Distance-to-Target Dynamic Weighting for Robot Arm IK Networks

Rockwon Kim
ETRI
rwkim@etri.re.kr

Hyonyoung Han
ETRI
hyonyoung.han@etri.re.kr

Abstract—Inverse kinematics (IK) learning for robot arms involves multiple objectives such as minimizing end-effector position and orientation errors while respecting joint limits and avoiding collisions. These objectives often progress at different rates during training. In particular, position error tends to decrease relatively quickly, while orientation error improves more slowly, which can lead to imbalance when losses are weighted equally. We describe a Distance-to-Target Dynamic Weighting (DTD-W) approach that adjusts task weights according to their distance from predefined target losses, combined with an uncertainty-based weighting scheme. Experiments on a Panda 7-DOF robot arm indicate that this combined method provides a more balanced reduction of position and orientation errors compared to fixed or single weighting strategies.

Index Terms—inverse kinematics, multi-objective learning, loss weighting, uncertainty

I. Introduction

Training neural networks for robot arm IK often requires balancing heterogeneous objectives. In our setting, the network maps an end-effector pose (position and orientation) to corresponding joint angles. During training, we observe that the position loss decreases rapidly while the orientation loss remains relatively high. Without a mechanism to adapt the weights of these objectives, one loss may dominate, leaving the other insufficiently optimized.

Several methods have been proposed to balance multiple losses automatically. Approaches such as GradNorm [1] and Dynamic Weight Averaging (DWA) [2] adjust weights based on gradient magnitudes or relative changes in loss. Uncertainty-based weighting [3] normalizes losses according to predicted noise levels. While effective in certain settings, these methods do not explicitly incorporate target performance levels, which are often available in robotics tasks (e.g., desired error thresholds in millimeters or degrees).

We consider a Distance-to-Target (DTD) weighting strategy, which uses the normalized distance from each loss to its predefined target. This is combined with uncertainty-based weighting to account for task variability. The aim is to improve balance between position and orientation objectives in IK training.

II. Related Work

Dynamic weighting for multi-task learning has been extensively studied in computer vision and robotics. Chen et al. [1] proposed GradNorm, which balances gradient magnitudes to equalize task learning rates. Liu et al. [2] introduced Dynamic Weight Averaging based on temporal loss changes; see also EMA-based weighting strategies [4]. Multi-objective optimization approaches such as MGDA [5] focus on Pareto efficiency but do not directly encode task-specific thresholds. Our method differs in that it makes explicit use of target losses, while also retaining uncertainty modeling to adapt weights during training.

In the context of inverse kinematics (IK), neural networks have been widely explored as an alternative to analytical solvers. Cursi et al. [6] proposed a feedforward model that combines position and velocity errors under fixed weighting to improve end-effector trajectory prediction.

While these studies highlight the potential of neural approaches for IK, they do not explicitly address the issue of imbalanced loss terms or incorporate dynamic weighting strategies. Some models achieve reasonable performance without such mechanisms, yet challenges remain in balancing position and orientation accuracy, handling collision avoidance, and enforcing joint constraints. More recently, generative models have been proposed for IK [7], showing improved precision over conventional neural methods. This may be attributed to their inherent flexibility in handling multi-objective trade-offs. Motivated by this, we argue that introducing dynamic weighting into non-generative neural IK frameworks can offer a practical way to better manage competing objectives and further enhance performance.

III. Methodology

A. Problem Setup

We train a ResNet-based network (1024 units, 5 residual blocks) to map end-effector poses to joint angles for a Panda 7-DOF arm. The loss is defined over three objectives:

$$\mathcal{L} = \sum_{t \in \{\text{pos,ori,coll}\}} w_t \, \mathcal{L}_t,$$

where t denotes position, orientation, and collision penalties. Joint limit constraints are not treated as an explicit loss term. Instead, we adopt a joint-limit-satisfying output parameterization, in which the raw network outputs are

passed through a bounded activation (sigmoid or tanh) and linearly scaled to the range $[q_{\min}, q_{\max}]$. This guarantees that the predicted joint angles always remain within their valid ranges without requiring an additional penalty term.

B. Loss for Position and Orientation

For each predicted joint configuration $\hat{q}$, forward kinematics was used to obtain the end-effector position $\hat{p}$ and orientation $\hat{R}$. The position loss was defined as the mean squared error (MSE) between the predicted and target positions:

$$\mathcal{L}_{\text{pos}} = \|\hat{p} - p\|_2^2. \tag{1}$$

The orientation loss was defined using the geodesic distance on $SO(3)$ between the predicted and target orientations:

$$\mathcal{L}_{\text{ori}} = \arccos\left(\frac{\text{trace}(\hat{R}R^\top) - 1}{2}\right), \tag{2}$$

which measures the minimal rotation angle between two rotation matrices.

C. Collision Loss

For collision avoidance, we adopt a capsule-based approximation of robot links, a common practice in robotics for efficient distance computation. The collision loss is then defined using a smooth penalty function on the minimum capsule-to-capsule distance. We introduce a safety margin of 5 mm: if the distance exceeds this margin, the loss becomes nearly zero, whereas distances below the margin incur increasing penalties. To ensure differentiability, we use the softplus function as a smooth approximation of the hinge loss:

$$\mathcal{L}_{\text{coll}} = \tau \cdot \log\big(1 + \exp\big(\tfrac{m-d}{\tau}\big)\big), \tag{3}$$

where d is the minimum distance, m is the safety margin (5 mm), and τ controls smoothness.

D. Distance-to-Target (DTD)

Let $\varepsilon > 0$ be a small constant. For each task $t \in \{\text{pos}, \text{ori}, \text{coll}\}$ with target L_t^{target}, we define $d_t^{(\cdot)}$ as the distance-to-target, α_t as the scale factor, and $w_t^{(e)}$ as the dynamic weight:

$$\begin{aligned} d_t^{(0)} &= \frac{L_t^{(0)}}{L_t^{\text{target}} + \varepsilon}, \\ \alpha_t &= 100 \cdot \frac{d_t^{(0)}}{\sum_t d_t^{(0)}}, \\ w_t^{(e)} &= \frac{d_t^{(e)}}{\sum_t d_t^{(e)}}. \end{aligned} \tag{4}$$

Algorithm 1 Training with DTD + Uncertainty Weighting

Require: targets L_t^{target}, smoothing τ, margin m, scales α_t, small ε

1: **for** epoch $e = 1..E$ **do**
2: Sample batch (x, q); predict $\hat{q}$; apply bounded scaling to $[q_{\min}, q_{\max}]$
3: FK: $(\hat{p}, \hat{R})$ from $\hat{q}$; compute $L_{\text{pos}}, L_{\text{ori}}$; compute L_{coll} via capsule distances with softplus, margin m
4: $d_t \leftarrow \frac{L_t}{T_t + \varepsilon}$; $\quad w_t \leftarrow \frac{d_t}{\sum_t d_t}$
5: $\mathcal{L} \leftarrow \sum_t \alpha_t \, w_t \, \frac{1}{2\sigma_t^2} \, L_t \; + \; \log \sigma_t$
6: Update network parameters and $\{\sigma_t\}$ with AdamW
7: **end for**

E. Uncertainty-Weighted and Hybrid Losses

We compare two loss formulations: (i) an uncertainty-weighted loss and (ii) a hybrid loss that additionally applies DTD-based dynamic weights.

$$\mathcal{L}_{\text{unc}}^{(e)} = \sum_t \Big(\alpha_t \, \tfrac{1}{2\sigma_t^2} \, L_t^{(e)} + \log \sigma_t \Big), \tag{5}$$

$$\mathcal{L}_{\text{hybrid}}^{(e)} = \sum_t \Big[\alpha_t \, w_t^{(e)} \, \tfrac{1}{2\sigma_t^2} \, L_t^{(e)} + \log \sigma_t \Big]. \tag{6}$$

Here, α_t is the fixed scale factor and $w_t^{(e)}$ is the dynamic weight; both apply only to the data term, while $\log \sigma_t$ is left unweighted. and $\sigma_t > 0$ is the learned task uncertainty (we optimize $s_t = \log \sigma_t$ for stability). Note that α_t and $w_t^{(e)}$ are applied only to the data term, while the regularizer $\log \sigma_t$ is left unweighted.

IV. Experiments

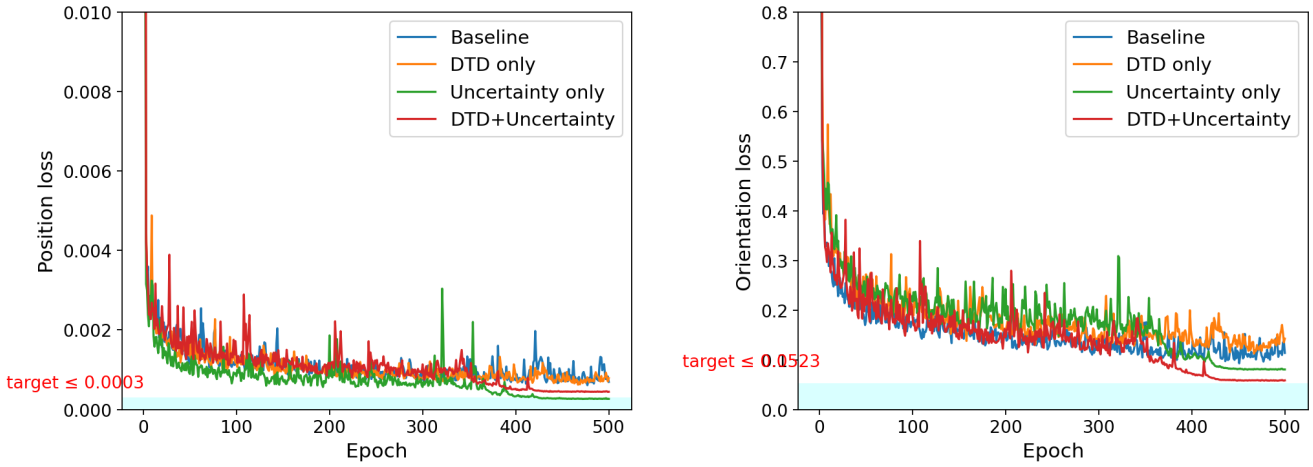
A. Dataset

The dataset was generated by uniformly sampling valid configurations of the Panda 7-DOF robot. Each joint range was divided into 16 intervals, and all possible combinations were taken to ensure broad coverage of the configuration space. Forward kinematics was then applied to compute the corresponding end-effector poses, resulting in a total of 32,462,548 samples. Each input is represented as a 7-dimensional end-effector pose vector

$$x = [\mathbf{p}, \mathbf{q}] \in \mathbb{R}^7,$$

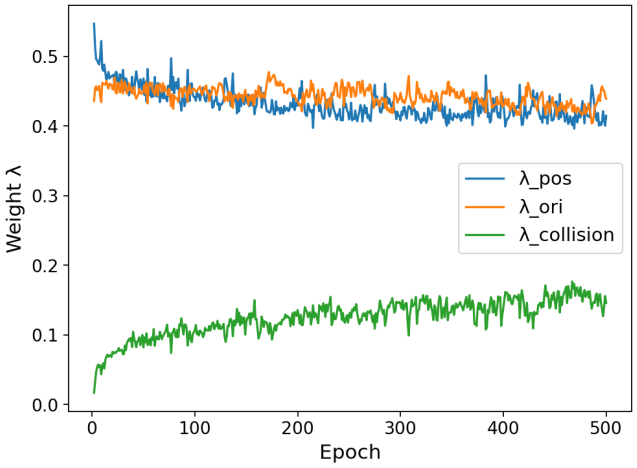
where $\mathbf{p} \in \mathbb{R}^3$ denotes the Cartesian position and $\mathbf{q} \in \mathbb{R}^4$ is the unit quaternion encoding orientation. The corresponding joint angles obtained from forward kinematics serve as training targets.

Note on dataset size. A full 16^7 grid would yield 268,435,456 joint combinations. We remove infeasible/duplicate configurations and apply stratified sampling under self-collision filtering, resulting in 32,462,548 valid samples used in our experiments.

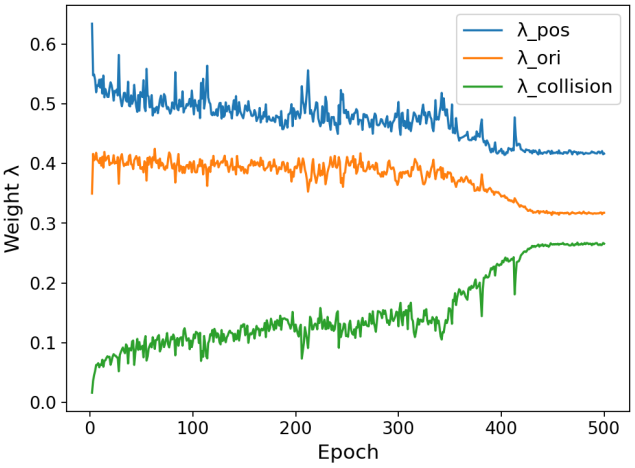


(a) Position loss convergence

(b) Orientation loss convergence



(c) Weight evolution (DTD only)



(d) Weight evolution (DTD + Uncertainty)

Fig. 1. Training results on Panda IK dataset. (a)(b) show the convergence of position and orientation losses across four settings (Baseline, DTD only, Uncertainty only, DTD+Uncertainty). (c)(d) illustrate the weight dynamics of position, orientation, and collision terms under different weighting strategies.

B. Training Setup

Each training epoch consists of 2,000,000 randomly sampled data points, partitioned into 160,000 for training, 20,000 for validation, and 20,000 for testing. The network is trained for 500 epochs in total.

The initial target losses and learning-rate settings are not designed to immediately reach sub-millimeter precision. Instead, they are intended to (i) prevent excessive disparity among the position, orientation, and collision objectives, and (ii) promote rapid convergence to a reasonable operating regime. Accordingly, in Stage 1 we set

$$\mathcal{L}^{(1)}_{\text{target}} = \{\text{pos} : [0.00009 \sim 0.0003],\ \text{ori} : 0.0525,\ \text{coll} : 0.009\},$$

where the position loss target is chosen within the range 0.00009 to 0.0003 depending on the dataset statistics and convergence behavior. The network learning rate is initialized at $\text{lr} = 1 \times 10^{-4}$, while the uncertainty parameters ($s_t = \log \sigma_t$) are updated with a reduced rate of $\text{lr} \times 0.1$.

Once convergence is observed, we proceed to Stage 2 for fine-tuning with tighter objectives:

$$\mathcal{L}^{(2)}_{\text{target}} = \{\text{pos} : 0.000025,\ \text{ori} : 0.0175,\ \text{coll} : 0.009\}.$$

In this stage, the network learning rate is reduced to $\text{lr} = 1.2 \times 10^{-5}$, and the uncertainty parameters are trained under the same rule, $\text{lr} \times 0.1 = 1.2 \times 10^{-6}$. This two-stage procedure allows the network to first stabilize in a broad feasible regime and then achieve higher precision through gradual refinement.

C. Compared Methods

We compare four weighting strategies:

- Baseline: equal weighting without DTD or Uncertainty
- DTD only: distance-to-target dynamic weighting
- Uncertainty only: uncertainty-based weighting
- DTD + Uncertainty: hybrid combination of both methods

D. Results

Table I lists the best (minimum) losses attained during training. The Baseline shows the largest position MSE (0.000852 mm^2) and a relatively high orientation error (0.12 rad). DTD only reduces the position MSE (0.000764 mm^2) but slightly worsens orientation (0.13 rad). Uncertainty only achieves the lowest position MSE (0.000268 mm^2) with an orientation error of 0.08 rad. The DTD+Uncertainty hybrid attains the best orientation accuracy (0.06 rad) while keeping position MSE low (0.000447 mm^2), yielding the most balanced overall performance.

TABLE I
Minimum Position MSE (mm^2) and Orientation Error (rad) Across Weighting Strategies on Stage 1

Method	Pos. MSE (mm^2)	Ori. Error (rad)
Baseline	0.000852	0.12
DTD only	0.000764	0.13
Uncertainty only	0.000268	0.08
DTD + Uncertainty	0.000447	0.06

Figure 1 illustrates the loss convergence patterns and weight dynamics during Panda IK training. Panels (a) and (b) show the learning curves of position and orientation errors. In the Baseline, the position error decreases rapidly, whereas the orientation error converges more slowly. When applying DTD only, the position error improves but the orientation error becomes less stable. In contrast, with Uncertainty-based weighting alone, the orientation error decreases more steadily. When DTD and Uncertainty are combined, both errors converge below a certain threshold in a balanced manner. This suggests that if the reduction of losses is skewed toward one objective, the other may fail to decrease sufficiently, highlighting the need for balanced convergence.

Panels (c) and (d) compare the dynamics of task weights. With DTD only (c), the weights are determined solely by the relative size of the remaining distance to the target loss. As a result, the ratio between position and orientation weights remains nearly constant, consistent with the weighting rule in Eq. (6). Although such a scheme may appear to encourage balanced reduction of position and orientation errors, the actual model behavior does not reflect this; in practice, the weighting strategy does not directly translate into proportional error reduction, which prevents full attainment of the target losses. The gradual increase in the collision weight is explained by the faster reduction of position and orientation losses, which shifts weight toward the relatively slower-decaying term.

Given that the Stage 1 results (Table I) show the hybrid (DTD+Uncertainty) scheme yields the best overall balance, we adopt the same hybrid loss for Stage 2. Starting from the Stage 1 model, we fine-tune with the tighter Stage 2 targets and reduced learning rates described earlier. We then evaluate by predicting joint angles for 1,000 valid end-effector goals, repeated 100 times. The aggregated results (mean±std over 100 runs) are summarized in Table II.

TABLE II
Stage-2 Hybrid (DTD+Uncertainty) Results over $100 \times 1{,}000$ Evaluations

Metric	Mean	Std
Pos. MSE (mm^2)	0.000033 (5.73 mm)	0.000005 (2.25 mm)
Ori. Error (rad)	0.063 (3.61 deg)	0.0015 (0.86 deg)
Inference time (sec / query)	0.0038 (0.0013 sec)	0.0013 (0.0013 sec)

V. Discussion

Our experimental findings demonstrate that the combination of the Distance-to-Target (DTD) strategy with uncertainty weighting provides a method for balancing heterogeneous IK objectives. We conducted a two-stage training process. Initially, we used intentionally loose loss targets to prevent large inter-task disparities and facilitate rapid convergence to a reasonable solution space. The subsequent Stage 2 involved fine-tuning with tighter targets and reduced learning rates. This allowed the model to further improve in a stable manner, resulting in mean errors of a few millimeters (position) and a few degrees (orientation). This enhanced performance came with a low inference latency of a few milliseconds.

These results confirm the core intuition behind DTD: by directing more weight toward underperforming objectives, DTD focuses optimization where it is most needed, while uncertainty weighting suppresses noise and instability, making the overall training process more stable. This stability is especially critical during fine-tuning, where the learning rate is reduced and even small updates can destabilize learning.

This approach provides a transparent and intuitive way to manage task priorities by setting physically meaningful targets (e.g., in millimeters or degrees). Our results suggest that the hybrid scheme is not overly sensitive to the exact target values within a reasonable range, although extreme values may still disrupt the balance.

However, our study has several limitations. First, the training and evaluation were based on poses generated by forward kinematics with a simplified capsule-based collision approximation, without modeling real-world actuation dynamics or sensor noise. Second, while we compared our method against a baseline and single-weighting strategies (DTD and Uncertainty only), we did not include other multi-objective optimization methods such as MGDA or gradient-balancing variants. Such comparisons would be informative but would have added considerable complexity. Finally, collision was treated as a simple geometric proximity objective, and richer constraints like

environmental contact affordances were not considered. These areas represent promising directions for future research.

VI. Conclusion

We presented a hybrid loss weighting method for IK learning that combines Distance-to-Target and uncertainty-based weighting. The approach is simple to implement, interpretable (targets in mm/deg), and effective at balancing position, orientation, and collision objectives. On a Panda 7-DOF benchmark, the hybrid method achieved the best overall trade-off on Stage 1 and further improved under Stage 2 fine-tuning with tighter targets and reduced learning rates, reaching low millimeter-scale position errors and few-degree orientation errors at millisecond inference times.

Future work includes: (i) automatic or curriculum-based target scheduling, (ii) broader comparisons with Pareto-front multi-objective methods, (iii) real-robot validation under sensing and actuation noise, and (iv) extension to trajectory-level objectives (velocity/acceleration limits and smoothness). We believe data-driven IK serves as a crucial link, translating high-level perceptual information like images and text into the low-level joint movements that control a robot's physical actions.

Acknowledgment

This work was supported by the Ministry of Trade, Industry and Energy (MOTIE), Korea, under the Global Industrial Technology Cooperation Center (GITCC) Program, supervised by the Korea Institute for Advancement of Technology (KIAT), under Task No. P0028420.

References

[1] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in Proceedings of the 35th International Conference on Machine Learning (ICML), ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 794–803.

[2] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1871–1880.

[3] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7482–7491.

[4] A. Lakkapragada, E. Sleiman, S. Surabhi, and D. P. Wall, "Mitigating negative transfer in multi-task learning with exponential moving average loss weighting strategies (student abstract)," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 13, 2023, pp. 16 246–16 247.

[5] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 525–536.

[6] F. Cursi, F. Carrabs, M. Nappi, and V. Riccio, "Augmenting loss functions of feedforward neural networks with differential relationships for robot kinematic modelling," in International Conference on Advanced Robotics (ICAR), 2021.

[7] B. Ames, J. Morgan, and G. Konidaris, "IKFlow: Generating diverse inverse kinematics solutions," IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 7177–7184, 2022.