Using Synthetic Data in Supervised Learning for Robust 5-DoF Magnetic Marker Localization

Mengfan Wu, Thomas Langerak, Otmar Hilliges, and Juan Zarate

1TU Berlin and Huawei Research, 10623 Berlin, Germany
2Department of Computer Science, ETH Zürich, 8092 Zürich, Switzerland

Tracking passive magnetic markers plays a vital role in advancing healthcare and robotics, offering the potential to significantly improve the precision and efficiency of systems. This technology is key to developing smarter, more responsive tools and devices, such as enhanced surgical instruments, precise diagnostic tools, and robots with improved environmental interaction capabilities. However, traditionally, the tracking of magnetic markers is computationally expensive due to the requirement for iterative optimization procedures. Moreover, these methods depend on the magnetic dipole model for their optimization function, which can yield imprecise outcomes due to the model's significant inaccuracies when dealing with short distances between non-spherical magnet and sensor.

Our article introduces a novel approach that leverages neural networks (NNs) to bypass these limitations, directly inferring the marker's position and orientation to accurately determine the magnet's five degrees of freedom (5 DoFs) in a single step without initial estimation. Although our method demands an extensive supervised training phase, we mitigate this by introducing a computationally more efficient method to generate synthetic, yet realistic data using Finite Element Methods simulations. Our novel method uses the rotational symmetry of axis-symmetric magnetic markers to transform the 3-D simulations into 2-D. The benefits of fast and accurate inference significantly outweigh the offline training preparation. In our evaluation, we use different cylindrical magnets, tracked with a square array of 16 sensors. We perform the sensors' reading and position inference on a portable, NN-oriented single-board computer, ensuring a compact setup. We benchmark our prototype against vision-based ground-truth data, achieving a mean positional error of 4 mm and an orientation error of 8° within a 0.2 × 0.2 × 0.15 m working volume. These results showcase our prototype's ability to balance accuracy and compactness effectively in tracking 5 DoFs.

Index Terms—Machine learning (ML), passive magnet localization, permanent magnets, position and orientation tracking.
its simplicity and robustness; however, future work could potentially extend the ideas presented here to other types of magnetic markers.

To track a magnetic marker, we need to map magnetic signals, measured by one or more Hall sensors, to a marker’s 5-degree-of-freedom (5 DoF) position and orientation (rotation). The most straightforward approach involves directly interpolating the reading intensity within the sensor array. In [20] and [21], this approach was used to track a stylus and a gaming object in a 2-D plane. In more complex configurations (e.g., [3], [4], [22]), multiple sensors with known spatial arrangements created an overconstrained system of equations, ensuring a unique solution for the magnet’s state. These triangulation methods are fast but are not robust to sensor noise and limited to positional degrees of freedom.

A more robust approach, which also infers rotation, involves minimizing the difference between a theoretical magnetic field model, based on a magnetic model, and the sensor readings. To achieve this, several non-linear optimization algorithms have been proposed [23], [24]. Taylor et al. [19] implemented an analytical computation of gradients to expedite the optimization process. In [25], their algorithm decoupled the marker’s orientation from its position, enhancing calculation speed and providing guarantees regarding global minima. Nonetheless, these iterative approaches come with significant challenges. First, gradient descent algorithms are computationally demanding, leading to a tradeoff between tracking precision and frequency. Second, iterative non-convex optimization can be susceptible to the non-uniqueness of the solution, potentially converging to wrong local minima. These challenges make these methods highly dependent on their initialization.

Furthermore, these methods assume that the magnetic model, which predicts the theoretical magnetic field, is valid. All the mentioned methods use the first term of the multi-pole series expansion derived from Maxwell’s equations. The magnetic dipole approximation assumes all the magnets to be spherical, providing the simplest explicit expression for the magnetic field concerning distance to the source. However, this approach has significant limitation. For non-spherical magnets, reliable results are only obtained when the magnets are far from the sensor [26]. Crucially, in gradient descent optimization process, a small magnetic field approximation error can lead to substantial positional discrepancies.

Recently, machine learning (ML) has emerged as a valuable approach to circumvent the computational burden of iterative methods. ML, especially neural networks (NNs), excels in approximating non-linearities through sequential multiply–addition operations. In addition, it does not require initial estimates, often achieving reliable results with a single inference. While inference is rapid, creating a large and representative dataset for training to generalize well to unseen samples can be a challenging and time-consuming process.

In [11], [27], and [28], ML was used to predict the locations of tracked magnets using input from magnetic sensors. The data for training the neural network were collected in vivo by placing magnets at various known locations and gathering sensor readings. Generally, the data collection process is lengthy and not scalable to new scenarios and markers. Su et al. [29] used a neural network to track a magnetic marker, though they trained on synthetic data generated by the dipole model. While they eliminate the need for iterative gradient-based solutions, the work did not address the limitations of the underlying model, retaining inaccuracies for non-spherical or closely located magnets. Recently, Parizi et al. [16] and Sasaki and Ohta [30] demonstrate the use of NNs trained on synthetic data to track active markers. Although we share goals with these works, their results applied exclusively to active emitters and did not consistently outperform methods based on iterative optimization.

To address the data limitation challenge, we use finite element methods (FEMs) to simulate the complete set of Maxwell’s equations (see Fig. 1 for a method overview). FEM is computationally intensive, particularly when using a fine mesh for accurate results. Therefore, it is unsuitable for real-time modeling. However, FEM can be leveraged to generate noise-free synthetic datasets for training ML models. In various other fields, the use of FEM-generated synthetic datasets has successfully trained NNs, although not for magnetic tracking. Examples include mechanical deformations [31], elastoplasticity [32], material inspection [33], and nano-structures [34]. To further reduce computation time while increasing dataset size, we focus on axis-symmetric markers. We enable this approach by introducing a coordinate transformation algorithm that capitalizes on our markers’ symmetric properties, converting FEM simulation results from 2-D into 3-D.

To evaluate the efficacy of our system, we evaluate our method in both simulation and experiments. In simulation, we find that our method outperforms iterative methods, in terms of accuracy, robustness, and computational efficiency. We also compare our FEM-based approach, to a method that is trained on magnetic dipole generated data, where our method significantly outperforms the baseline. In real-world experiments, we assess various cylindrical magnets with 5 DoFs using a lower power portable Nvidia Jetson Nano, where sensing and tracking are performed in parallel. We achieve
an averaged error of 4 mm and an orientation error of less than 8°, with an interactive rate of 75 Hz when using eight sensors.

In summary, our work contributes significantly to the magnetic tracking literature in five key ways.

1) We propose a supervised learning approach that enables real-time tracking of arbitrary axis-symmetric magnets using NNs, overcoming the computational limitations of previous iterative solutions by approximating the inverse function of the magnetic field.

2) To address the challenge of data limitation, we leverage FEM to simulate Maxwell’s equations, generating noise-free synthetic datasets for training ML models.

3) We introduce a coordinate transformation algorithm that capitalizes on our markers’ symmetric properties, converting FEM simulation results from 2-D into 3-D, enabling accurate high-resolution magnetic field data for training NNs.

4) Due to the computational efficiency of our approach, we enable low-power and portable applications. We demonstrate this using an Nvidia Jetson Nano.

5) We demonstrate the validity of our approach with in silico and real-world evaluations. These contributions collectively enable more accurate and efficient magnetic tracking solutions, offering the potential for the development of truly portable tracking devices.

II. METHOD

Our contribution revolves around a novel tracking method through supervised learning, specifically using a multi-layer perceptron (MLP). Our tracking pipeline consists of two distinct phases: the training phase involves comparing the MLP’s output to the position and orientation used to generate synthetic sensor readings through FEM, while during the inference phase the sensor readings are inputted into the MLP. Our tracking pipeline consists of two distinct phases: the training phase involves comparing the MLP’s output to the position and orientation used to generate synthetic sensor readings through FEM, while during the inference phase the sensor readings are inputted into the MLP.

In this section, we provide a detailed explanation of our approach, starting with the creation of a high-resolution synthetic dataset (Section II-A). Subsequently, we detail the architecture and training process of our neural network (Section II-B). Finally, we outline the hardware setup (Section II-C).

A. Synthetic Dataset

Our focus is on axisymmetric magnetic markers, encompassing any shape and size of magnets, provided they have rotational symmetry around their magnetization axis. This category encompasses cylinders, spheres, and toroids with arbitrary cross sections, covering the most commonly used types of permanent magnets.

Instead of costly 3-D simulations, we leverage the symmetry to conduct just one high-resolution 2-D FEM simulation for each magnet shape. From a single 2-D FEM simulation, we generate synthetic sensor readings for any location and orientation of the magnet. We achieve the 3-D volume by revolving this 2-D cross section around the magnet’s principal axis. This approach of creating a synthetic dataset significantly enhances computational efficiency during training, striking an optimal balance between detailed simulation and minimal data storage requirements.

We use COMSOL Multiphysics to acquire FEM data. The simulation centers on the magnet, and due to the magnets’ symmetrical properties we constraint it to a single quadrant. We reconstruct the 3-D magnetic B-field, using 2-D FEM generated data and a sampled point. For generating training data and evaluating the neural network, we use the transformation delineated in Algorithm 1 (all the variables are explicated in Nomenclature) with coordinate systems, vectors, and scalars illustrated in Fig. 2. This algorithm converts 2-D FEM data into 3-D synthetic sensor readings as follows. Initially, it requires the magnet’s current position and the locations of each sensor, all within a fixed coordinate system, $C_D$, centered at the array of sensors. Subsequently, we establish a coordinate system, $C_M$, centered on the magnet’s current position and orientation. The three perpendicular axes of $C_M$ are along $u$, $v$, and $w$ in $C_D$. $u$, $v$, and $w$ are then normalized to create coordinate transformation matrix $M$. The algorithm then uses $2D - FEM$ to extract the magnetic flux density, $B^M$ in $C_M$, with the sensor’s coordinates in $C_M$ as $(d_u, 0, d_w)$, where the second coordinate is always 0 since we obtain FEM data in 2-D. Ultimately, we transform $B^M$ back into the original global coordinate system, $C_D$, to be used as features for training the MLP.

B. Tracking With NNs

1) Multi-Layer Perceptron: We use an MLP as our network architecture. The MLP takes in a 3n-element vector as input,
which contains the \((x, y, z)\) magnetic flux densities \(B\) from the \(n\) sensors. Its output is a six-element tuple, comprising the magnet’s position, \(p = [p^x_m, p^y_m, p^z_m]\), and orientation, \(o = [o^x_m, o^y_m, o^z_m]\). Therefore, we define the MLP as a non-linear mapping from the sensor readings to the tracking variables

\[
F : \mathbb{R}^3n \rightarrow \mathbb{R}^6; F(p^x_m, p^y_m, p^z_m, o^x_m, o^y_m, o^z_m) = [p, o].
\] (1)

It is important to note that although we have a 5 DoF output, three in \(p\) and two in \(o\), we represent the orientation vector in the Cartesian coordinates. This approach helps avoid numerical discontinuity, particularly when the azimuth angle transitions from 359° to 0°. Our MLP architecture, depicted in Fig. 1, consists of a three-layer perceptron with 2048 units/layer, excluding the input and output layers. We use ReLU as the activation functions.

2) Pre-Processing: Crucially, the dipole model (2) indicates that the magnetic field decreases with the cube of the distance \((1/r^3)\), where \(r\) represents the distance between the source and the sensors, and \(m\) denotes the dipole moment

\[
B(p_m) = \frac{\mu_0}{4\pi} \left[ \frac{3r(m \cdot r)}{r^5} - \frac{m}{r^3} \right].
\] (2)

We have empirically determined that system training fails to converge when using magnetic readings directly as inputs. This issue may arise because the input values change dramatically, by several orders of magnitude, as the magnet moves from near a sensor (approximately \(10^{-2}\) Tesla) to the boundary of the working volume (around \(10^{-6}\) Tesla), consistent with the predictions of the dipole model where \(|B| \propto 1/r^3\).

To address this challenge, we re-scale the input signals by taking their cubic root, \(f(B) = (|B|^{1/3})B \propto 1/r\). The alteration in distribution is visible in Fig. 3, where the values of \((|B|^{1/3})B\) are distributed on a wider range and more balanced. In general, an MLP benefits from having inputs that are more uniformly spread across the same order of magnitude.

3) Training: We train our NNs using randomly sampled data within a cubic volume measuring \(0.2 \times 0.2 \times 0.15\) m³, where the sensor array covers the bottom face. The magnets’ orientations are sampled as points on a unit sphere and paired with their locations to serve as training labels, as outlined in Algorithm 1.

For our loss function, we use a weighted sum of the positional and orientational differences

\[
L = \|\mathbf{p}_{\text{true}} - \mathbf{p}_{\text{pred}}\|^2 + \eta \frac{\|\mathbf{o}_{\text{true}}\|}{\|\mathbf{o}_{\text{true}}\|} - \frac{\|\mathbf{o}_{\text{pred}}\|}{\|\mathbf{o}_{\text{true}}\|}^2.
\] (3)

Our goal is to minimize both the positional and orientational discrepancies between the predictions and ground truth. We introduce the weight \(\eta\) to balance the differing scales of orientation and position error terms. After parameter tuning, we set \(\eta = 10^{-3}\) as the standard in all the experiments.

C. Implementation

1) Software: We implement the MLP in python with PyTorch. We use Adam [35] with an initial learning rate \(\gamma = 10^{-4}\). We train for 40 epochs, and the learning rate decays by 0.98 after each epoch in training. We generate \(10^6\) random points per epoch to train the model with a batch size of 256. Using Algorithm 1, we generate the training data independently and identically distributed. Data generation and training of the MLP takes \(\sim 1\) h on a standard desktop.

2) Hardware: Our hardware’s foundation consists of 16 tri-axial magnetometers (MLX90393, Melexis). These sensors feature a linear range up to 0.05 Tesla, aligning well with the magnetic fields we anticipate from our test magnets at close range. These sensors can achieve a data output rate of 716.9 Hz.

Moreover, we calibrate our system using the ellipsoid-fitting method referenced in [36]. The arrangement of the 16 sensors follows a \(4 \times 4\) grid pattern, maintaining a 52-mm interval between the centers of adjacent sensors, as depicted in Fig. 4.

To maintain minimal instrumentation and facilitate portable applications, such as prosthetics, we have implemented the tracking inference on an AI-oriented single-board computer (Jetson Nano, NVIDIA). This device weighs only 138 g and also supports direct sensor readings using the I2C communication protocol. The minimum time required to read 16 sensors sequentially is approximately 24 ms.

III. IN SILICO EVALUATION

In this section, we compare the performance of our neural-network-based tracking method with that of an iterative,
gradient-descent-based technique. This comparison uses simulated sensor readings, allowing us to control experimental variables such as the initialization of the iterative, optimization-based algorithm. We first introduce how we sample data points to evaluate both the methods. Then we compare our method versus the iterative baseline in terms of accuracy as function of initialization, as well as computational time. Finally, we compare our method trained on FEM data versus trained on data generated with the dipole model.

**Sampling Method:** For in silico evaluation, we randomly sample magnet positions and orientations and compute the corresponding magnetic flux densities. We either use the magnetic dipole model to sample from or use FEM-generated data. On one hand, using dipole-model-generated data ensures that the characteristics of the synthetic signals match the internal physical model used in the optimization method. On the other hand, the FEM-generated data are more realistic. We initialize the optimization method by offsetting the sampled position and orientation. Offsetting the position and orientation means we initialize the iterative methods with the ground truth plus the offset (see Fig. 5). We expect that a larger offset results in a worse performance for the iterative methods. We configure the iterative algorithm to stop upon reaching the maximum number of iterations. This process is similar to a real-world application, where we would initialize with the last known position and the newly measured magnetic flux densities, and we have a limited computational budget.

**A. Influence of Initialization on the Optimization Methods**

In our simulation study, we conducted a comparison between our method and an optimization-based technique by analyzing 400 data points, sampled as outlined above. For the optimization process, we used PyTorch, configuring the magnet’s position and orientation as adjustable parameters. This setup leveraged automatic differentiation for efficient computation, following the approach detailed in [37]. The optimization algorithm was L-BFGS, a quasi-Newton method. L-BFGS optimizes step size through a line search mechanism [38]. For the optimization-based approach, we used the magnetic dipole model as our internal physical model, integrating it as our loss function in a manner consistent with [19].

Our results are presented in two parts: first, we assess the accuracy of our method compared with the optimization baseline for various initialization errors; and second, we compare the tracking performance of our method trained on FEM data versus data generated by the dipole model.

1) **Accuracy:** The full results of our optimization method are detailed in Fig. 6, showcasing the tracking errors relative to the maximum iteration count across different initialization of position [Fig. 6(a)] and orientation [Fig. 6(b)]. Similar results are shown for the angular error (Fig. 7) For a detailed analysis, we have also provided statistical analysis of the tracking outcomes after 50 iterations in Table 1. Note that our MLP method is a single inference step and does not require multiple iterations or an initialization.

Since the non-converging cases in optimization method yield a long tail in error distribution, the average errors of optimization methods are not representative of the performance. Instead, we evaluate the results with the median value of errors. Our approach showed improved performance with a median positional error of $e_p = 1.40$ mm and a median orientation error of $e_\theta = 3.34^\circ$, significantly outperforming the baseline’s median positional error of $e_p = 26.80$ mm and orientation error of $e_\theta = 29.96^\circ$ under nearly all the test conditions. As expected, the accuracy of the optimization-based approach improved with more iterations. This trend was also observed in relation to the quality of the initialization; more accurate initial conditions led to more precise optimization outcomes. However, our method was only surpassed in scenarios with the highest number of iterations (50) and the optimal
initial conditions, specifically at 80 mm of initial positional error and 10° of initial orientation error, along with 30 mm of initial positional error and 45° of initial orientation error.

Decreasing the number of iterations led to a decrease in performance. With ten iterations, the median positional error increased above 200 mm in certain instances, especially when the initial orientation significantly deviated from the target. We observed that even with initial orientations closely aligned to the target [with an offset of 45°], the resulting positional errors on par with scenarios with a large initialization offset.

2) Running Time: Our multilayer perceptron (MLP) model delivers outcomes after a single inference step, which includes feature engineering, along with additions and multiplications within its hidden layers. On the other hand, optimization-based approaches necessitate the calculation of second-order gradients relative to the estimated positions at every iteration, proceeding until convergence is achieved or the maximum number of iterations achieved. We evaluated the speed differences between these two methodologies. To ensure a fair comparison, we executed both the methods on the same hardware, specifically a laptop powered by an Intel i7-7500U CPU.

Fig. 8 illustrates the comparison between the total processing time of our data-driven MLP approach and the
optimization-based method across various maximum iteration counts. A single inference operation for 5 DoF tracking using the MLP is completed in just 0.8 ms; this time frame includes the feature engineering stage. In contrast, the L-BFGS optimization process demands approximately 1 ms per iteration. As shown in Fig. 6, achieving satisfactory outcomes with the optimization method often requires dozens of iterations, depending on how well the process is initialized. Consequently, the optimization approach can take tens of milliseconds to converge. Thus, it is evident that our method surpasses the iterative optimization baseline in terms of speed.

B. NNs Trained With FEM Versus Dipole Model

Previous works train NNs on data generated with the approximated analytical model [29], [39]. In contrast, we propose using FEM to generate a dataset and take full advantage of the powerful representation of NNs for non-linear systems. To ensure a fair comparison, we distinguish two scenarios for training the MLP: 1) using FEM data and 2) using the magnetic dipole model directly, which is similar to the model used in the optimization method.

Our evaluation encompasses six different magnet shapes, similar to those in our practical experiments (Section IV). Including, a variety of shapes from cylinders to disks, all magnetized along their principal axis, and we also assess performance on a spherical magnet, where the dipole model theoretically provides an exact solution. For each magnet shape, we generated a dataset consisting of 1000 samples. Fig. 9(a) presents a comparison of the accuracy between MLPs trained on two synthetic datasets: one created from FEM simulations, as explained in Section II-A, and the other based on the magnetic dipole approximation (2). In addition, Fig. 9(b) illustrates the angular error measurements. Both the FEM and dipole trained methods perform well in terms of the rotational error (see Fig. 7). Overall, this is an indication that our FEM-based approach generalizes well across all the shapes.

To ensure the validity of our data analysis, we first conducted a Shapiro–Wilk test to confirm the normal distribution of our data. Subsequently, we applied a Student’s T-test to identify significant differences between MLPs trained with dipole- and FEM-generated datasets.

The results clearly demonstrate that our FEM-based approach significantly outperforms the dipole-model-based baseline in positional accuracy for cylindrical and disk magnets across all the tests (with all $p < 0.005$), while showing no significant difference for the spherical magnet ($p = 0.95$). These results are as expected, since there is no difference between the dipole model and the FEM data for a sphere. We observed greater differences in the results as the shape of the magnet distances itself from the sphere, thereby reinforcing our hypothesis of the superiority of our method for magnets not shaped like spheres.

IV. EXPERIMENTAL EVALUATION

We compare the results of our MLP tracking method to experimental ground-truth data collected with OptiTrack. We not only evaluate positional and orientation accuracy but also evaluate the computational performance on a lightweight portable computer; enabling application such as prosthetics. We experiment with different numbers of Hall sensors in our system (4, 8, 12, or 16), to evaluate the computational speed versus accuracy tradeoff.

A. Experimental Setup

We equipped the permanent magnet with OptiTrack markers for precise tracking, illustrated in the inset of Fig. 4. The setup also features ten OptiTrack cameras, and the cylindrical magnets used are identical to those described in Section III-B for simulation purposes.

In our experiments, we manipulate the magnet above the sensor grid, allowing for free movement and tilting, while ensuring optimal visibility of the optical markers to
the cameras. This movement is executed at a pace similar to joystick gaming, providing a dynamic test environment. To address any discrepancies in timing between the magnetic and optical tracking systems, we implement a calibration process that involves adjusting a time-offset variable within the magnetic signal data. This adjustment aims to minimize tracking errors. Furthermore, we use piecewise cubic hermite interpolating polynomial (PCHIP) interpolation [40], to reconcile any discrepancies in sampling frequencies between the two systems. This approach ensures the synchronization and accuracy of our tracking data across both the modalities.

B. Results

1) Accuracy: Fig. 10 presents a comparison of a single trajectory as tracked by both our MLP method and the OptiTrack system, aligned within the same time frame. This specific trajectory involves a magnet with dimensions of 10 mm in diameter and 20 mm in height. Throughout the tracking period, the deviation between the two trajectories remained below the 4 mm threshold for more than half of the duration, with the largest discrepancies occurring during rapid maneuvers but not exceeding 10 mm. It is important to note that the discontinuities observed in the OptiTrack trajectories can be attributed to partial occlusion of the infrared markers.

We show (Fig. 11) the positional error as a function of distance from the center of the sensor array, demonstrating the robustness of our tracking method at extended ranges. Furthermore, we see that the rotation error is stable for all the distances, once again highlighting the robustness of our method.

In Fig. 12, we detail the tracking performance statistics for five different cylindrical magnets and number of sensors. Consistent with expectations, the use of a greater number of sensors results in reduced tracking errors for all the magnets tested. With the entire array of 16 sensors activated, the average errors in both position and orientation typically fall below 4 mm and 8°, respectively, with the exception of the pole-shaped magnet (with dimensions of 5 mm in diameter and 25 mm in height).

2) Computational Time: Finally, we evaluated the responsiveness of our tracking method by comparing the performance of inferences executed on the CPU versus the GPU of the Jetson Nano. It is crucial to highlight that sensor data collection was consistently handled by the CPU through the I2C communication protocol, taking an average time of 1.75 ms for each sensor read.

When focusing on the inference aspect, running the process on the CPU took approximately 28 ms, and this duration remained fairly consistent regardless of the increases in the input feature size (i.e., number of sensors). On the other hand, executing the inference on the GPU showed a variation in processing times, starting at 10 ms with a setup of four sensors and extending up to 15 ms when using the full array of 16 sensors. This variation underscores the GPU’s ability to efficiently manage larger datasets, albeit with a modest increase in processing time as the number of sensors—and consequently, the dimensionality of the input data—grows.

V. APPLICATION EXAMPLE

To show applicability of our approach, we implemented it on a novel haptic input and output device (Fig. 13) [5].
implementation consists of eighth Hall sensor configured in a circle on two different level planes. The positions and angles are tracked directly on the Jetson Nano. The latency is below 40 ms and the tracking frequency reaches up to 83 Hz, which is more than sufficient for most interactive applications. We refer to the supplementary material for a video of the demonstration.

VI. DISCUSSION

In Section III-A, we identified a known limitation of iterative methods: tracking accuracy heavily depends on the initial estimation, particularly the magnet’s initial orientation. The optimization often becomes trapped in a local minimum if the initial orientation estimate significantly deviates from the actual value.

However, as Fig. 6 demonstrates, with a reasonably accurate initial orientation (mismatch less than 45°), the convergence is less susceptible to other perturbations. The results after 50 iteration steps, as seen in Table I, reveal that the third quartile errors are as significant as the initialization errors. This finding suggests that the optimization method might not reach the global minimum even after 50 epochs.

In contrast, tracking with NNs is independent of initialization. Table I shows that the MLP can surpass the optimization-based method in all cases except those with the most accurate initial estimations and the maximum number of iterations. Furthermore, both the third quartile positional and orientation errors are consistently within acceptable limits when using MLP, underscoring its stability.

Remarkably, we were able to directly apply the models trained with simulations in experimental tests without adjusting any hyperparameters. However, we observe a simulation-to-real gap where performance declines when training on synthetic data and evaluating on real data. This is likely due to sensor and background noise, or because our real-world magnetic markers are not perfect. For future work, we consider incorporating background and sensor noise during neural network training, akin to approaches used in other magnetic tracking systems [41], [42]. Furthermore, we can fine-tune on real-world data. Finally, future work could include investigating additional inputs to the neural network that describe the properties of the real-world magnet (such as a magnetization scalar). These adaptations will likely diminish the simulation-to-reality gap, enhancing the accuracy of the MLP method.

In Section III-B, we examined the impact of using training datasets generated either through FEM or the magnetic dipole model. As anticipated, models trained with the dipole method showed performance more closely aligned with FEM-trained models when the magnet’s shape was closer to or exactly a sphere. For cylindrical magnets, models trained with FEM simulations improved tracking accuracy by 0.2–1.2 mm. However, we also noted that the tracking performance of the MLP degraded as the magnet shapes deviated further from a spherical form (see Fig. 12). The bigger errors in pole-shaped magnet might be because of the relatively small magnetic momentum. The volume of the magnet is smaller, so the magnetic field around it is weaker. Consider the same level of noise for sensors, the signal-noise-ratio (SNR) for the pole-shaped magnet is smaller and leads to worse performance.

We observed that the total time for the one-shot inference in MLP was comparable to each iteration of the many required by iterative methods. Notably, the MLP can be activated sporadically on demand, without the need to continuously track and lock the target to ensure correct convergence within a few iteration steps. We also demonstrated the feasibility of implementing an MLP tracking algorithm on a portable, energy-constrained device, such as the Jetson Nano. We found that the inference time using the GPU was about the same as reading eight sensors via the I2C protocol on the CPU. Thus, the sensor reading process currently limits the refresh rate of our prototype. Alternative protocols such as SPI could potentially alleviate this bottleneck.

A limitation of data-driven methods is the requirement to retrain the neural network for each new condition, such as different numbers and placements of sensors or changes in the magnet’s shape. Although our training process, including data generation, takes only about 1 h, this requirement could hinder applications that necessitate online optimization of sensor location. The training process could be facilitated with certain tradeoff on data variety by storing and reusing generated data points in each epoch of training, rather than generating new data. Other promising directions for future research include using NNs to track multiple magnets, using neural network predictions to initialize optimization-based methods, and exploring the use of recurrent NNs to enhance temporal consistency.

VII. CONCLUSION

In this article, we demonstrated the accuracy and efficiency of using NNs to predict the location and orientation of magnets directly. We combined 2-D FEM-simulated data with a coordinate transformation algorithm to generate synthetic training data on demand for any type of axis-symmetric magnet. The tracking performance of NNs was stable and did not experience the convergence issues often seen in optimization-based tracking methods. Our experiments also showed that it is feasible to move the tracking algorithms to energy-restricted devices, thereby enabling portable interactive magnetic applications.

ACKNOWLEDGMENT

This work was conducted at the Department of Computer Science at ETH Zürich. This work was supported in part
by European Research Council (ERC) under Grant OPTINT StG2016-717054. Mengfan Wu and Thomas Langerak contributed equally to this work.

**References**

[1] X. Han, H. Seki, Y. Kamiya, and M. Hikizu, “Wearable handwriting input device using magnetic field,” in Proc. SICE Ann. Conf., Sep. 2007, pp. 365–368.

[2] Y. Ma, Z.-H. Mao, W. Jia, C. Li, J. Yang, and M. Sun, “Magnetic hand tracking for human-computer interface,” IEEE Trans. Magn., vol. 47, no. 5, pp. 970–973, May 2011.

[3] K.-Y. Chen, S. N. Patel, and S. Keller, “Finexus: Tracking precise motions of multiple fingertips using magnetic sensing,” in Proc. CHI Conf. Hum. Factors Comput. Syst., May 2016, pp. 1504–1514.

[4] J. McIntosh, P. Strohmeyer, J. Knibbe, S. Boring, and K. Hornbæk, “Magnetips: Combining fingertip tracking and haptic feedback for around-device interaction,” in Proc. CHI Conf. Hum. Factors Comput. Syst., May 2019, pp. 1–12.

[5] T. Langerak, I. J. Zárate, D. Lindlbauer, C. Holz, and O. Hilgios, “Omni: Volumetric sensing and actuation of passive magnetic tools for dynamic haptic feedback,” in Proc. 33rd Ann. ACM Symp. User Interface Softw. Technol., 2020, pp. 594–606.

[6] D. Chen et al., “MagX: Wearable, untethered hands tracking with passive magnets,” in Proc. 27th Ann. Int. Conf. Mobile Comput. Netw., Oct. 2021, pp. 269–282.

[7] T. Langerak, I. J. Zárate, V. Vechev, D. Lindlbauer, D. Panozzo, and O. Hilgios, “Optimal control for electromagnetic haptic guidance systems,” in Proc. 33rd Ann. ACM Symp. User Interface Softw. Technol., Oct. 2020, pp. 951–965.

[8] A. Abler, J. Zárate, T. Langerak, V. Vechev, and O. Hilgios, “Hedgehog: Handheld spherical pin array based on a central electromagnetic actuator,” in Proc. IEEE World Haptics Conf. (WHC), Jul. 2021, pp. 133–138.

[9] S. Krueger, H. Timinger, R. Grewer, and J. Borgert, “Modality-integrated catheter tracking for X-ray vascular interventions,” Phys. Med. Biol., vol. 50, no. 4, pp. 581–597, Feb. 2005.

[10] M. Salerno, R. Rizzo, E. Sinibaldi, and A. Menciassi, “Force calculation of multiple permanent magnets,” IEEE Trans. Robot., vol. 21, no. 2, pp. 708–716, Apr. 2016.

[11] Z. Sun, L. Maréchal, and S. Foong, “Passive magnetic-based localization of an object magnetic tracking for multi-arm continuum robots,” IEEE Trans. Magn., vol. 50, no. 4, pp. 581–597, Aug. 2021.

[12] A. J. Petruska and J. J. Abbott, “Optimal permanent-magnet geometries for dipole field approximation,” IEEE Trans. Magn., vol. 49, no. 2, pp. 811–819, Feb. 2013.

[13] J. Mach., “Automatic differentiation in machine learning: A survey,” arXiv:1412.6980, Jul. 2020.

[14] T. Langerak, J. Zárate, D. Lindlbauer, C. Holz, and O. Hilgios, “Omni: Volumetric sensing and actuation of passive magnetic tools for dynamic haptic feedback,” in Proc. 33rd Ann. ACM Symp. User Interface Softw. Technol., 2020, pp. 594–606.

[15] D. Chen et al., “MagX: Wearable, untethered hands tracking with passive magnets,” in Proc. 27th Ann. Int. Conf. Mobile Comput. Netw., Oct. 2021, pp. 269–282.

[16] T. Langerak, J. J. Zárate, V. Vechev, D. Lindlbauer, D. Panozzo, and O. Hilgios, “Optimal control for electromagnetic haptic guidance systems,” in Proc. 33rd Ann. ACM Symp. User Interface Softw. Technol., Oct. 2020, pp. 951–965.

[17] A. Abler, J. Zárate, T. Langerak, V. Vechev, and O. Hilgios, “Hedgehog: Handheld spherical pin array based on a central electromagnetic actuator,” in Proc. IEEE World Haptics Conf. (WHC), Jul. 2021, pp. 133–138.

[18] S. Krueger, H. Timinger, R. Grewer, and J. Borgert, “Modality-integrated catheter tracking for X-ray vascular interventions,” Phys. Med. Biol., vol. 50, no. 4, pp. 581–597, Feb. 2005.

[19] M. Salerno, R. Rizzo, E. Sinibaldi, and A. Menciassi, “Force calculation of multiple permanent magnets,” IEEE Trans. Robot., vol. 21, no. 2, pp. 708–716, Apr. 2016.

[20] Z. Sun, L. Maréchal, and S. Foong, “Passive magnetic-based localization of an object magnetic tracking for multi-arm continuum robots,” IEEE Trans. Magn., vol. 50, no. 4, pp. 581–597, Aug. 2021.

[21] A. J. Petruska and J. J. Abbott, “Optimal permanent-magnet geometries for dipole field approximation,” IEEE Trans. Magn., vol. 49, no. 2, pp. 811–819, Feb. 2013.

[22] J. Mach., “Automatic differentiation in machine learning: A survey,” arXiv:1412.6980, Jul. 2020.
Mengfan Wu received the B.E. degree in telecommunication from Tongji University, Shanghai, China, in 2018, and the M.S. degree in electrical engineering and information technology from ETH Zürich, Zürich, Switzerland, in 2021. He is currently pursuing the Dr.Ing. degree with the Telecommunication Networks Group, TU Berlin, Berlin, Germany.

He is currently with the Advanced Wireless Technology Laboratory, Huawei Munich Research Center, Munich, Germany.

Thomas Langerak received the B.Sc. degree in industrial design from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2016, and the dual M.Sc. degree in computer science from Aalto University, Helsinki, Finland, and Twente University, Enschede, The Netherlands, in 2018. He is currently pursuing the Ph.D. degree with the Advanced Interactive Technologies Group, ETH Zürich, Zürich, Switzerland.

His research interests include HCI and haptics.

Otmar Hilliges received the Diploma degree in computer Science from Technische Universität München, Munich, Germany, and the Ph.D. degree in computer science from LMU München, Munich, in 2009.

Then he joined the I3D Group, Microsoft Research Cambridge, Cambridge, U.K., first as a Post-Doctoral and later as a Researcher. Since 2013, he has been with ETH Zürich, Zürich, Switzerland. He is a Full Professor with the Computer Science Department, leads the AIT Laboratory, Denton, TX, USA, and serves as the Head of the Institute of Intelligent Interactive Systems. His research covers machine perception, computer vision, and machine learning.

Juan Zarate received the M.Sc. and Ph.D. degrees in physics from Instituto Balseiro, Bariloche, Argentina, in 2007 and 2013, respectively working on microfabricated infrared sensors.

In 2014, he joined the Soft Transducers Laboratory, EPFL, Lausanne, Switzerland, to work on haptic displays. Since 2018, he has been part of the Advanced Interactive Technologies Group, ETH Zürich, Zürich, Switzerland. His current research covers volumetric video, haptics, and HCI.