DNN Initialized Inverse Kinematics

Dandi Desta
John A. Paulson School of Engineering and Applied Sciences
Harvard University
Cambridge, MA, USA

***Abstract*—Efficient inverse kinematics (IK) remains a central challenge in real-time robotic manipulation, particularly when iterative solvers must operate under tight latency and reliability constraints. Although classical numerical methods can achieve high precision, their performance depends strongly on initialization and can degrade near singularities, local minima, or competing kinematic branches. Conversely, purely learned IK methods offer fast inference but often lack the accuracy required for precision control. This paper introduces a hybrid framework for the Universal Robots UR5e in which a seed-conditioned deep neural network (DNN) predicts an informed initial joint configuration from a target SE(3) end-effector pose and the robot's current joint state, and a Newton-Raphson refinement stage then recovers a precise solution. Trained on a 10-million-sample synthetic dataset, the proposed model leverages current-state conditioning to improve branch selection and place the solver within a favorable convergence basin. Preliminary experiments indicate that, compared with best-of-5 random initialization, the method improves convergence reliability by 78% and reduces average solver iterations by 5.7×, with only ∼0.5 ms inference overhead. The results demonstrate that learned warm-starting can significantly improve the efficiency and robustness of iterative IK, offering a practical path toward integration with existing high-precision robotics pipelines.**

***Index Terms*—Inverse Kinematics, Deep Learning, UR5e, Newton-Raphson, Manipulator Dynamics**

I. Introduction

Inverse kinematics (IK) is a fundamental problem in robot manipulation. Given a desired end-effector pose, an IK solver computes a joint configuration that places the robot at that pose. For 6-DOF manipulators such as the Universal Robots UR5e, IK must be solved repeatedly during reaching, grasping, trajectory tracking, and motion planning. Although IK is a classical problem, practical performance remains highly dependent on the solver used and, in particular, on the quality of its initialization. In real-time or repeated-query settings, even modest reductions in solve effort can significantly improve responsiveness, reliability, and overall control performance.

Classical numerical IK methods, including Newton-Raphson and Levenberg-Marquardt-type solvers, can recover highly accurate solutions, but they are often sensitive to the initial joint estimate. Poor initialization may lead to slow convergence, convergence to an undesirable solution branch, or outright failure near singularities and joint-limit-constrained regions. These challenges are especially important for manipulators such as the UR5e, where multiple valid joint configurations may correspond to the same end-effector pose. A solver that is both precise and consistently well-initialized is therefore highly desirable for practical robotic control.

Recent learning-based methods offer an appealing alternative by directly predicting joint angles from a target pose using neural networks. These approaches can provide microsecond-scale inference speed, but when used alone they often do not achieve the sub-millimeter precision and reliability required for high-accuracy manipulation. This creates a natural opportunity for hybrid methods that combine the speed of learned prediction with the accuracy of classical numerical refinement.

In this work, we propose a hybrid IK framework for the UR5e in which a seed-conditioned deep neural network predicts a high-quality initial joint configuration, which is then refined by a classical Newton-Raphson solver. Rather than replacing numerical IK, the proposed method is designed to assist it. The network takes as input both the desired end-effector pose and the robot's current joint state, allowing it to produce an initialization that is not only close to a valid solution, but also biased toward a branch consistent with the robot's current configuration. This improves the likelihood that iterative refinement begins inside a favorable convergence basin.

To evaluate this idea, we generate a large-scale synthetic dataset of reachable UR5e poses and associated joint configurations using forward kinematics. The model is trained on pose and seed-state pairs and learns to predict joint configurations in a representation suitable for stable angular learning. At inference time, the learned prediction serves as a warm start for numerical IK, and the resulting hybrid pipeline is benchmarked against conventional random initialization strategies.

The main contribution of this paper is a seed-conditioned learned initialization framework for iterative IK on the UR5e. More broadly, this work shows that deep learning can improve classical robotic solvers not by replacing them, but by acting as a lightweight front-end that improves convergence behavior while preserving numerical precision. Empirical evaluations demonstrate substantial gains in convergence reliability and iteration efficiency, suggesting that learned warm-starting is a practical direction for real-time robotic manipulation pipelines.

The remainder of this paper is organized as follows. Section II reviews relevant background and related work in inverse kinematics and learning-based initialization. Section **??** presents the proposed methodology, including dataset generation, network design, and numerical refinement. Section **??** describes the experimental setup and evaluation protocol. Section **??** reports the empirical results, and Section **??** concludes

This work was completed for ES259 Advanced Robotics.

the paper.

II. Background and Related Work

A. Problem Formulation and Kinematics

The forward kinematics of a serial manipulator defines a non-linear mapping $f : \mathbb{R}^n \to SE(3)$ from a vector of joint angles θ to the Cartesian end-effector pose x. The inverse kinematics (IK) problem seeks the inverse mapping, $f^{-1}(x)$, to determine the necessary joint configuration for a desired target pose. For a 6-DOF manipulator like the Universal Robots UR5e, the solution space is non-linear and heavily constrained by physical joint limits. Because the mapping is not bijective, a single Cartesian pose may correspond to multiple valid joint configurations, commonly referred to as kinematic branches.

The differential relationship between joint velocities and end-effector velocities is governed by the manipulator Jacobian matrix $J(\theta) \in \mathbb{R}^{6\times n}$, such that $\dot{x} = J(\theta)\dot{\theta}$ [1]. Because closed-form analytical solutions are highly specific to individual robot geometries and often struggle with generalized constraints, practical control frameworks typically rely on numerical methods driven by this Jacobian formulation.

B. Classical Numerical Solvers

Traditional numerical IK solvers frame the problem as an iterative minimization of Cartesian error. Algorithms such as Newton-Raphson calculate iterative joint updates using the inverse or pseudoinverse of the Jacobian:

$$\theta_{k+1} = \theta_k + J^{\dagger}(\theta_k)\Delta x \tag{1}$$

While these methods can achieve exceptionally high precision, their performance is fundamentally limited by their reliance on local first-order approximations [2]. When a manipulator is fully extended or the initial guess θ_0 places the robot far from the target, higher-order kinematic effects dominate, causing the first-order Jacobian approximation to break down and inducing endless solver oscillation or jitter [2].

Furthermore, the pseudoinverse method suffers from severe numerical instability in the neighborhood of kinematic singularities [1]. Near a singularity, the Jacobian loses full row rank [2], and even microscopic Cartesian adjustments demand impossibly large changes in joint angles, leading to solver divergence [1].

In real-world applications, hard physical joint limits compound these mathematical vulnerabilities. Joint limits render the search space non-smooth; as iterative solvers step along the gradient to minimize Cartesian error, they frequently push joints beyond their physical boundaries [3]. The necessary repeated clamping of these values leads the solver down "garden paths," permanently trapping the algorithm in local minima even when a valid solution exists elsewhere in the workspace [3]. While state-of-the-art classical frameworks like TRAC-IK mitigate these issues by concurrently running pseudoinverse and optimization-based solvers [3], their convergence speed and reliability remain acutely sensitive to the quality of the initial joint seed.

C. Learning-Based Inverse Kinematics

To circumvent the computational bottlenecks and singularity traps of numerical solvers, recent research has explored Deep Neural Networks (DNNs) to directly approximate $f^{-1}(x)$. Architectures ranging from Multilayer Perceptrons (MLPs) to Recurrent Neural Networks (RNNs) [4] have been successfully trained to map target poses directly to joint angles with microsecond latency. Because these networks learn the global non-linear manifold of the manipulator, they are inherently immune to local singularities and do not rely on matrix inversions [4]. Studies evaluating data representations have further demonstrated that feeding orientation data as unit quaternions significantly enhances network performance compared to larger rotation matrices [4].

Despite their exceptional inference speeds, pure DNN approaches are fundamentally limited by their precision. While a network can rapidly output a configuration that is globally close to the target, it struggles to guarantee the sub-millimeter tolerances required for precise industrial manipulation. Furthermore, when deploying these models on physical hardware, slight inaccuracies in the predicted joint-space target amplify into significant Cartesian orientation errors at the end-effector (Sim2Real translation error) [4]. Consequently, pure learning-based methods are insufficient as standalone IK solvers.

D. Hybrid Approaches and the Proposed Framework

The complementary strengths of numerical and learning-based methods present a natural opportunity for hybrid architectures. While some recent frameworks have utilized neural networks to provide a "warm start" for classical solvers, the vast majority of these models predict initializations conditioned strictly on the target Cartesian pose.

Our proposed framework addresses a critical gap by conditioning the DNN prediction on both the target pose *and* the robot's current joint state. By incorporating current-state context, the network is forced to learn the optimal kinematic branch, predicting an initialization that minimizes required joint movement. This seed-conditioned prediction effectively bypasses the failure modes of classical solvers by directly dropping the Newton-Raphson refinement stage into a limit-compliant, singularity-free convergence basin, amortizing the cost of global search while preserving absolute mathematical precision.

III. Methodology

The proposed framework utilizes a data-driven approach to map task-space coordinates to a limit-compliant, singularity-free joint-space initialization, which is subsequently refined by a classical numerical solver. Based on extensive architectural ablations, we designate a 10-million-sample seed-conditioned Residual Multilayer Perceptron (ResMLP) as our canonical production model.

A. Robot Kinematics and Problem Formulation

The target hardware for this study is the Universal Robots UR5e, a 6-DOF revolute manipulator with a maximum reach

of approximately $850\,\mathrm{mm}$. The forward kinematics (FK) are computed utilizing the Product of Exponentials (PoE) formulation. Given a desired end-effector transformation $T_{\text{target}} \in SE(3)$, the inverse kinematics (IK) problem requires finding a joint configuration $q = [q_1, \ldots, q_6]^T$ such that $f(q) = T_{\text{target}}$.

Because the UR5e admits up to eight valid IK solutions for a given pose and is subject to Jacobian singularities, purely numerical solvers such as the Newton-Raphson method exhibit high sensitivity to initial conditions [1]. We hypothesize that a deep neural network (DNN)—conditioned on both the target pose and the manipulator's current joint state—can predict an informed initial estimate that consistently falls within the convergence basin of a damped Newton-Raphson solver [2].

B. Data Generation and Representation

To train the neural network, a synthetic dataset comprising 10 million kinematically reachable poses was generated. For each sample, uniform random joint angles q^* were passed through a vectorized PoE implementation to rapidly compute the corresponding ground-truth Cartesian pose. To ensure continuous, differentiable representations suitable for stable deep learning, the following data formulations were established:

- **Task-Space Representation (9D):** To circumvent gimbal lock and $SO(3)$ discontinuities inherent to Euler angles and quaternions, the 3D translation vector is concatenated with a 6D continuous rotation representation (comprising the first two columns of the 3×3 rotation matrix), forming a 9D task-space vector.
- **Joint-Space Representation (12D):** To eliminate the $\pm\pi$ wraparound discontinuity associated with angular data, the ground-truth joint angles are encoded as 12D sine/cosine pairs.
- **Seed Conditioning Strategy:** To provide the network with current-state context, a simulated current joint configuration, q_{seed}, is synthesized by perturbing the ground-truth joints with Gaussian noise and clamping the values to physical joint limits:

$$q_{\text{seed}} = \text{clip}(q^* + \mathcal{N}(0, \sigma^2), q_{\min}, q_{\max}), \tag{2}$$

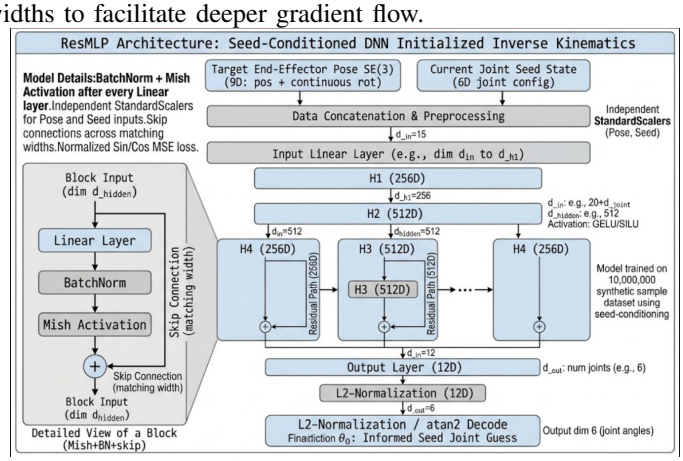
where $\sigma = 0.5\,\mathrm{rad}$. This perturbation simulates a realistic, noisy controller state residing in the neighborhood of a valid solution. Independent standard scalers were subsequently fit to both the 9D pose and 6D seed inputs.

C. Model Architecture

Preliminary architectural ablations utilizing standard MLPs, deep ResMLPs, and Mixture-of-Experts failed to reduce Cartesian error below $360\,\mathrm{mm}$. Purely pose-conditioned networks lack necessary context regarding the robot's current configuration, forcing the model to minimize a scalar Mean Squared Error (MSE) across the divergent physical solutions of the UR5e's eight kinematic branches [4]. Consequently, the network predicts an expected geometric value that typically corresponds to a physically invalid configuration.

To resolve this multi-modality, the proposed architecture employs a seed-conditioned ResMLP. The network receives a concatenated 15D input (9D pose + 6D seed). The inclusion of the seed state breaks the kinematic multi-modality by allowing the network to explicitly learn the local solution branch closest to the current configuration.

The model contains 535,820 trainable parameters, projecting the 15D input through sequential linear layers with widths of $[256, 512, 512, 256]$, terminating in a 12D linear output layer. Mish activations and Batch Normalization are applied after every hidden layer, which demonstrated vastly superior training stability compared to GELU and LayerNorm baselines. Residual skip connections are employed across matching widths to facilitate deeper gradient flow.



Proposed seed-conditioned ResMLP architecture for informed inverse kinematics initialization.

D. Training Methodology

The 10-million-sample dataset was partitioned into 80% training, 10% validation, and 10% testing subsets. The model was optimized using Adam with an initial learning rate of 1×10^{-3}, governed by a learning rate scheduler (ReduceLROnPlateau) and early stopping criteria.

A critical component of the training pipeline is a custom Normalized Sine/Cosine Loss function. This function applies L_2-normalization to each predicted sine/cosine pair prior to computing the MSE against the target:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{6} \left\| \frac{\hat{v}_{i,j}}{\|\hat{v}_{i,j}\|_2} - v^*_{i,j} \right\|_2^2. \tag{3}$$

where $\hat{v}_{i,j} \in \mathbb{R}^2$ is the predicted 2D vector for the j-th joint of the i-th sample, and $v^*_{i,j} = [\sin q^*_{i,j}, \cos q^*_{i,j}]^T$ is the ground-truth target. This operation projects the network's predictions strictly onto the unit circle, preventing the model from achieving artificially low loss gradients via physically impossible trigonometric outputs.

E. Inference and Refinement Pipeline

During real-time deployment, the hybrid solver operates in two sequential stages:

1) **DNN Initialization (∼0.5 ms):** The target pose and current joint state are scaled and passed through the DNN. The 12D output is L_2-normalized per joint pair

and decoded into a 6D initial joint estimate q_{hat} using the two-argument arctangent function (atan2).

2) **Newton-Raphson Refinement (∼1.0 ms):** The neural estimate q_{hat} initializes a damped least-squares Newton-Raphson solver [2], which computes the spatial Jacobian $J \in \mathbb{R}^{6\times6}$ via PoE screw axes. The solver iteratively refines the joint angles by computing the update step Δq:

$$\Delta q = J^T(JJ^T + \lambda^2 I)^{-1}e, \tag{4}$$

where $e \in \mathbb{R}^6$ is the Cartesian error twist and $\lambda^2 = 1 \times 10^{-4}$ is the damping factor. The iterative updates are restricted by a step clamp of $|\Delta q_k| \leq 0.5\,\text{rad}$ per iteration. The algorithm terminates successfully when position error is $< 1\,\text{mm}$ and rotation error is $< 0.573°$.

IV. Experimental Setup and Evaluation Protocol

To rigorously evaluate the proposed hybrid framework, we benchmarked the seed-conditioned DNN against standard initialization strategies using a suite of 1,000 randomly generated, kinematically feasible UR5e target poses.

A. Hardware and Baseline Definition

All training, data generation, and inference benchmarking were conducted in a CPU-only environment to simulate the constrained compute conditions often found in real-time industrial controllers.

To provide a robust and fair baseline, the proposed DNN-initialized solver was evaluated against a classical Newton-Raphson solver utilizing a "best-of-5" random initialization strategy. Rather than using a single random start—which fails the vast majority of the time—the baseline draws five independent uniform random configurations from $[-\pi, \pi]^6$ and returns the one that converges first.

B. Evaluation Metrics

The system was evaluated across two primary domains:

- **Network Accuracy (Joint-Space):** The raw predictive capability of the DNN was measured using the Mean Absolute Error (MAE) in degrees between the predicted joints (after decoding from sine/cosine pairs) and the ground-truth joints.
- **Solver Efficiency (Task-Space):** The efficacy of the hybrid pipeline was measured by tracking the ultimate convergence rate (%) and the number of solver iterations required to reach the target pose. A trial was strictly defined as "converged" only if the final joint configuration achieved a position error of $< 1\,\text{mm}$ and a rotation error of $< 0.573°$ $(0.01\,\text{rad})$ within a maximum budget of 200 iterations.

V. Results

A. The Necessity of Seed Conditioning

Before evaluating the final hybrid solver, architectural ablations were performed to quantify the impact of current-state seed conditioning. Wide and deep ResMLP variants trained exclusively on the 12D task-space pose completely failed to converge on a valid solution, plateauing at a Cartesian position error of approximately $360\,\text{mm}$.

Because a single Cartesian pose corresponds to multiple valid joint configurations, scalar MSE loss forces unconditioned networks to predict a geometric centroid between the competing kinematic branches. Introducing the 6D seed vector effectively collapsed this multi-modality, allowing the network to target a specific local branch and achieve highly accurate joint predictions.

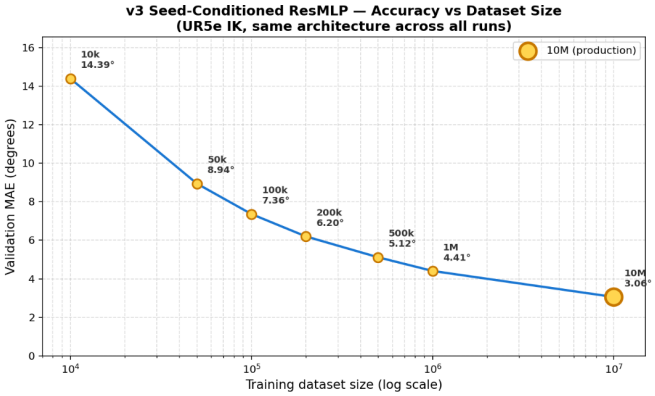


Fig. 1. Log-linear scaling performance of the seed-conditioned ResMLP. The validation Mean Absolute Error (MAE) decreases consistently as the dataset size scales from 10^4 to 10^7 samples.

B. Data Scaling Performance

The production seed-conditioned ResMLP was trained across varying dataset sizes from 10,000 to 10 million samples to observe scaling behavior.

Network accuracy exhibited near log-linear improvement with respect to data volume. Every tenfold increase in dataset size roughly halved the validation MAE. At 10,000 samples, the network achieved a validation MAE of $14.39°$. At 1 million samples, this dropped to $4.41°$. At the full 10-million sample scale, the network reached a validation MAE of $3.06°$. Notably, the gap between training and validation loss closed entirely at the 10-million scale, indicating that the model successfully learned the underlying kinematic manifold without overfitting.

==========

C. IK Benchmark Performance

The 10-million sample model was subsequently deployed as the warm-start initializer for the Newton-Raphson solver and tested against the 1,000-pose benchmark.

The DNN-initialized pipeline dramatically outperformed the best-of-5 random baseline across all metrics:

- **Convergence Reliability:** The baseline random-initialization strategy successfully converged on 44.0% of the target poses. The DNN-initialized solver achieved a convergence rate of 77.4%, representing a 76% relative improvement in reliability.

- **Iteration Efficiency:** When the random baseline successfully converged, it required a median of 30 iterations. The DNN-initialized solver required a median of only 7 iterations.
- **Speedup:** On average, the learned warm-starting provided a $4.9\times$ speedup in solver iterations.

Crucially, the DNN inference step added only $\sim 0.5\,\mathrm{ms}$ of computational overhead. By effectively placing the initial guess deep inside the convergence basin of the local kinematic branch, the network bypassed the costly global search phase typically required by numerical solvers, allowing the Newton-Raphson algorithm to rapidly polish the solution to sub-millimeter accuracy.

VI. Discussion and Future Work

A. Performance Trade-offs and Industry Impact

The development of a hybrid IK solver requires balancing inference speed against predictive accuracy. Increasing the network's parameter count could theoretically map the kinematic manifold with higher fidelity, but the resulting computational bloat would negate the speed advantages of neural initialization. Conversely, reducing the dataset size degrades prediction accuracy, increasing the likelihood that the numerical solver fails to converge.

The proposed 535,820-parameter model trained on 10 million samples represents a highly effective sweet spot for a CPU-trained model. While the model achieves a $\sim 3^\circ$ joint-space error, the critical takeaway is that this accuracy is sufficient to consistently place the subsequent Newton-Raphson solver into a reliable convergence basin. The $\sim 0.5\,\mathrm{ms}$ neural inference overhead is practically negligible when it reliably eliminates dozens of expensive numerical iterations.

This efficiency is particularly crucial for the future of dynamic robotics. In traditional, repetitive manufacturing, IK is often pre-computed offline. However, as industry shifts toward autonomous manipulation, unstructured environments, and high-speed, non-repetitive tasks, robots must compute IK on the fly. In these tight, real-time control loops, saving milliseconds per solve while guaranteeing exact Cartesian precision offers a substantial operational advantage.

B. Limitations and Future Directions

While the current framework demonstrates a solid baseline improvement, several avenues remain for future exploration:

- **Hardware Acceleration and Data Scaling:** The current methodology was constrained to CPU-only training. Migrating the pipeline to a GPU-accelerated environment would permit training on datasets vastly exceeding 10 million samples. Given that the validation loss had not yet fully asymptoted, further scaling is likely to yield continued accuracy improvements.
- **Edge Computing Integration:** To push the inference latency well below the current $0.5\,\mathrm{ms}$ threshold, future deployments should investigate edge computing architectures. Executing the neural forward pass on dedicated hardware accelerators (such as local TPUs or FPGAs) could easily cut the inference time to the microsecond scale, ensuring strict compliance with high-frequency (e.g., $1\,\mathrm{kHz}$) industrial control loops.
- **Online Reinforcement Learning (RL):** The current model relies entirely on a supervised, offline synthetic dataset. A highly promising next step is to integrate a Reinforcement Learning framework. By treating the Newton-Raphson iteration count as a penalty signal, an RL agent could continuously fine-tune the network weights during real-world operation, allowing the robot to autonomously improve its initialization strategy and adapt to changing environments or hardware wear over time.

VII. Conclusion

This paper introduced a hybrid inverse kinematics framework for the UR5e manipulator that bridges the gap between the speed of deep learning and the absolute precision of classical numerical methods. By conditioning a Residual MLP on both the target end-effector pose and the robot's current joint state, the network successfully resolves the multi-modal nature of the IK problem, predicting highly informed joint initializations. Empirical results demonstrate that this learned warm-starting significantly outpaces random initialization strategies, dramatically improving both iteration efficiency and overall convergence reliability. As robotics moves increasingly toward dynamic, real-time autonomy, lightweight hybrid architectures offer a robust and highly scalable path forward.

References

[1] S. R. Buss, "Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods," *Unpublished Report*, 2009. [Online]. Available: http://math.ucsd.edu/~sbuss/ResearchWeb/ikmethods

[2] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics Tools*, vol. 10, no. 3, pp. 37–49, 2005.

[3] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 928–935.

[4] A. Calzada-Garcia, J. G. Victores, F. J. Naranjo-Campos, and C. Balaguer, "Inverse Kinematics for Robotic Manipulators via Deep Neural Networks: Experiments and Results," *Applied Sciences*, vol. 15, no. 13, p. 7226, 2025.