental artifacts (e.g. scene, NLoS issues). As an example, Fig. 3a shows the performance of Extended Kalman Filter (EKF) for fusing ORB-SLAM3 and UWB on one of our trajectories (§6). The fused location is quite erroneous (a median error of `80 cm`), leaving a significant room for improvement. On the other hand, Particle filters require a very large number of samples to accurately estimate the posterior probability density of the system. In complex environments, this results in particle depletion and consequently poor performance.

Deep learning based fusion solutions are shown to be effective in identifying nonlinearities, thus accurately fuse multiple sources of information. Popular recent solutions include fusing passive sensors such as VO+IMU fusion [17], mmWave+IMU fusion [35]. These blackbox (BB) models are trained on large-scale raw sensor data to obtain relative location or pose estimates. Porting these solutions to fuse raw RF and VO data, to predict *absolute* position puts a lot of burden on the model in capturing the complex geometric problem structure inherent to RF and camera based localization. This in-turn makes the model heavy, and overly reliant on input data distributions resulting in poor performance in untrained environments. Fig. 3b shows the performance of a BB model (model details in §4.4) that fuses UWB and camera raw data to estimate absolute location in an unseen environment. It is evident that pure deep learning (BB) models, while delivering superior performance in trained environments (see 3a), fail to generalize when deployed in unseen environments in the wild. More importantly, the problem with deploying these BB models in practice is that extremely slow and energy hungry on mobile devices. Because of these practical limitations, industry-grade solutions [9, 31] still rely on classical filtering approaches for real-time tracking and energy efficiency, compromising on accuracy[20].

In summary, while algorithmic solutions bring robustness to operation in various (untrained) environments but are unable to deliver effective fusion, data-driven approaches deliver high accuracy through superior fusion. This motivates the need for a second layer of diversity that brings together the strengths of both algorithmic (generalizability from robustness) and data-driven (high accuracy from effective fusion) approaches, while the first layer of diversity is offered by sensing modalities.

3 CHALLENGES IN REALIZING ROVAR'S DUAL-LAYER DIVERSITY

Building a fusion approach that incorporates RoVaR's dual-layer diversity entails addressing several critical questions.
(1) Given the pros and cons of active and passive tracking, how should their measurements be fused to automatically overcome sensor hardware and environmental artifacts and deliver both accuracy and scalability?
(2) How should the algorithmic and data-driven approaches split the burden of the sensor fusion pipeline to complementarily bring together their accuracy and robustness benefits?
(3) Can the resulting approach be light-weight for real-time operation on resource-constrained device platforms?

RoVaR adopts a systematic approach towards addressing these challenges to deliver a *robust, accurate and scalable* tracking solution.

4 ROVAR: DESIGN

4.1 Overview of RoVaR

RoVaR incorporates dual-layer diversity by (a) employing an algorithm-driven approach in the first stage to engineer features from individual RF (UWB ranges) and Visual (camera frames) sensor inputs; (b) followed by a data-driven (machine learning, ML) approach for effective fusion of these complementary sensor features through a cross-attention mechanism in the second stage.

RoVaR's algorithmic component, namely multi-lateration for UWB and ORB-SLAM3 for camera inputs, capture the physical and environmental dependencies of the localization problem while providing the absolute and relative position estimates, respectively. This, along with other sensor and environment-specific artifacts (RF channel quality, camera inter-frame keypoints, etc) together form the complete set of features that are fed into its ML component, which first employs a CNN encoder to extract dependencies within the individual sensor features to capture the certainty of their respective location estimates. This is followed by a cross-attention mechanism along with a 2-layer LSTM network (to capture temporal dependencies), to effectively fuse these disparate (absolute and relative) location-related features and predict the final absolute location estimate of the device with a high accuracy even under challenging environmental conditions. Estimating a device's *absolute* location estimate allows RoVaR to easily scale to multi-agent collaborative applications, where all agents are tracked within a common/global frame of reference.

Leveraging the initial location estimates from algorithms as features frees the ML module from the burden of capturing the physical and geometrical embeddings inherent to localization, allowing it to focus solely on sensor fusion. This results in two key benefits: (i) robustness and generalizability: delivering accurate tracking even in unseen challenging environmental conditions, and (ii) real-time tracking: significantly reduced end-to-end computations making the entire pipeline lightweight, enabling it to run in real-time even on resource-constrained mobile platforms.

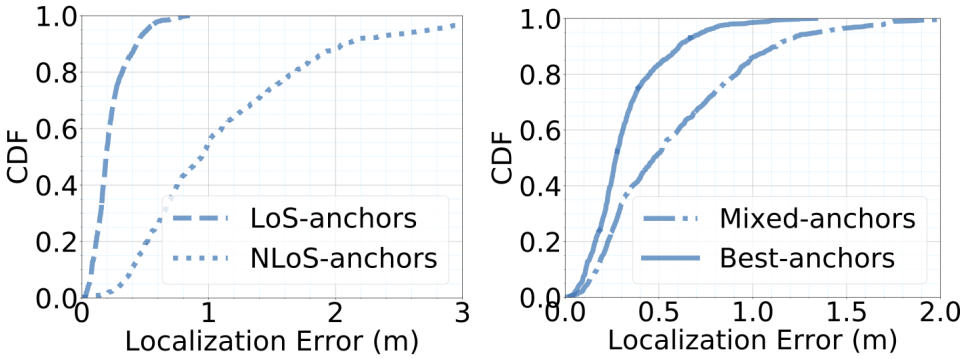
RoVaR Instantiation: A key contribution of RoVaR is the abstraction of dual layer diversity in both sensing modality and fusion methodology. In case of single agent tracking, it is straightforward to start the tracking by leveraging both modalities from the beginning. However, in case of multiple agents, RoVaR first relies on RF absolute coordinate space to quickly localize agents relative to each other in a global frame of reference, thus avoiding the need for all the agents to be co-located for common frame of reference (a synchronization technique as used by many existing AR multi-player gaming solutions such as ARCore [31] and ARKit [9]). The fusion model can then simultaneously leverage the output of passive tracking information to improve RF's absolute position. While RoVaR uses UWB for RF modality, similar infrastructure assisted active tracking solutions e.g., WiFi and mmWave solutions can be used in conjunction with the visual modalities to bring the global frame of reference advantages. To improve the active tracking accuracy further, many of these modalities can be used together when available. As we describe in §7, the ranges and link specific information can be collected from WiFi

and mmWave for active tracking and passed to our fusion model similar to our UWB ranges and certainty features. The model can be scaled to many such tracking modalities because of its simple concatenation of features and position tracking from different algorithmic solutions.

4.2 Algorithm-driven Feature Engineering

The goal of this stage is to generate features that capture not only location estimations of the individual sensor algorithms, but also the sensor/environmental artifacts that determine the certainty of these estimations.

4.2.1 RF Module: UWB sensors use time-of-flight measurement to determine the range R_i between the device and an anchor i. Range estimations to at least 3 anchors (or fewer if AoA info. is available) along with the anchors' location information is required for the multi-lateration algorithms to solve for the absolute location of the device. Given a set of n anchors at fixed position (x_i, y_i) with $i = 1..n$, the absolute 2D location of the device (x, y) can be estimated by minimizing the error $f_i = R_i - \sqrt{(x_i - x)^2 + (y_i - y)^2}$ across the anchors. Among the many different optimizations that are available [16, 24, 26], to solve the multi-lateration problem, RoVaR adopts the least-squares based approach.

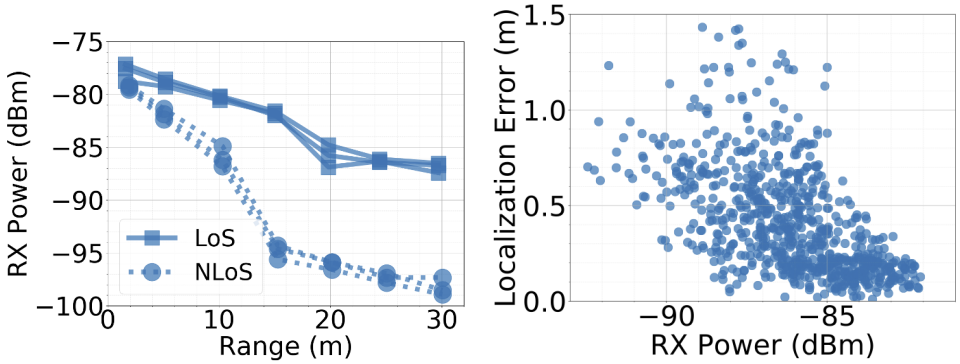


(a) Error in LoS vs. NLoS cases. (b) Best case vs. practical error

Fig. 4. LOS and NLOS Localization errors for UWB.

(1) Extracting features to capture NLoS impact: Two critical components that impact the localization accuracy are: (i) multi-lateration optimization error, and (ii) environmental conditions (multipath, NLoS) that manifest as inaccurate range estimates (see Fig. 2a). While the former can be estimated as a direct output of the optimization solution, the latter is challenging to address, and requires us to understand the impact of RF propagation and channel characteristics on UWB ranging error.

Impact of NLoS anchors on ranging accuracy: We study the impact of NLoS anchors on ranging and localization accuracy with five anchors (from our testbed described in §5), as the device to be located moves in a given trajectory, exposing both LoS and NLoS paths to various anchors. Fig. 4 captures the impact when all anchors are in LoS Vs. when all anchors are in NLoS. A low median error of 0.2m in the case of LoS is amplified to 1m (max. error of 3m) when all anchors are in NLoS. In a more practical setting, when the 5 anchors consist of a *Mix* of both LoS and NLoS, this error is 0.55m. However, selecting the *Best* set of 3 (out of 5) anchors (potentially LoS anchors) would reduce the median error to only 0.25m with the 90th percentile gain of 0.5m over the mixed anchor scheme. While these suggest that filtering out NLoS anchors can reduce the error significantly, selecting the best anchors is nontrivial without knowledge of the ground truth environmental conditions, a challenge we address next.



(a) Range vs. received power.

(b) Received power vs. Error

Fig. 5. Received power vs. localization error.

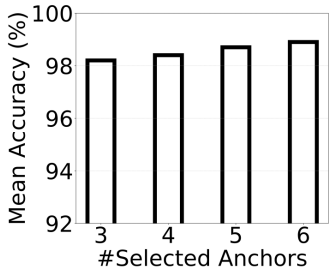


Fig. 6. Accuracy with different no. of anchors.

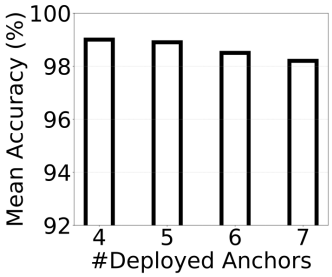


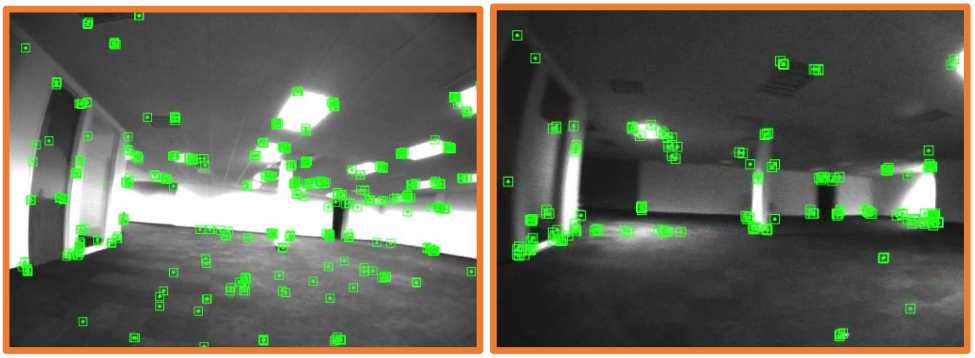
Fig. 7. Accuracy with 3 best anchors.

Selection of features that affect ranging: We investigate several link related metrics (e.g., first path power, amplitude, channel impulse power, etc) and find that amongst all metrics, received signal power (estimated as shown in equation below) exhibits a highly discriminative behavior with respect to range estimates.

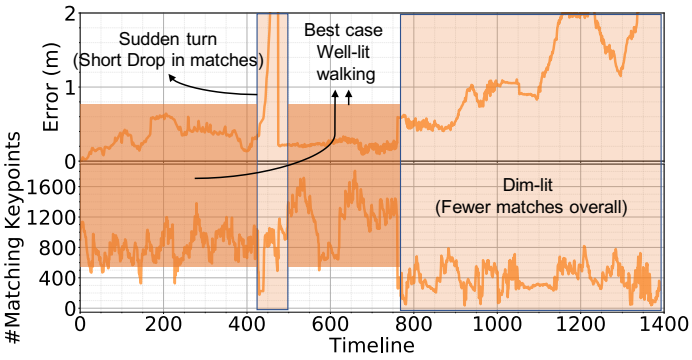
$$P_{rx} = 10 \times \log_{10}\left(\frac{C \times 2^{17}}{N^2}\right) - A \; dBm \tag{1}$$

where, C is the Channel Impulse Response Power, N is a preamble accumulation count used to normalize the amplitude of channel impulse responses [21], A is a constant determined with the pulse repetition frequency (details found in [1]). Fig. 5a shows the impact of received power on ranging under LoS and NLoS conditions for three different anchors, averaged over several device locations. As the range from the anchor increases, the received power under NLOS goes below -95dBm even within 15m, while it remains above -90dBm even after 30m for LoS. Consequently we find that that the received power has a strong negative correlation with localization error (Fig. 5b). Thus, received power to an anchor (P_i) along with its range (R_i) can serve as an effective discriminative feature, capturing the impact of NLoS on the accuracy of its range and eventually location estimate, although with a nonlinear relationship.

(2) Leveraging RF features to address NLoS: The extracted feature $\{P, R\}$, serves two purposes in RoVaR: (i) to filter out accurate range estimates used for localization when > 3 anchors are available, and (ii) indirectly



(a) Well-lit (568 matches) (b) Dim-lit (252 matches)



(c) Matches vs. Error

Fig. 8. VO matching keypoints (markers in green) vs. tracking error under different lighting conditions.

capture the certainty (variance) of the location estimate for subsequent fusion. Using these features, we next explain how filter out poor anchors to avoid incorrect ranges for localization.

Anchor selection for location estimate: In order to select a best subset of anchors among all available anchors, we use simple ML models (e.g., SVM). This design choice keeps anchor selection light weight, and avoids burden on the fusion model. We train a classifier that selects the best K anchors for localization by fitting a model with the ranges from all anchors along with their corresponding received powers as input, and the best anchor set (providing min. error compared to ground-truth) as the output binary vector. We use a multi-output classification using classifier chains [44] to exploit the correlation among the anchors rather than independently selecting each anchor. We fit three models— SVM, Logistic Regression, and a RandomForests classifier and perform a grid search on each of them to tune the best parameters. After the grid search, we select a best performing model (SVM in our case) and its optimal parameters from the search. Figure 6 and 7 shows the prediction performance in terms of mean accuracy, when predicting a subset of anchors. As shown, the model performance is highly accurate with the accuracy increasing as expected when more anchors are available for selection. Note that the model only helps filter the inputs ($K = 3$ in RoVaR), while the multi-lateration algorithm is still responsible for estimating the location estimate.

(3) Feature composition for fusion model: While anchor selection filters NLoS anchors to improve the location estimate, a device often might not have access to 3 LoS anchors. As seen in Fig. 4, the maximum localization error is still over 1m even after selecting the best set of anchors. Thus, feeding the absolute location estimates alone can misguide the fusion framework to adapt inefficiently to different RF conditions. Hence, RoVaR leverages the additional feature of received powers together with the ranges as a form of certainty measure (in modeling anchors as LoS and NLoS) and combines these certainty features with absolute location estimates from the multi-lateration algorithm as a composed RF input to our fusion framework. Formally, the input from the RF path to RoVaR's module is given as $F_r = < X_u \oplus P_i | R_i >, < Y_u \oplus P_i | R_i >, i \in [1, K]$, where $\oplus$ is the concatenation operation of location estimates with the device's range and received power to each anchor.

4.2.2 VO Module: Most visual tracking solutions use a stream of stereo frames to incrementally (relatively) localize the camera/device on a frame-by-frame basis, by extracting unique features from each frame. Typical features can be SIFT, SURF, ORBs [45], etc. RoVaR (ORBSLAM3) employs ORB features that are invariant to scale, rotation, and translation properties. At a given time instance (t), these features are extracted from two or more camera frames (V_t^s), reprojected onto the real world to estimate the depth of each feature, and the scale of tracking. These matched features are used to find correspondences with a previous reference frame(s) (V_{t-1}^s) and create a set of matching keypoints which are then used to compute relative displacement estimates ($\Delta x_v, \Delta y_v$) to get the current position (x_v, y_v).

(1) Avoiding drift by tracking translation & heading: Being a relative tracking approach, even temporary environmental artifacts (limited visual features, dynamic scenes, etc.) that degrade just a few displacement estimates, result in the continuous accumulation of errors over time. Hence, naively combining VO's relative position estimate directly with UWB's absolute location estimate, can lead to long-term drift problems in the final fused location estimate. RoVaR addresses this challenge by employing relative displacement in translation (r_v) and the heading (θ_v) directly rather than using the final relative estimates of position, to drive the fusion module. Hence, even when odometry faces displacement errors temporarily, the resulting error propagation is only in displacement (heading continues tracking the absolute trajectory direction), which is transient and does not propagate. Further, even this transient error propagation is completely eliminated, when its relative estimates (r_v, θ_v) are fused together with UWB's absolute location estimates (x_u, y_u), and result in accurate tracking.

(2) Composing features to capture estimate certainty: Note that the relative estimates can themselves be erroneous even in the absence of any error drift. Short-term environmental artifacts (e.g. occlusion lighting) can result in inaccurate estimates with up to 1m error (see Fig. 13) even in best visually feature-rich environments. To compensate for such inaccurate estimates, RoVaR employs additional features to capture the certainty of the tracking algorithm's estimate .

Recall that, the features extracted from the images determine the VO's tracking accuracy and robustness. We dissect ORB-SLAM3's tracking component to understand its features in capturing certainty of its estimates. A straightforward feature is the number of ORB features that are matched by the algorithm across all stereo frames. However, these cannot be directly used, as there can be spurious matches and outliers. The latter can be determined based on their estimated depth and removed before feeding them to the tracking algorithm. After filtering the outliers, the final matching keypoints are the ones most relevant for tracking. We study the effect of matching keypoints on tracking performance under different environments and find that the tracking error is stongly influenced by the keypoints. Figure 8 shows the number of matching keypoints under well-lit (568) and dim-lit (252) conditions, whose corresponding tracking error is shown in Figure 8c. Note that if the keypoints go below 100 for longer periods, the tracking completely fails. We also explored other features such as depth of matching features, outliers and inliers, reprojection error, etc, but we find matching keypoints (M) to best capture the certainty of the tracking estimates. Hence, RoVaR combines this certainty feature M with the relative

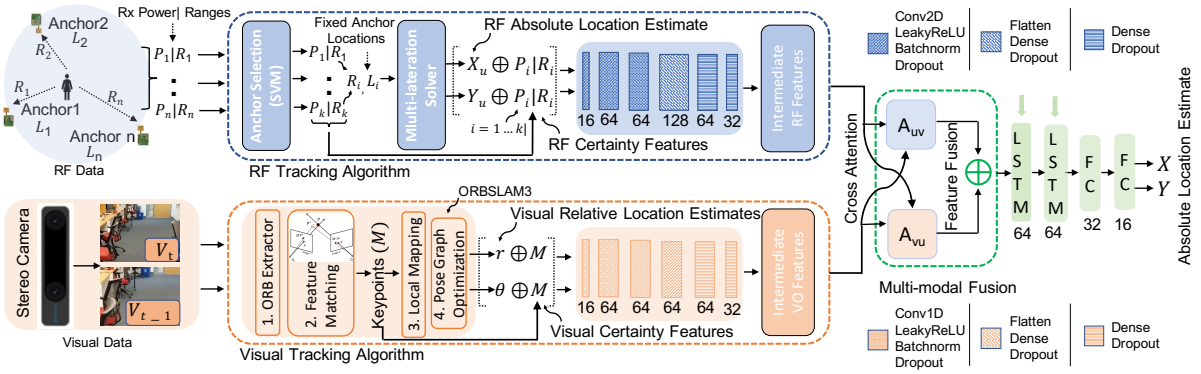


Fig. 9. RoVaR Hybrid Fusion. RoVaR embraces the benefits of classicial solutions at the first layer, and employs an ML pipeline for multi-modal sensor fusion to obtain absolute location estimate.

tracking estimates (r, θ) as a composed VO input to our fusion framework. Formally, the input from the VO path to RoVaR's module is given as $F_v = < r \oplus M >, < \theta \oplus M >$, where $\oplus$ is the concatenation operation.

4.3 Sensor Fusion through Cross-Attention

Fig 9 shows the end-to-end architecture of RoVaR's hybrid fusion. RoVaR first prepares the composed features from individual sensors by passing them through a simple CNN network. This allows the location estimate and its certainty-related features to be embedded into a more representative feature that can enable effective fusion.

RoVaR then adopts an *Attention* mechanism for fusion, a commonly used learning technique [54, 58], for adaptively weighting the features of the sensors, so as to leverage their complementary nature. While self attention [32] weights the features of an individual sensor to self-adapt and eliminate the influence of outliers, cross-attention focuses on weighting each sensor with respect to one another to extract inter-sensor correlations and leverage their complementary nature. Intuitively, the model would weight the RF estimates (with higher certainty) more when VO encounters unfavorable environments (e.g., dim-light; lower certainty of VO), and the VO estimates more when RF estimates suffer from NLoS anchors. RoVaR directly adopts cross-attention since the features engineered by its algorithms already capture self-attention (incorporating features that correlate with tracking error). The two cross-attention masks (A_{rv}, A_{vr}) are *jointly* learned using the RF and VO features (F_r, F_v) respectively.

$$A_{rv} = Sigmoid((W'_{rv}F_v)^T W''_{rv}F_v), A_{vr} = Sigmoid((W'_{vr}F_r)^T W''_{vr}F_r) \quad (2)$$

where, W' and W'' are the weights learned during the training, which transform the extracted features into a lower dimension version of original RF and VO inputs, that extracts the underlying global topology of data. On a high level, the equation captures meaningful features through local convolutions and long term dependencies through embedding spaces, jointly adapting the masks by capturing cross-correlations between the two sensors. After the masks are learned, each mask is applied to its respective sensor feature (element-wise, $\odot$) and then merged (concatenated, $\oplus$) together to provide the fused feature $(A = [A_{rv} \odot F_v] \oplus [A_{vr} \odot F_r])$. Finally, the output from cross-attention, which is a single dimension flattened array, is forwarded to an LSTM network - a 2 layer RNN (with 64 hidden units per layer) to model the temporal dependency of the fused features, followed by 2 Fully Connected (FC) layers that finally outputs the predicted absolute location. The LSTM's ability to access its outputs from prior time instants (i.e., prior absolute location estimates), enables it to effectively predict *absolute* location estimate while relying on the *relative* location estimate from VO (when RF features have large uncertainty), thereby eliminating (resetting) potential VO drifts.

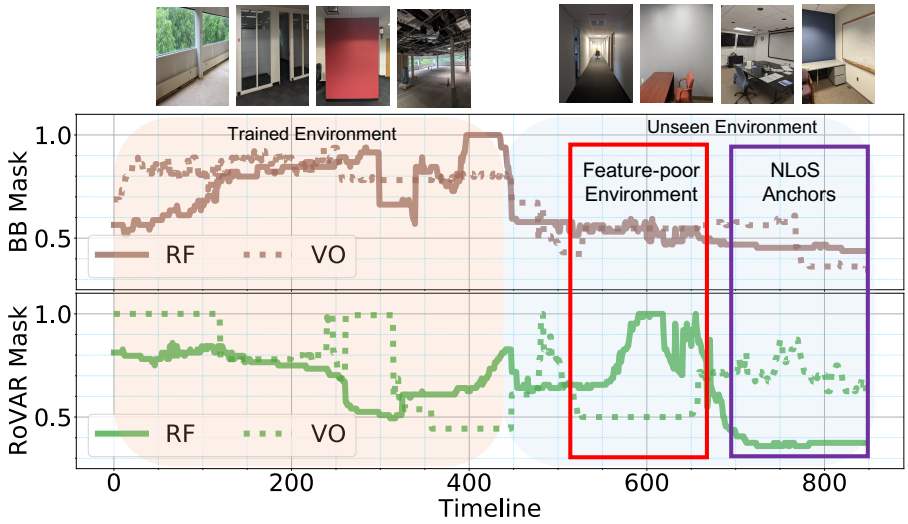


Fig. 10. Attention mask for BB vs. RoVaR. BB's attention is comparable to RoVaR in trained environemnt (orange region) while failing in unseen places (light blue region).

Effectiveness of Cross-Attention: Fig. 10 shows RoVaR's attended weights for RF and VO feature vectors. As the device suffers dim-light conditions, RoVaR increases RF feature attention from .6 to 1, while decreasing that of the VO's features from .9 to .5, and in NLoS, RF features are weighted low (decreased from 1 to 0.1), while VO's weights increase from 0.5 to 0.85. Given the independent and complementary nature of the RF and VO sensors as well as their environmental artifacts, it is clear that RoVaR's cross-attention plays a valuable role in effective sensor fusion.

4.4 RoVaR vs. Pure Deep Learning Fusion

At this point, it is natural to wonder - *"How and why is RoVaR better than prior machine learning based fusion solutions?"* Prior works (e.g., milliEgo [35], DeepTIO [47], VINET [17], etc.) have employed raw sensor data to directly train a (BlackBox – BB) ML model, compared to RoVaR's hybrid approach, where ML is largely leveraged for fusion alone. These blackbox models tend to be more complex (e.g., additional attention layers), requiring higher computational resources (e.g., more convolution and RNN layers). Further still, having to extract the challenging problem structure inherent to *absolute localization*, they are unable to generalize and deliver robust performance in unseen environments (see §6).

For an objective comparison, we also design a pure ML-driven counterpart solution to RoVaR (called BlackBox, BB), by instrumenting the network architecture similar to those used in [35] for our UWB and stereo-camera inputs, as shown in Fig. 11. The key differences in BB are that, it has a deep feature extraction network for both UWB and camera streams, and employs a self-attention module. The role of self-attention is to enable a form of filtering of the weak input features, while emphasizing those that contribute to an accurate output prediction. After the features are self-attended, it follows a cross-attention and fusion pipeline similar to RoVaR.

While RoVaR's detailed performance comparison with BB is deferred to §6, we highlight two of its key advantages: i) much of its feature extraction burden is driven by efficient algorithms, allowing its ML-component to focus solely on fusion, enabling it to generalize and accurately track absolute location even in unseen environments (see Fig. 10), and ii) lack of a complex feature extraction network reduces its model complexity significantly, making it light-weight and deployable on resource-constrained platforms providing real-time operation. In our testbed (described in §5), we show that RoVaR has only 1.2 M parameters while BB requires 70 M, and when

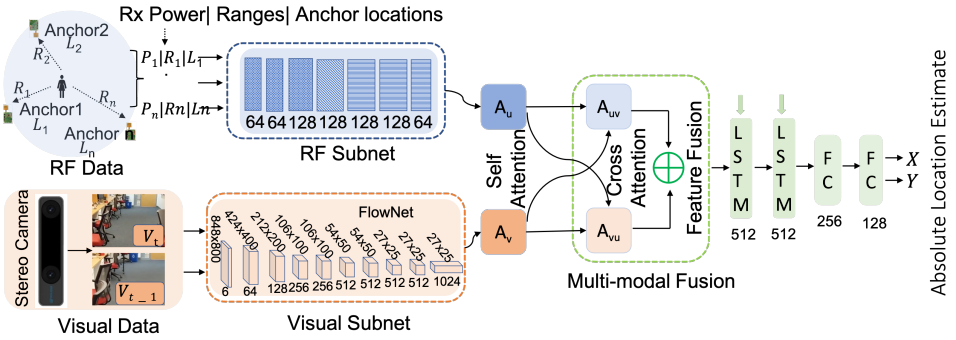


Fig. 11. Blackbox fusion model (BB). Unlike RoVaR, BB directly takes raw inputs to extract features through DNN for fusion.

translated into raw data (after model compression), RoVaR's memory footprint is 62× smaller than BB, directly leading to a 5 fold decrease in inference latency (see §6.7).

5 IMPLEMENTATION

RoVaR system components: We implement a hand held prototype of RoVaR, containing: (a) Decawave EVK1000 [2] UWB radios for RF localization (§4.2.1), (b) Intel T265 stereo camera [3] for VO tracking (§4.2.2) and (c) A ThingMagic M6E-Nano RFID reader [4] for Ground Truth (GT) measurements. The RFID reader and the UWB radio are connected to a RaspberryPi-4 (RPi) computer via the RPi's serial and Ethernet ports, respectively. Fig. 12 shows the RoVaR prototype. As seen in the picture, the camera, UWB and the RFID reader are all on the same vertical plane, ensuring that the location determined by each sensor are on the same X-Y plane. A laptop (or a Jetson Nano for real-time implementation) is connected to the Intel camera via an ethernet cable for recording the camera images. The raw measurements are recorded at 30Hz. At each timestamp, three readings are recorded: 1) UWB ranges on RPi, 2) RFID ground-truth on the same RPi, 3) camera images on the Linux laptop. Both RPi and Laptop are synchronized using an NTP server within a local LAN. A locally hosted NTP server has a time synchronization accuracy of under 1ms (in our experiments we observe 200us synchronization accuracy). Therefore, the sensors have more than required scale of synchronization because their sampling rate is much lower (≈33ms) than the NTP accuracy (<1ms). Note that the UWB tag has a periodicity of 10Hz i.e., it records one reading at every 100ms. Therefore, we record the same UWB reading for every 90ms because of low frequency. Finally, RFID reader has a 50Hz sampling rate which is higher than both camera and UWB sampling rates.

Testbeds: We build several testbeds across multiple floors of an office and a home with varying light and scenary. They include conference rooms with artificial lights, semi-constructed cemented-area, glass walled rooms, and office corridors surrounded by obstacles (cubicles or offices) (see Fig 12). Lighting conditions in all but one, can be controlled using light-dimmers, while one testbed has only natural sunlight. We use a subset of testbeds (trained environments) for data collection, while the rest are kept completely unseen for testing the generalizability of RoVaR.

Data collection: Each testbed has 2-to-3 different trajectories that contain simple straight lines to more complex curvy-trajectories. For each trajectory, we collect data in: (i) bright and dim lighting conditions, and (ii) when user is walking and running. The UWB anchors are often occluded due to physical structures and hence the UWB data contain both LoS and NLoS ranges. We have collected data from 72 different traces, totalling a distance of 4.8K meters and 232,000 data points for training the RoVaR models.

Ground Truth (GT): We use RFID tags, placed continuously every 10cm along a user's walking path for GT

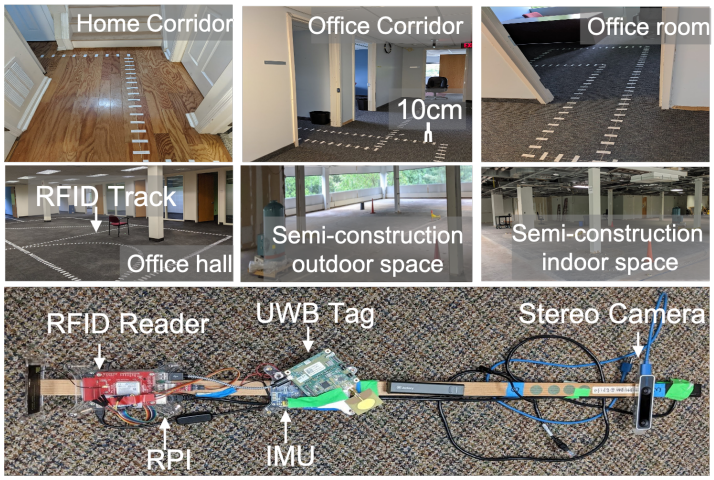


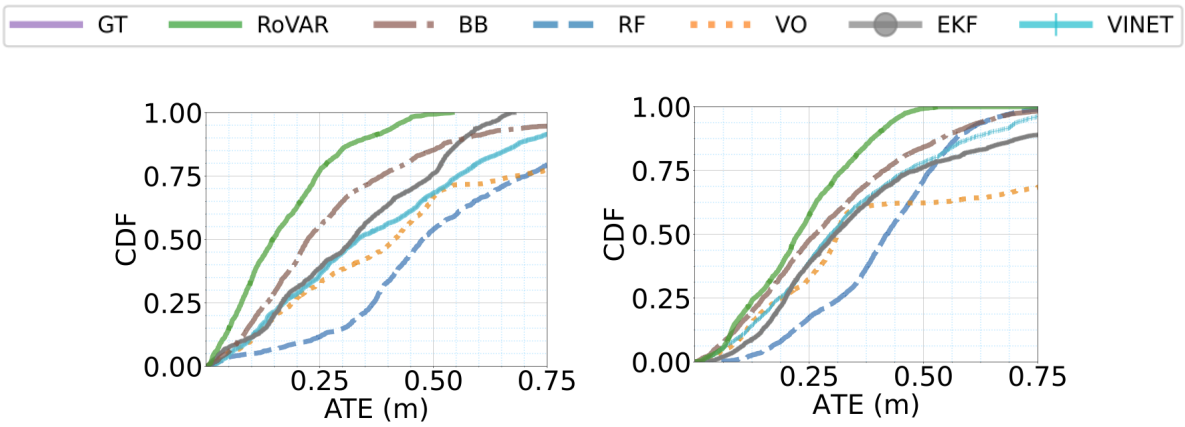
Fig. 12. RFID (GT) tracks and RoVaR prototype

data. While other modalities like Lidars are conducive for single-user GT, the physical RFIDs provide absolute GT enabling multi-agent tracking evaluations. We manually record each RFID's X-Y location, and as the user walks along the predefined trajectories holding the RoVaR prototype in the vertical position (see Fig. 12), the RPi which is connected to the RFID reader (with the antenna at the bottom of the stick) records a time-stamped EPC and RSSI information (at 50Hz) of the RFID directly below it. The Tx power of the RFID reader is set to a minimum so that the RFID antenna reads the RFID tag only when it is a couple of inches directly above it. Despite such power control, we found that the RFID reader read more than one tag i.e., nearby tags occasionally if the reader is exactly in the center of two tags. To address this problem, we collect additional RSSI information from the tags along with the tag position and select a tag that has highest RSSI value. Selecting a tag with higher RSSI ensures a most accurate or nearest tag, and therefore the most accurate ground-truth location as well. Further, the velocity of the device, derived from a short sliding window of tag reads (say 20 tags), allows for accurate interpolation of location within the 10cm granularity.

UWB localization: We use 7 static anchors placed at known locations in each testbed. The RPi connected to the UWB radio on the RoVaR prototype ranges sequentially with each UWB anchor. RoVaR implements an optimized ranging protocol [2], to reduce the total packet exchanges with each anchor. Range estimates are recorded with timestamps at 15Hz speed. RoVaR implements the geometry constrained least-squares multi-lateration algorithm [24] for localization.

ORB-SLAM3: We use Intel T265 camera [3] for VO. The T265 has stereo cameras with fisheye lens and records images at 848x800 pixel resolution. The recorded camera images is time-synced with the UWB and the GT data. We use ORB-SLAM3 ROS implementation [14] for tracking.

Network training: We use 70% of our dataset for model-training, 10% for validation and 20% for evaluation. We ensure that our dataset is diverse to avoid the case of over-fitting. We use Pytorch [41], an open source machine-learning (ML) library to implement the RoVaR models. Before feeding the data to the network, we normalize the input data by subtracting the mean over the dataset. We use Adam optimizer with L2 norm as the loss function during the training. RoVaR's model is lightweight and takes less than an hour for training even on a desktop-grade Nvidia-1070 GPU.



(a) Favorable (b) Practical

Fig. 13. RoVaR Vs. Baseline performance in Trained environments.

6 EVALUATION

6.1 Experimental Methodology

We extensively evaluate RoVaR for both single and (collaborative) multi-user tracking scenarios under diverse (trained and unseen) environmental conditions for two different human-motions (running and walking) to demonstrate its robustness and efficacy vis-a-vis other state-of-the-art solutions (described below). Our evaluation testbeds in home and office environments have varying texture (rich and poor), lighting conditions (good and dim) and healthy mix of LoS and NLoS UWB anchors (for RF) to mimic real world deployments. We use the Absolute Trajectory Error (ATE) [51], calculated as the Euclidean distance between corresponding points on the estimated and the GT trajectories, to quantify the performance of RoVaR.

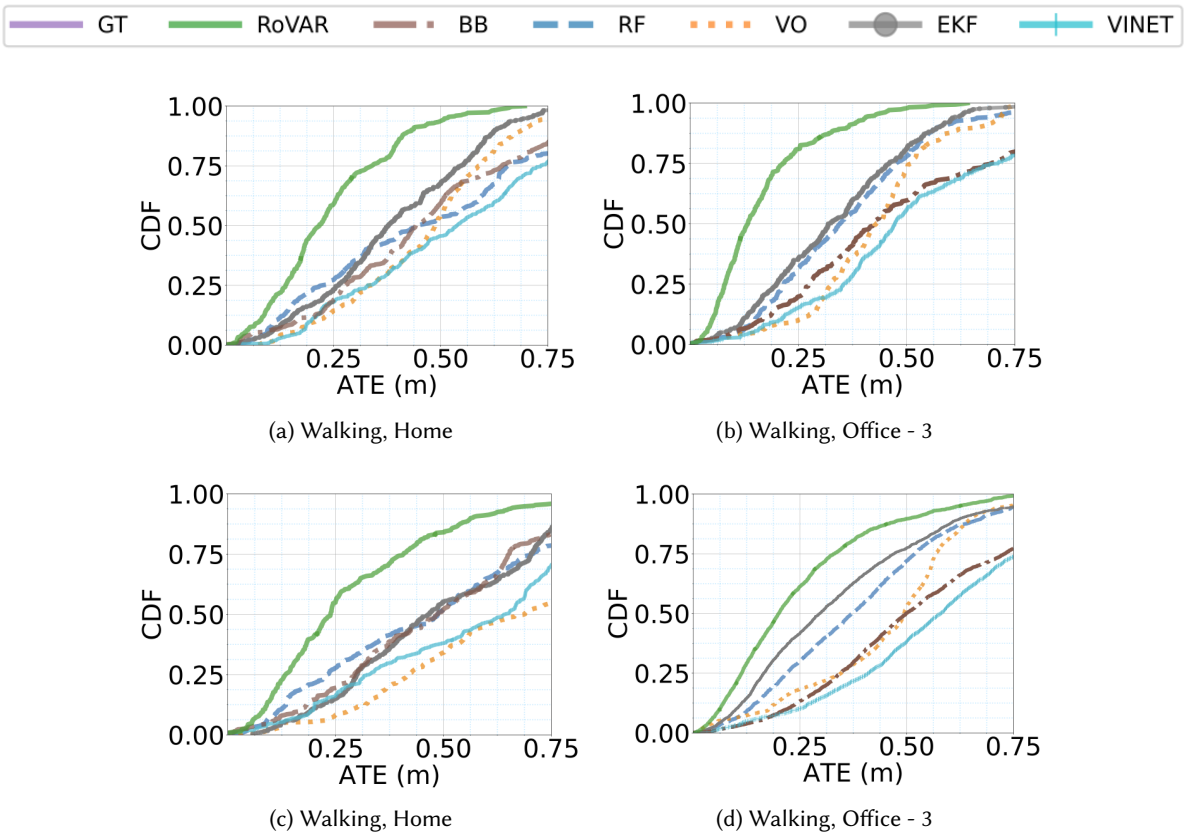
6.1.1 Baselines: We compare RoVaR with a suite of baseline solutions that include both algorithmic and ML based fusion: 1) **RF** is the multilateration-based RF localization solution described in §2.1, 2) **VO** is the ORB-SLAM3 VO tracking solution described in §2.1, 3) **EKF** is the Extended Kalman-filter, a commonly used algorithmic and industry-grade practical solution (e.g., ARCore [31] and ARKit [9]). We tune the EKF parameters to ensure its optimal performance[4], 4) **BB** is the Black Box ML based fusion described in §4.3. 5) **VINET** is a state-of-the-art ML based VIO tracking solution [17], that uses both visual and IMU data for relative tracking. We use the same model as [17] without having to retrain the model by collecting IMU data in our environment. Fig. 12 shows that the IMU sensor is placed alongside the UWB tag and the IMU readings are collected on the same RPi of RFID reader so that the IMU data is also in sync with the RFID and camera data. We collect IMU data at a rate of 100Hz. Note that **BB** and **EKF** are the state-of-the-art active+passive fusion solutions while **VINET** is only state-of-the-art VIO relative tracking solution.

6.2 Performance in Trained Environments

We begin our evaluations in environments used for training the ML models.

6.2.1 Favorable Conditions: We first evaluate in environments rich in texture, good lighting (good for VO) and LoS RF anchors (good for RF tracking). Fig. 15 shows the Ground Truth (GT) and the estimated trajectories of

[4]Tuning the EKF to find the optimal *process* and *measurement* noise covariance matrices is nontrivial in practice. We follow a general approach by iterating the filter until all the estimates until all the parameters converge.



(a) Walking, Home

(b) Walking, Office - 3

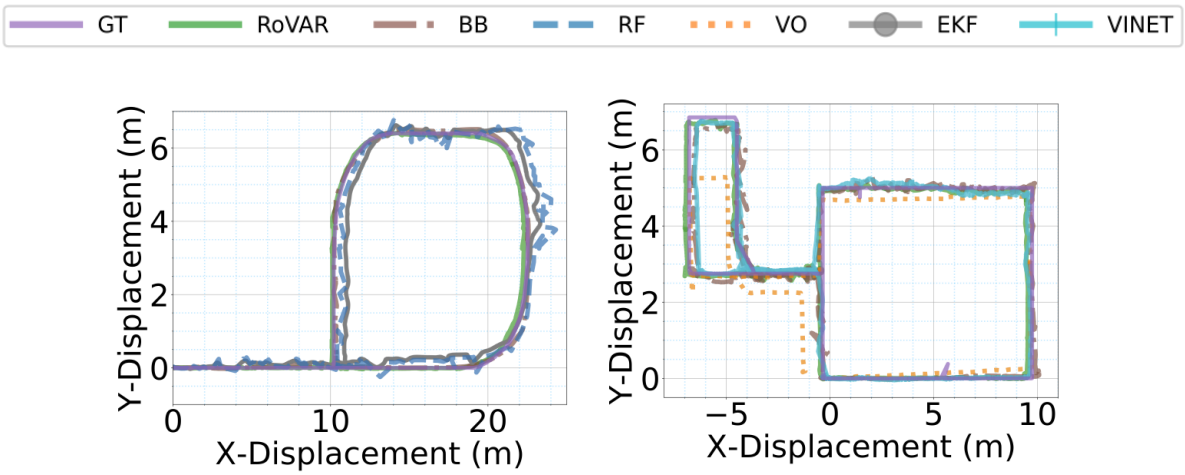
(c) Walking, Home

(d) Walking, Office - 3

Fig. 14. RoVaR Vs. Baseline performance in Unseen environments. Favorable conditions(a,b), Practical conditions(c,d).

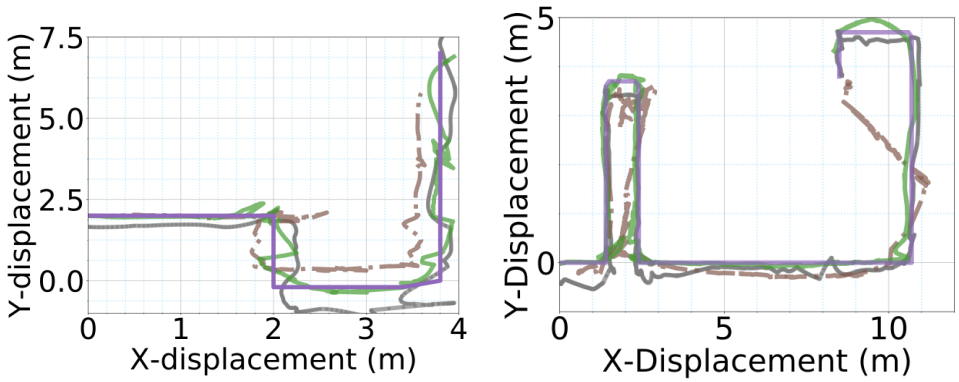
all the solutions, when a user is walking and running. The CDFs in Fig.13a show the ATE (Absolute Trajectory Error) information for each solution. RoVaR has significantly better performance (median error 17cm) over all the alternatives with 35% and 64% better median-accuracy even compared to BB and VINET, respectively. While the algorithmic solutions (RF and VO) are agnostic to environmental conditions, the trained environments allow us to capture BB and VINET's performance in their most favorable scenarios. As expected, BB and VINET perform better than the individual algorithms, RF, VO, as well as their fused version, EKF. This is because VO, despite its inherent error-correction mechanism, suffers during trajectory-turns where the number of ORB feature matches are relatively lower than when walking in straight lines, leading to accumulation of error; while RF delivers a coarser accuracy compared to visual tracking even in the best case. The EKF fusion aims to minimize individual algorithmic errors, but is still vulnerable to the individual sensor errors and artifacts, while ML solution are able to mitigate algorithmic uncertainties using data-driven models.

6.2.2 Practical Conditions: Next, we evaluate RoVaR in the same environments, but by introducing dim lighting conditions (in a room that is part of the trajectory - Fig. 15), and moving a subset of RF anchors into NLoS. Results are shown in Fig. 13b. We see that RF and VO algorithms suffer due to NLoS anchors and dim lighting, respectively, with VO (median error 40cm) performing worst. VINET which relies mainly on the VO performance seem to suffer too due to VO's degradation. BB performs better (median error 25cm) than the algorithmic ones (EKF - median error 35cm), owing to its model (which leverages RF and visual raw data) capturing environmental



(a) Walking, Office - 1 (b) Running, Office - 2

Fig. 15. RoVAR Vs. Baselines trajectory in trained environment – Favorable condition (left), Practical condition (right). The trajectory on the right shows two rectangular regions, where the smaller rectangular region is a separate room space with dim lighting condition failing most of the existing algorithms because of lack of good features in the dim light.



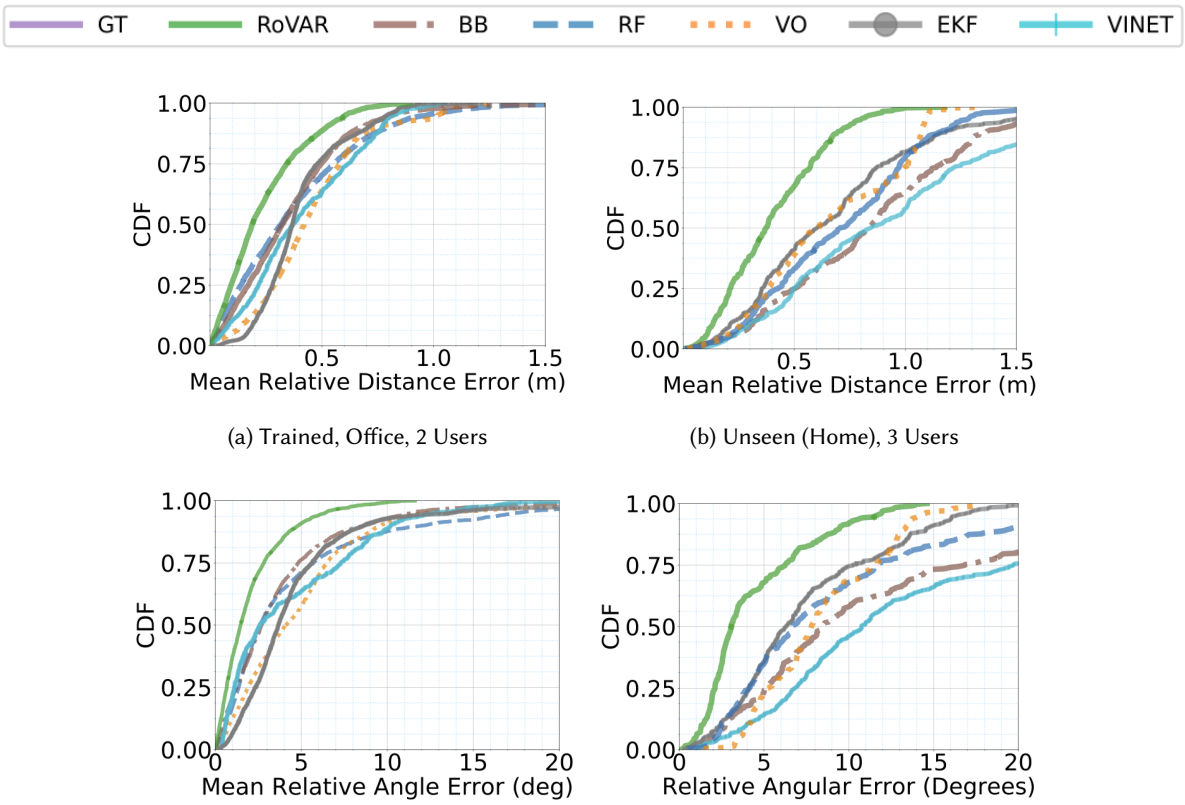
(a) Walking, Home (b) Running, Office - 3

Fig. 16. RoVAR Vs. Baselines trajectory in Unseen environment in Practical conditions. The trajectory on the right (one of the most complex trajectories in our dataset) shows a narrow hallway in the beginning of the trace, where most of the existing solutions fail because of both plain feature less scene with white walls and severely occluded NLoS scenario for UWB ranging.

artifacts and their impact on fusion in the trained scenarios. Finally, RoVAR due to its RF anchor selection module increases the accuracy in RF tracking, and combined with its dual-layer diversity in fusion performs best with median ATE 17cm, a gain of 47% and 64% in accuracy over BB and VINET, respectively.

6.3 Robustness in Unseen Environments

In order to test RoVAR's ability to generalize in untrained (unseen) environments, we evaluate its performance in completely different input distributions (new places) that are kept aside while training. We deploy two separate testbeds - one in a separate office environment and another in a home environment - for this purpose. As an



(a) Trained, Office, 2 Users

(b) Unseen (Home), 3 Users

(c) Trained, Office, 2 Users

(d) Unseen (Home), 3 Users

Fig. 17. RoVaR performance for multi-users. Relative distance error (a,b) and Relative angular error (c,d)

example of an unseen home environment is shown in Fig. 12 on the top left corner of the image with "home corridor" headline. As shown, the home environment has wooden floor which is never seen in any of the trained office environments. Note that the home environment scenes are not included as part of the training data, so it is completely new to the model. More examples of the trained vs. unseen environments are also shown in Fig. 10 to demonstrate the effectiveness of our cross-attention technique, where trained images are shown on the left and unseen images are shown on the right. The corresponding trajectories with tracking performance are visualized in Fig. 16 demonstrating large trajectory deviations.

6.3.1 Favorable Conditions: As before, we begin evaluating with favorable conditions, i.e., ensure visual scenery has rich texture, good lighting and the RF anchors are all in LoS. Fig. 14a-b shows that while RF, VO and EKF algorithms' performance are similar to the trained environments due to their environmental-agnostic nature, BB and VINET solutions' performance deteriorates significantly (>2× degradation compared to trained environments), with the top 10% of points having errors >1m. Clearly, BB and VINET are unable to generalize and perform robustly in untrained environments. On the other hand, RoVaR has a median error of only ≈20cm, 15cm in Home and Office - 4 (similar to its performance in trained environments) and a maximum error that is < 0.75cm (≈

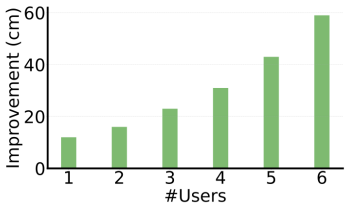


Fig. 18. RoVaR's relative tracking improvement over BB across six users.

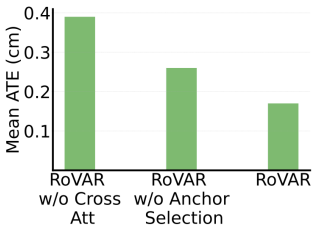


Fig. 19. Breakdown of RoVaR's individual components.

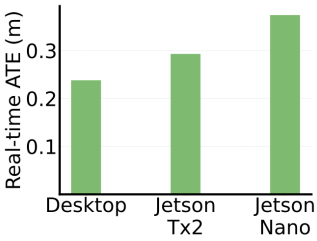


Fig. 20. RoVaR's real-time accuracy.

230% increase in median-accuracy). *RoVaR's strategy of leveraging RF and VO algorithms for extracting tracking features, allows its sensor fusion to perform robustly even in untrained environments with high accuracy.*

6.3.2 Practical Conditions: Untrained environments with practical conditions are perhaps the most challenging. To highlight RoVaR's robustness, Fig. 16 compares two trajectories: one favorable (left) and another practical (right). We find that RoVaR shows a consistent behavior in diverse scenarios (trained/unseen as well as favorable/practical conditions), while the alternatives suffer in one or the other scenarios. The corresponding error CDFs for unseen case are shown in Fig. 14c-d. All of the alternatives suffer in this case because of the noise from practical conditions for EKF (RF+VO), and limited generalizability of BB/VINET. However, RoVaR offers a median error of only 18cm (office-3) and 23cm (Home) even in these adverse cases, showing an evidence that *its hybrid fusion strategy gracefully delivers in diverse environments.*

6.4 Scalability across Multiple Agents

Keeping collaborating applications in mind, we evaluate RoVaR's ability to track multiple agents (users) with respect to each other. We have up to three users moving simultaneously along the same trajectory, but in different directions. To capture the tracking accuracy across multiple users simultaneously, we calculate the mean error in the relative positions for every pair of users. For every user-pair, this in turn, is captured jointly by the error in their relative distance and angle compared to ground truth. In the interest of space, we show the results (Fig. 17) for two environments (trained office and unseen home environment) in practical conditions. *RoVaR, with its ability to absolutely localize every user (with help of anchors), and leverage dual-layer diversity, performs best with 30cm, 35cm relative-distance, 2.5°, 3° relative-angular errors in trained and unseen environments, respectively. With gains increasing for more users, it can scale to collaborative applications easily even in unseen environments.* This is a 1.2× (distance) and 1.6× (angle) improvement over BB and VINET respectively. RF, VO and EKF's relative-distance error is better than BB and VINET owing to better robustness, but less than RoVaR by 36%, 41% and 35% respectively.

To show the effectiveness of RoVaR for multi-agent scenario, we also evaluate tracking performance across more users by emulating simultaneous trajectories of multiple users. To do this, we leverage trajectories from our test dataset and synchronize the timestamps of data from different trajectories to replay multiple users as if they were simultaneously following those trajectories. To introduce the diversity in terms of type of trajectory as well as the environmental conditions, we randomly select the users' trajectories from different categories. We evaluate this up to six users. For each multi-user scenario, we replay five different combinations of traces and report the average tracking performance across five traces. We observe that RoVaR achieves 44cm relative tracking accuracy for six users which translates to ≈2.3× improvement over the next best algorithm BB.

Additionally, we also evaluate a real world multi-user AR gaming scenario, where startup latency plays a key role. We experiment with two solutions: 1) ARCore cloud anchors [31], a state-of-the-art industry solution for

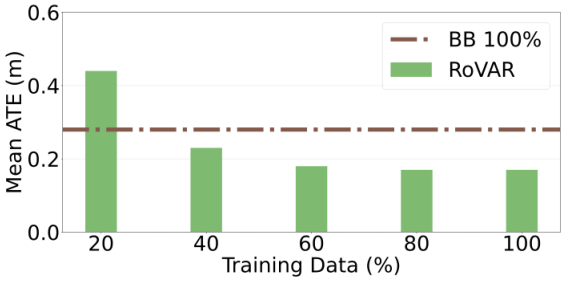


Fig. 21. Training data size Vs. Performance.

multiplayer AR games, 2) RoVaR. With ARCore, multiple users should agree on a common origin object and all of them should be co-located and view the object from the exact viewpoint. On the other hand, RoVaR gives the flexibility of global frame of reference and hence many users can be instantaneously localized relative to each other. We find that it takes more than 15 seconds on average to start a multiplayer AR game in case of ARCore while RoVaR allows the players to start the game instantly.

6.5 Ablation Study

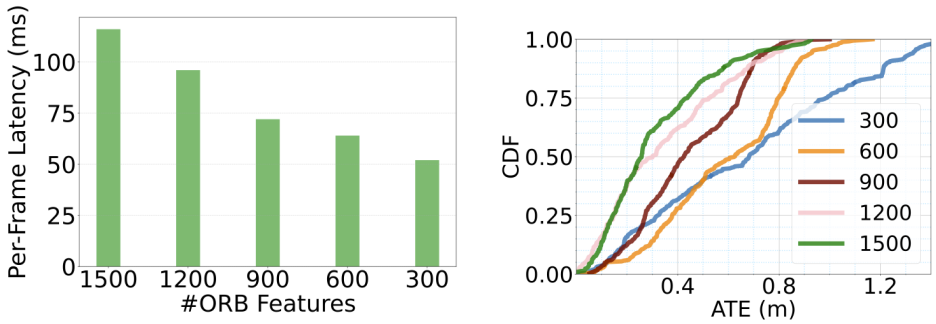
We show an ablation study to understand the performance benefits individual components in RoVaR. Fig. 19 shows the impact of RoVaR's individual components: 1) RoVaR without cross-attention mechanism (i.e., directly merging features to predict the final location), 2) RoVaR without anchor filtering. The figure shows that both components are critical to the performance of RoVaR. Similarly, RoVaR without anchor selection performs poorly because of the influence of NLoS anchors' inaccurate ranging is affecting the localization accuracy. In case of RoVaR without cross-attention, the system performs poorly because the features from *both* the sensors are merged to predict the location even when one of them is facing bad environmental conditions.

6.6 Low Training Requirements

In addition to bringing robustness to new environments, *the reduced burden on RoVaR's fusion module, enables it to operate at a significantly reduced training cost compared to pure ML-driven solutions.* Fig. 21 shows RoVaR's performance as an increasing function of training data size (out of entire dataset available to BB). RoVaR is able to reduce the training size requirements by over 60%, while still delivering a comparable performance.

6.7 Real-time System Performance and Practicality on Mobile Platforms

RoVaR's fusion model requires significantly less memory and processing power making it extremely lightweight compared to BB model. Table 3 highlights the effectiveness of RoVaR in multiple fronts— *latency, memory and power consumption* on mobile platforms (Jetson Nano/TX2). Moreover, the BB model is too memory intensive to run on these low-end devices unless we downsample it to 480p resolution from 848x800. Even then, the running time of BB model is 5X more (348ms) than RoVaR. More importantly, RoVaR has 3× less power consumption compared to BB model, which makes RoVaR a well suited ML-driven fusion methodology that can be deployed on power constrained mobile devices in practice. On *low-end platforms* like Jetson Nano, RoVaR's fusion model can easily run in real-time (9.5ms). However, the VO (ORBSLAM3) algorithm, the main overhead contributor to the RoVaR pipeline can afford to use 900 ORB features (for latency 75 ms – see Fig. 22a) instead of the optimal choice of 1500 ORB features, resulting in RoVaR's overall latency to be around 90ms (20ms more than TX2). Adopting VO optimizations such as EdgeSLAM [12] over edge-cloud networks can further reduce the latency. Nonetheless, the drop in overall median error (Fig. 22b) due to this setting is only 15cm resulting in RoVaR's



(a) Latency vs. ORB Features (b) Accuracy vs. ORB Features

Fig. 22. Latency and Tracking accuracy trade-off with ORB features in ORBSLAM3.

Table 3. Latency, Memory, and Energy usage (*video is downsampled to 480p for BB because it cannot run on low-end devices. The reported power consumption is measured on Jetson Nano platform using its i2c interface power rails (e.g., [10])).

	RF	VO	BB	RoVaR Fusion	RoVaR Overall
Desktop (ms)	1.80	43.67	126.91	2.61	48.08
Jetson TX2 (ms)	3.20	61.67	348*	5.34	70.21
Jetson Nano (ms)	4.62	118.34	596*	9.50	132.46
Model/Binary size	13KB	100KB	341.15MB	5.05MB	5.28MB
Power (Nano)	150mW	320mW	2620mW*	340mW	810mW

median ATE of 38cm (Fig. 20) in untrained home environment, which is still better the alternatives (as seen in §6.3). This performance highlights that RoVaR can be deployed in practice easily compared to BB models that are difficult to realize on low-end platforms with high accuracy in unseen places.

7 DISCUSSION

(1) Reducing RF Deployment Overhead: The infrastructure assistance (anchors) for RF localization is becoming common-place. WiFi APs (e.g. Google Nest products) already support ranging mechanisms (FTM [25]) similar to UWB, albeit with a maximum of 160 MHz bandwidth. While RoVaR is equally applicable to leveraging WiFi-based absolute localization, we believe UWB is likely to become part of our APs as well in future (already available in Apple and Samsung smartphones [6, 7]). Further, one can also reduce the dependence on 3 anchors (either for UWB or WiFi) by leveraging just a single multi-radio (antenna) anchor that provides both angle (AoA) and range information (e.g. NXP UWB chips [8]) to directly localize a device. This single reference anchor could be built as part of an access point, application (e.g. game) controller, just a smart device in the environment to track multiple agents simultaneously.

(2) From Position to Pose: While ORB-SLAM3 can already deliver relative pose estimation, RoVaR's UWB-localization currently supports only position. Employing a multi-radio (two synced radios) design on the device (e.g. one on either side of a headset), can allow the device on the agent to also determine its orientation from RF AoAs. Similar to fusion of position from RF and visual sensors, one can also fuse orientations from the two sensors, to yield absolute pose of the agent. This is part of our next step.

(3) Addition of Inertial Sensors: RoVaR's objective has been to bring dual-layer diversity to sensor fusion and hence focused on two key sensors, cameras and RF (UWB) for the start. One can extend RoVaR to a third popular sensor, namely inertial (IMU) sensors, for which there exists a rich body of sensor fusion work [17, 35, 55]. RoVaR

would employ a third algorithmic branch for IMU-based feature extraction (position and other relevant features). The key adaptation needed to RoVaR's framework would be to consider cross-sensor attention between every pair of the three sensors (cameras, UWB and IMU), before fusing them to bring further robustness and accuracy to multi-agent tracking.

8 CONCLUSION

We tackled the problem of bringing robustness and accuracy to multi-agent tracking in practical, indoor environments, by introducing the framework of dual-layer diversity in sensor fusion. RoVaR, an embodiment of this framework, brought together the multi-agent and high-resolution tracking benefits of active (UWB) and passive (visual odometry) tracking modalities through an intelligent combination of both algorithmic and data-driven techniques to deliver robustness in everyday environments. RoVaR's comprehensive evaluation showcased its significant benefits over prior approaches that rely on a single layer of diversity. We believe RoVaR's hybrid, dual-layer diversity approach to sensor fusion offers an important step in opening the door for exciting multi-agent collaborative applications in everyday indoor environments.

REFERENCES

[1] 2020. Decawave DW1000 USER MANUAL. https://www.decawave.com/sites/default/files/resources/dw1000_user_manual_2.11.pdf.
[2] 2020. Decawave UWB Two-Way Ranging. https://www.decawave.com/product/evk1000-evaluation-kit/.
[3] 2020. Intel RealSense Tracking Camera T265. https://www.intelrealsense.com/tracking-camera-t265/.
[4] 2020. M6E Nano RFID Reader. https://www.sparkfun.com/products/14066.
[5] 2020. Nvidia Jetson TX2 Module. https://developer.nvidia.com/embedded/jetson-tx2.
[6] 2021. Apple's UWB Support and U1 chip. https://support.apple.com/en-us/HT212274.
[7] 2021. Samsung Galaxy S21+ and UWB. https://www.sammobile.com/news/uwb-explained-galaxy-s21-plus-s21-ultra.
[8] 2021. Single chip UWB SR040 UWB Module. https://www.nxp.com/products/wireless/secure-ultra-wideband-uwb/trimension-sr040-uwb-module-with-embedded-rf-connector-asmop1co0r1:ASMOP1CO0R1.
[9] 2022. Apple ARKit. https://developer.apple.com/augmented-reality/arkit/.
[10] 2022. Jetson Nano/TX2 Power Measurements. https://github.com/leonardopsantos/jetsonTX2Power.
[11] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. 2020. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.
[12] Ali J Ben Ali, Zakieh Sadat Hashemifar, and Karthik Dantu. 2020. Edge-SLAM: edge-assisted visual simultaneous localization and mapping. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. 325–337.
[13] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. 2015. Robust visual inertial odometry using a direct EKF-based approach. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 298–304.
[14] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics* (2021).
[15] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. 2019. Selective Sensor Fusion for Neural Visual-Inertial Odometry. arXiv:1903.01534 [cs.CV]
[16] Ka Wai Cheung, Hing-Cheung So, Wing-Kin Ma, and Yiu-Tong Chan. 2006. A constrained least squares approach to mobile positioning: algorithms and optimality. *EURASIP Journal on Advances in Signal Processing* 2006 (2006), 1–23.
[17] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. 2017. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
[18] Ionut Constandache, Sharad Agarwal, Ivan Tashev, and Romit Roy Choudhury. 2014. Daredevil: indoor location using sound. *ACM SIGMOBILE Mobile Computing and Communications Review* 18, 2 (2014), 9–19.
[19] Juan Antonio Corrales, FA Candelas, and Fernando Torres. 2008. Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter. In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 193–200.
[20] Tobias Feigl, Andreas Porada, Steve Steiner, Christoffer Löffler, Christopher Mutschler, and Michael Philippsen. 2020. Localization Limitations of ARCore, ARKit, and Hololens in Dynamic Large-scale Industry Environments.. In *VISIGRAPP (1: GRAPP)*. 307–318.
[21] Karthikeyan Gururaj, Anojh Kumaran Rajendra, Yang Song, Choi Look Law, and Guofa Cai. 2017. Real-time identification of NLOS range measurements for enhanced UWB localization. In *2017 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE, 1–7.

[22] Zakieh S Hashemifar, Charuvahan Adhivarahan, Anand Balakrishnan, and Karthik Dantu. 2019. Augmenting visual SLAM with Wi-Fi sensing for indoor applications. *Autonomous Robots* 43, 8 (2019), 2245–2260.

[23] Sanvidha CK Herath and Pubudu N Pathirana. 2013. Optimal sensor arrangements in angle of arrival (AoA) and range based localization with linear sensor arrays. *Sensors* 13, 9 (2013), 12277–12294.

[24] Zhu Hua, Li Hang, Li Yue, Long Hang, and Zheng Kan. 2014. Geometrical constrained least squares estimation in wireless location systems. In *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, 159–163.

[25] Mohamed Ibrahim, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Richard Howard, Bo Yu, and Fan Bai. 2018. Verification: Accuracy evaluation of WiFi fine time measurements on an open platform. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 417–427.

[26] LS Jayashree, S Arumugam, M Anusha, and AB Hariny. 2006. On the accuracy of centroid based multilateration procedure for location discovery in wireless sensor networks. In *2006 IFIP international conference on wireless and optical communications networks*. IEEE, 6–pp.

[27] Ruoxi Jia, Ming Jin, and Costas J Spanos. 2014. Soundloc: Acoustic method for indoor localization without infrastructure. *arXiv preprint arXiv:1407.4409* (2014).

[28] Seokmin Jung and Woontack Woo. 2004. UbiTrack: Infrared-based user Tracking System for indoor environment. *International Conferece on Artificial Reality and Telexisitence (ICAT04)* (2004), 1345–1278.

[29] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 269–282.

[30] Suren Kumar, Tim K Marks, and Michael Jones. 2014. Improving person tracking using an inexpensive thermal infrared sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 217–224.

[31] Micheal Lanham. 2018. *Learn ARCore-Fundamentals of Google ARCore: Learn to build augmented reality apps for Android, Unity, and the web with Google ARCore 1.0*. Packt Publishing Ltd.

[32] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*. PMLR, 3734–3743.

[33] Ruihao Li, Sen Wang, and Dongbing Gu. 2020. Deepslam: A robust monocular slam system with unsupervised deep learning. *IEEE Transactions on Industrial Electronics* 68, 4 (2020), 3577–3587.

[34] Heinrich W Löllmann, Christine Evers, Alexander Schmidt, Heinrich Mellmann, Hendrik Barfuss, Patrick A Naylor, and Walter Kellermann. 2018. The LOCATA challenge data corpus for acoustic source localization and tracking. In *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 410–414.

[35] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almalioglu, Pedro PB de Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. 2020. milliEgo: single-chip mmWave radar aided egomotion estimation via deep sensor fusion. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 109–122.

[36] Junhai Luo, Liying Fan, and Husheng Li. 2017. Indoor positioning systems based on visible light communication: State of the art. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2871–2893.

[37] Daniel Neuhold, Christian Bettstetter, and Andreas F Molisch. 2019. HiPR: High-precision UWB ranging for sensor networks. In *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 103–107.

[38] Lei Ni, Yuxin Wang, Haoyang Tang, Zhao Yin, and Yanming Shen. 2017. Accurate localization using LTE signaling data. In *2017 IEEE International Conference on Computer and Information Technology (CIT)*. IEEE, 268–273.

[39] Joan Palacios, Guillermo Bielsa, Paolo Casari, and Joerg Widmer. 2018. Communication-driven localization and mapping for millimeter wave networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2402–2410.

[40] Joan Palacios, Paolo Casari, and Joerg Widmer. 2017. JADE: Zero-knowledge device localization and environment mapping for millimeter wave systems. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* (2019).

[42] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. 2007. BeepBeep: A High Accuracy Acoustic Ranging System Using COTS Mobile Devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems* (Sydney, Australia) *(SenSys '07)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/1322263.1322265

[43] Nicolas Ragot, Redouane Khemmar, Adithya Pokala, Romain Rossi, and Jean-Yves Ertaud. 2019. Benchmark of visual slam algorithms: Orb-slam2 vs rtab-map. In *2019 Eighth International Conference on Emerging Security Technologies (EST)*. IEEE, 1–6.

[44] Jesse Read, Luca Martino, Pablo M Olmos, and David Luengo. 2015. Scalable multi-output label prediction: From classifier chains to classifier trellises. *Pattern Recognition* 48, 6 (2015), 2096–2109.

[45] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*. Ieee, 2564–2571.

[46] Antonio Ramón Jiménez Ruiz and Fernando Seco Granja. 2017. Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis. *IEEE Transactions on instrumentation and Measurement* 66, 8 (2017), 2106–2117.

[47] Muhamad Risqi U Saputra, Pedro PB de Gusmao, Chris Xiaoxuan Lu, Yasin Almalioglu, Stefano Rosa, Changhao Chen, Johan Wahlström, Wei Wang, Andrew Markham, and Niki Trigoni. 2020. Deeptio: A deep thermal-inertial odometry with visual hallucination. *IEEE Robotics and Automation Letters* 5, 2 (2020), 1672–1679.

[48] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. 2018. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 1–36.

[49] Arno Solin, Santiago Cortes, Esa Rahtu, and Juho Kannala. 2018. Inertial odometry on handheld smartphones. In *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 1–5.

[50] Shiyu Song, Manmohan Chandraker, and Clark C Guest. 2015. High accuracy monocular SFM and scale correction for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence* 38, 4 (2015), 730–743.

[51] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 573–580.

[52] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. 2017. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications* 9, 1 (2017), 1–11.

[53] Stephen P Tarzia, Peter A Dinda, Robert P Dick, and Gokhan Memik. 2011. Indoor localization without infrastructure using the acoustic background spectrum. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. 155–168.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[55] Raghav H Venkatnarayan and Muhammad Shahzad. 2019. Enhancing indoor inertial odometry with wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–27.

[56] Raghav Hampapur Venkatnarayan and Muhammad Shahzad. 2019. Measuring Distance Traveled by an Object using WiFi-CSI and IMU Fusion. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 1–2.

[57] Eric A Wan and Rudolph Van Der Merwe. 2000. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 153–158.

[58] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7794–7803.

[59] Yan Wang and Xin Li. 2017. The IMU/UWB fusion positioning algorithm based on a particle filter. *ISPRS International Journal of Geo-Information* 6, 8 (2017), 235.

[60] Yunze Zeng, Parth H Pathak, Zhicheng Yang, and Prasant Mohapatra. 2016. Human tracking and activity monitoring using 60 GHz mmWave. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 1–2.

[61] Ji Zhang and Sanjiv Singh. 2014. LOAM: Lidar Odometry and Mapping in Real-time.. In *Robotics: Science and Systems*, Vol. 2.