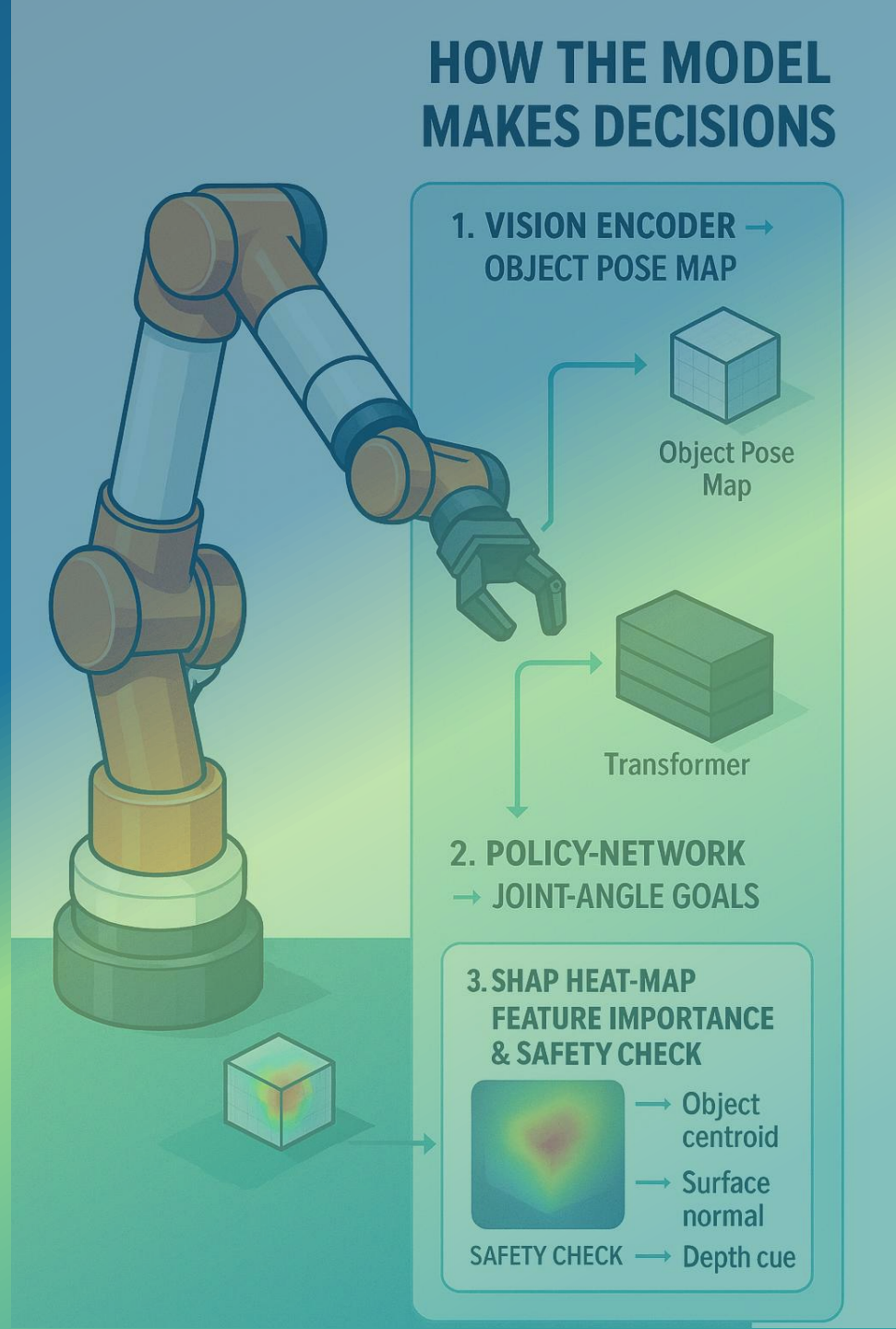
Thesis Defense

Explainable Neural Inverse Kinematics for Obstacle-Aware Robotic Manipulation: A Comparative Analysis of IKNet Variants

June 2025

Student: Sheng-Kai Chen 陳嵊凱
Advisor: Dr. Po-Chiang Lin 林柏江博士



Contents

Current Challenges

There are some challenges and regulations of IK and AI.

Inverse Kinematic

- **Problem**

  Traditional closed-form or numerical solvers handle moderate degrees of freedom, but their runtime grows with redundant joints, and they often require manual damping or iterative back-tracking to respect joint limits.

- **Issue**

  There are some IK Deep Learning models that can deal with the problems, but the models are lack with transparency.

AI

- **Need**

  Now the Deep Learning models need to be more trust-worthy, and it can be application in robotics.

- **Regulations**

  Now there are many regulations of using AI, e.g. EU AI Act

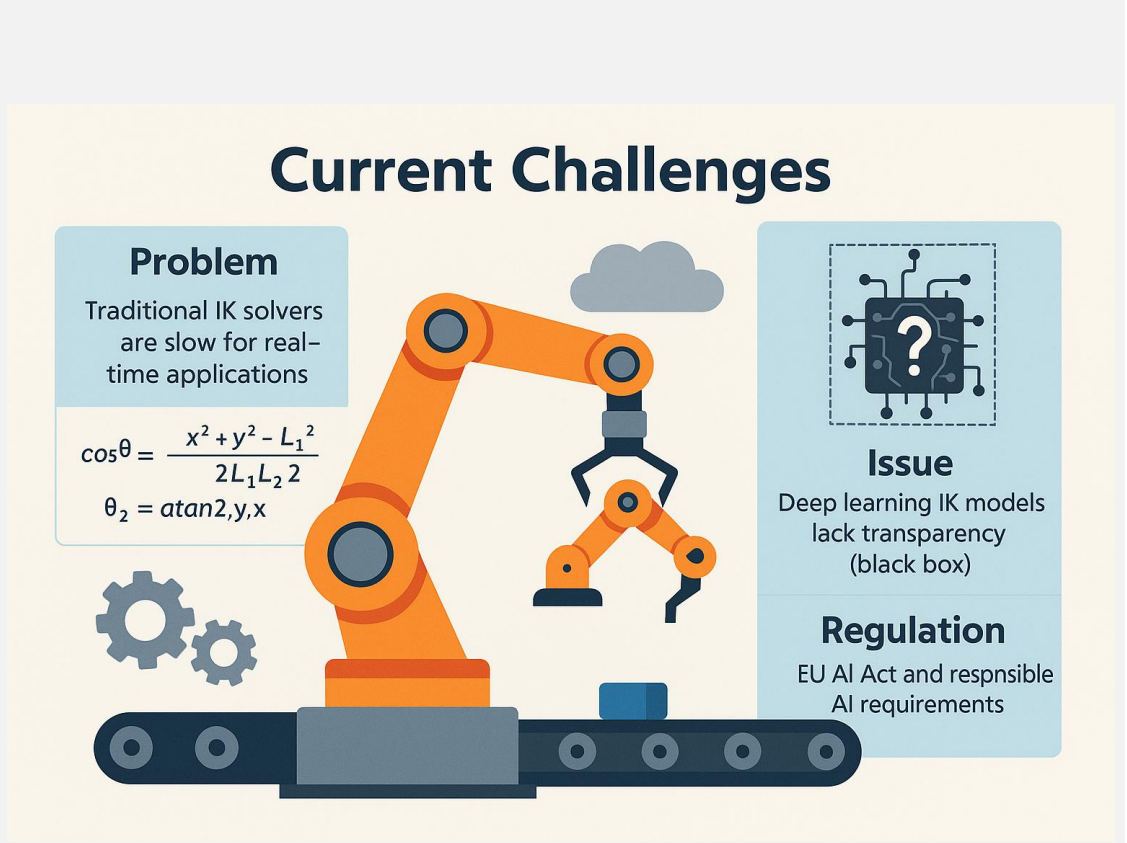


Fig 1. Current Challenges

Research Gap

Find out what is the main purpose in the research.

Comparison

Approach	Speed	Accuracy	Explainability	Obstacle Aware
Original IK	Slow	Good	Transparent	Limited
Neural IK	Fast	Good	Black Box	Unknown

Gap

- **3 No**

  No Explainability in neural inverse kinematic.

  No Connection between AI decisions and physical safety.

  No Tools to understand obstacle avoidance behavior.

Solution

Combine neural IK with explainable AI (XAI) to understand and improve robot decision-making

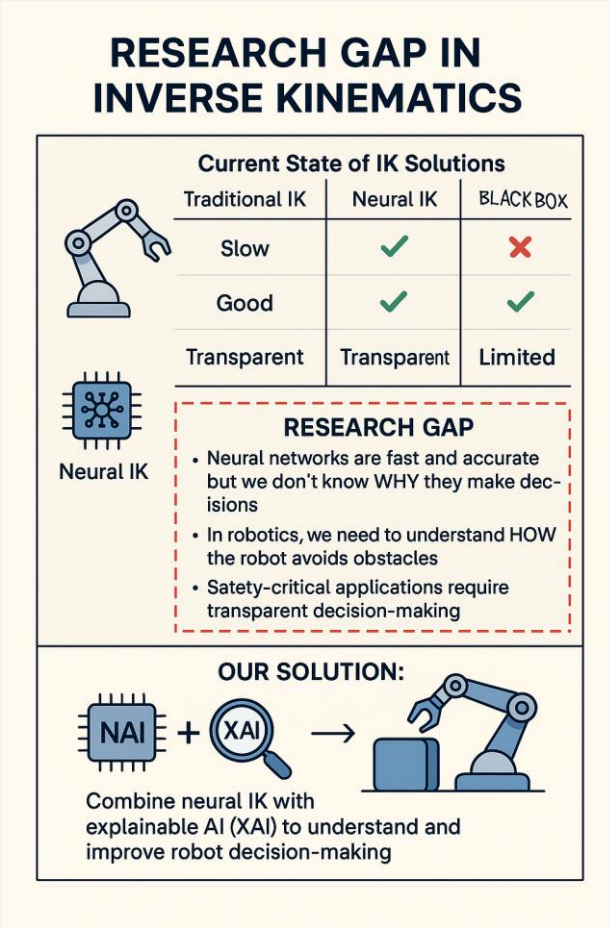


Fig 2. Research Gap

Research Questions

Questions need to be solved through the research.

Research Question 01

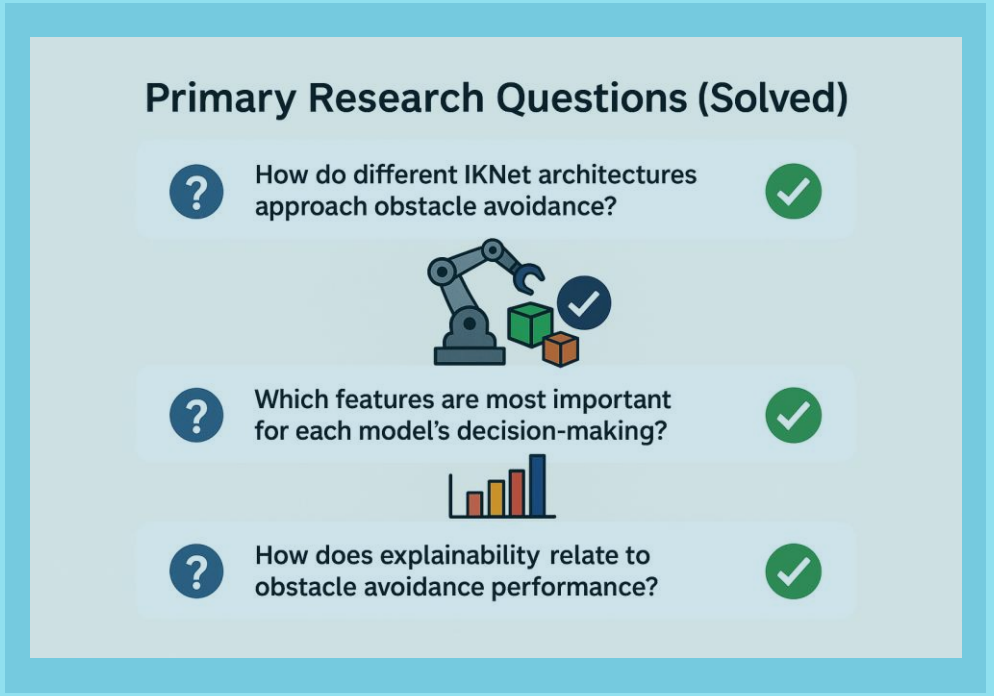
How do different IKNet architectures approach obstacle avoidance?

Research Question 02

Which features are most important for each model's decision-making?

Research Question 03

How does explainability relate to obstacle avoidance performance?



Contributions

Showing why this research is important to the field.

XAI-driven Analysis Framework for Neural IK

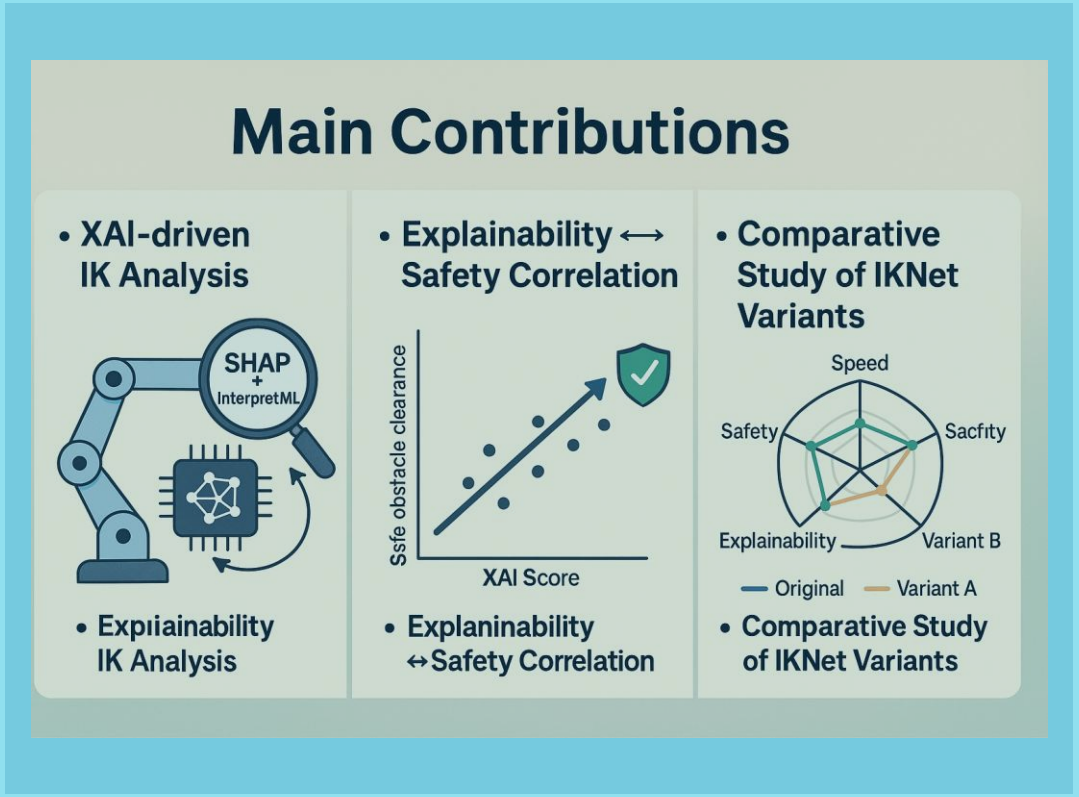
Comprehensive explainability with XAI tool and link feature attribution to physical robot behavior

Explainability-Safety Correlation

Connecting AI explanations to obstacle avoidance performance and demonstrates how balanced feature attribution leads to better safety

Comprehensive Comparative Analysis

Systematic evaluation of three IKNet variants, multi-scenario obstacle avoidance assessment and integrate XAI insights with physical safety metrics



Learning-based Inverse Kinematics

Get ideas to do the inverse kinematics.

Traditional IK

- **Jacobian pseudo-inverse**

  It solves for minimum-norm joint increments but oscillates near singularities unless damped.

- **CCD**

  The method converges quickly for serial chains yet produces zig-zag trajectories when the target is distant.

- **DLS**

  It trades accuracy for robustness by injecting a Tikhonov regulariser.

Neural IK

- **IKNet and DeepIK**

  Two of them paved the way for data-driven IK, obtaining MSE < on the KUKA LBR dataset (75 k samples) with five × 400-unit layers.

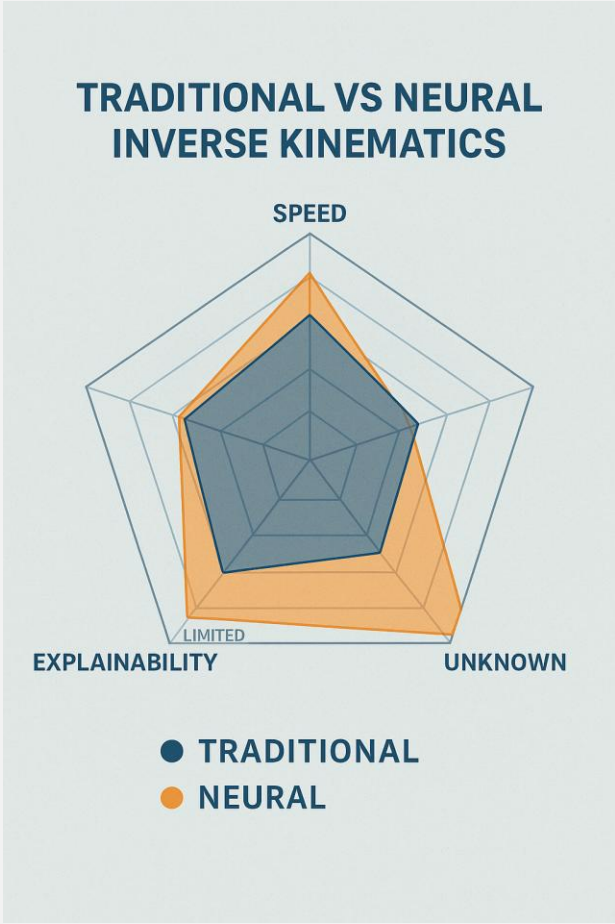


Fig 3. Inverse Kinematics

Explainable AI (XAI)

Get tools to start with XAI.

Model Analysis XAI

- **SHAP**

  SHAP offers local accuracy and consistency, requiring evaluations in the worst case but approximate SHAP scales linearly with samples.

- **LIME**

  LIME perturbs inputs to fit local ridge models and applicability to high-dimensional quaternion spaces is limited.

Comprehensive XAI

- **InterpretML**

  It allow interactive attribution dashboards with over 5 ms overhead via WebSocket' s and provide both glassbox and blackbox models analysis.

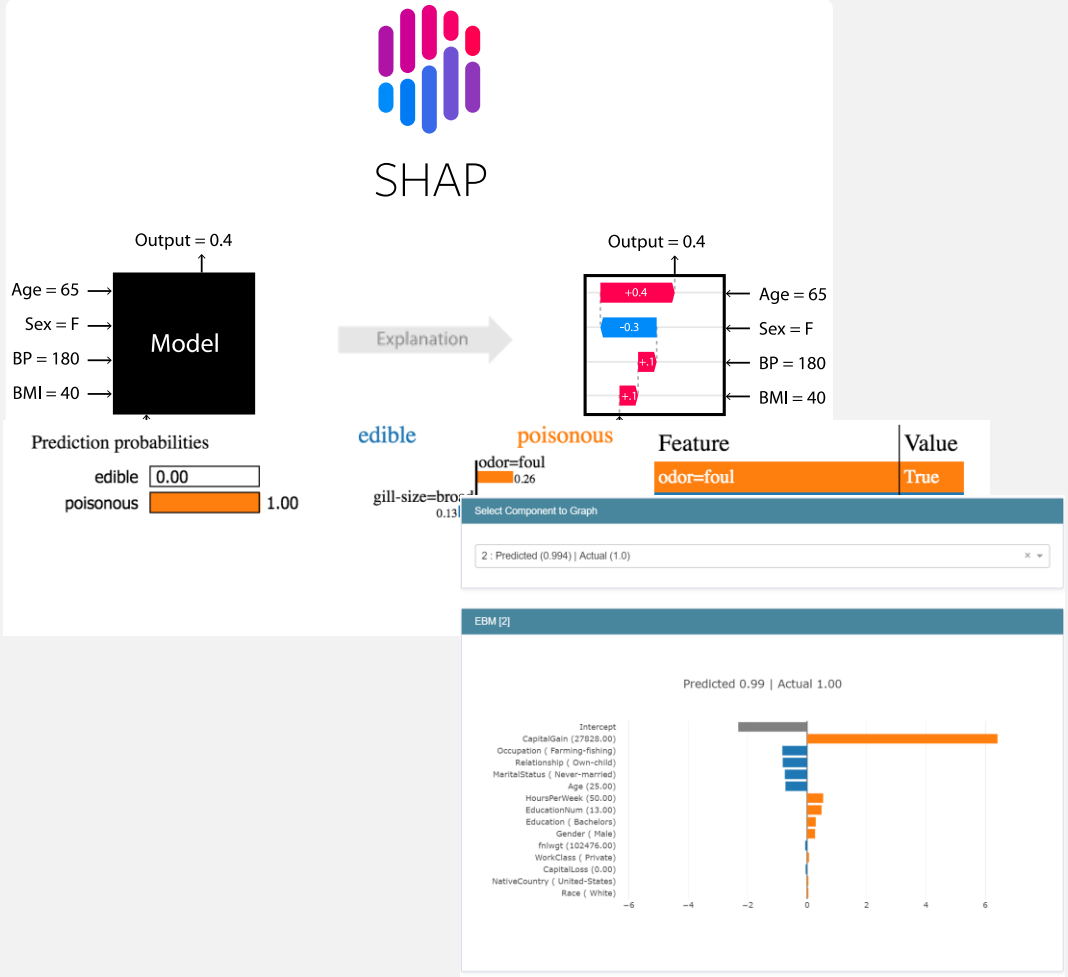


Fig 4. XAI Tools

Full Structure

The whole structure of the research.

IKNet Models

Use three original and improved IKnet models to train and using it for inverse kinematics also do the comparison.

XAI Analysis

Use two XAI tools, SHAP and InterpretML, to analysis the three models. Also, combined the results and show by feature importance.

Obstacle Avoidance

Create a visualization of obstacles and using the results of XAI analysis to see how the model done the decisions to avoid the obstacles.

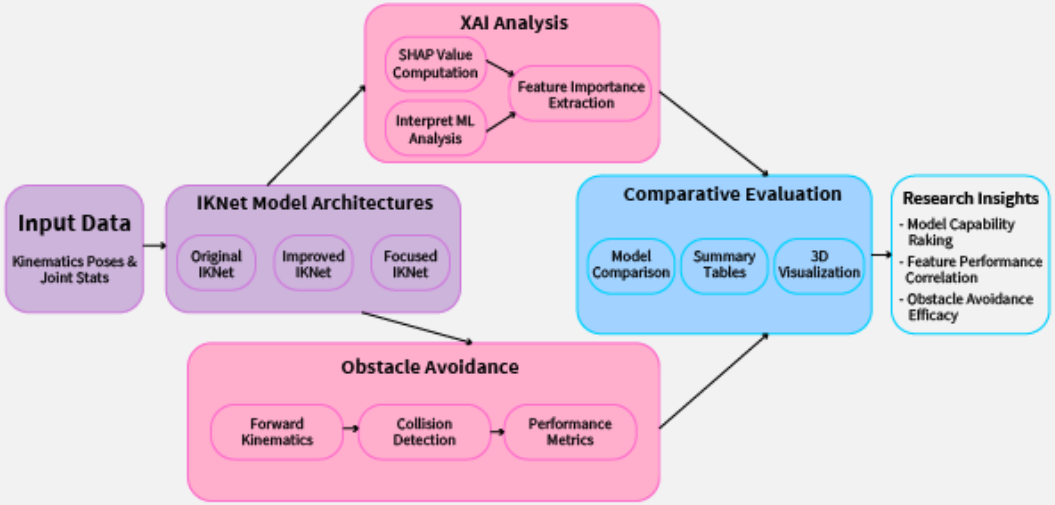


Fig 5. Full Structure of Research

IKNet Models

The structure of each models.

Original IKNet

- Input(7-Dimension)
- Hidden Layers (neurons)

  400 -> 300 -> 200 -> 100 -> 50 (ReLU)

  Dropout(0.1)
- Output(4-Dimension)
- Sequential feature processing

Improved IKNet

- Input(7-Dimension)
- Hidden Layers (neurons)

  128 -> 64 (ReLU) with BatchNormalization -> ResidualBlock

  Dropout(0.1)
- Output(4-Dimension)
- Enhanced feature propagation

Feature	Original IKNet	Improved IKNet	Focused IKNet
Architecture	Sequentially fully connected layers with decreasing dimensions.	Residual blocks with batch normalization	Separate branches for position and orientation
Input Processing	Unified processing of position and orientation	Unified processing with enhanced feature propagation	Specialized processing paths for position and orientation
Hidden Dimensions	[400, 300, 200, 100, 50]	[128, 64]	64 for each branch, 128 combined
Activation Function	ReLU	ReLU	ReLU
Regularization	Dropout (p=0.1)	Dropout (p=0.1) + Batch Normalization	Dropout (p=0.05)
Weight Initialization	Default PyTorch	Kaiming	Kaiming
Key Features	Simple, direct mapping	Residual connections, gradient flow enhancement	Explicit separation of position and orientation components
Design Philosophy	Gradually decreasing dimensionality	Enhanced feature propagation	Specialized feature extraction

Fig 6. Compare Table of three models

IKNet Models

The structure of each models.

Focused IKNet

- Input(7-Dimension)
- Hidden Layers (neurons)
  64 -> 128 (ReLU + Dropout) (3-Dimension)
  64 -> 128 (ReLU + Dropout) (4-Dimension)
  128 -> 64 -> 4 (ReLU + Dropout)
- Output(4-Dimension)
- Explicit separation of concerns

Loss Function

- Input: 7D pose vector (x,y,z + quaternion)
- Output: 4D joint angles
- Optimizer: Adam (lr=1e-3)
- Loss: Position + Orientation + Joint limits

$$L = w_{pos} \cdot E_{pos} + w_{orient} \cdot E_{orient} + w_{limit} \cdot \sum_{i=1}^{n} max(0, |\theta_{i,max}|)^2$$

Fig 7. Loss Function Formula

XAI Analysis

The structure of SHAP Analysis.

SHAP Implementation

- **Input: 7D Pose Vector [x, y, z, qx, qy, qz, qw]**
- **IKNet Model Prediction**
  - Original IKNet
  - Improved IKNet
  - Focused IKNet
- **SHAP Value Computation**
- **Feature Importance for each joint [joint1, joint2, joint3, joint4]**

Key SHAP Properties

- **Efficiency:** All feature contributions sum to prediction
- **Symmetry:** Equal features get equal attribution
- **Dummy:** Irrelevant features get zero attribution
- **Additivity:** Consistent across different models

Algorithm 1 SHAP Analysis for IKNet Models

Data: model, background_data, test_data, feature_names
Result: SHAP values and expected values
Create CPU copy of model for analysis
Define prediction function for model outputs
Initialize KernelExplainer with prediction function
Compute SHAP values for test data
return *SHAP values, expected values*

Fig 8. SHAP Algorithm

XAI Analysis

The structure of InterpretML Analysis.

SHAP Implementation

- **Step 01: Custom Feature Importance**
  Permutation-based analysis
  Generate importance distributions per joint
- **Step 02: Partial Dependence Analysis**
  Replace feature values with grid points
  Calculate average predictions and plot relationships
- **Step 03: Feature Interaction Analysis**
  Measure joint effects beyond individual contributions
  Generate interaction strength matrices

Formulas

- **Importance Score**

$$I_i = \mathbb{E}[|f(X) - f(X^i)|] \tag{7}$$

- **Partial Dependence**

$$\mathrm{PD}_i(x_i) = \mathbb{E}_{x_C}[f(x_i, x_C)] \tag{8}$$

Algorithm 5 InterpretML Analysis

Require: model, dataset, feature_names, joint_names, output_dir
Ensure: Dictionary of interpretability results

```
Sample data for analysis (up to 200 samples)
Create predict function for model
// Custom Feature Importance Analysis
for each joint in joint_names do
    for each feature in feature_names do
        Create perturbed dataset with shuffled feature values
        Calculate importance as mean absolute prediction difference
        Visualize feature importance
    end for
end for
// Manual Partial Dependence Analysis
for each joint in joint_names do
    for each feature in feature_names do
        Create grid spanning feature range
        for each grid point do
            Replace feature values and calculate average prediction
            Store in partial dependence values
        end for
        Generate partial dependence plot
    end for
end for
// Feature Interaction Analysis
for each joint in joint_names do
    Identify top features by importance
    for each pair of top features do
        Create 2D grid for feature combinations
        Calculate and visualize interaction effects
    end for
end for
Create consolidated visualization of all analyses
return interpretability results =0
```

Fig 9. InterpretML Algorithm

XAI Analysis

Visualization steps and key points.

Visualization Components

- **SHAP Bar Charts**
  Direct feature importance comparison
- **Heat Maps**
  Feature sensitivity across input space
- **Partial Dependence Plots**
  How features affect outputs
- **Feature Interaction Maps**
  2D relationship visualization

Output

- **Feature importance rankings per model per joint**
- **Sensitivity patterns across feature space**
- **Non-linear relationship identification**
- **Model comparison insights**

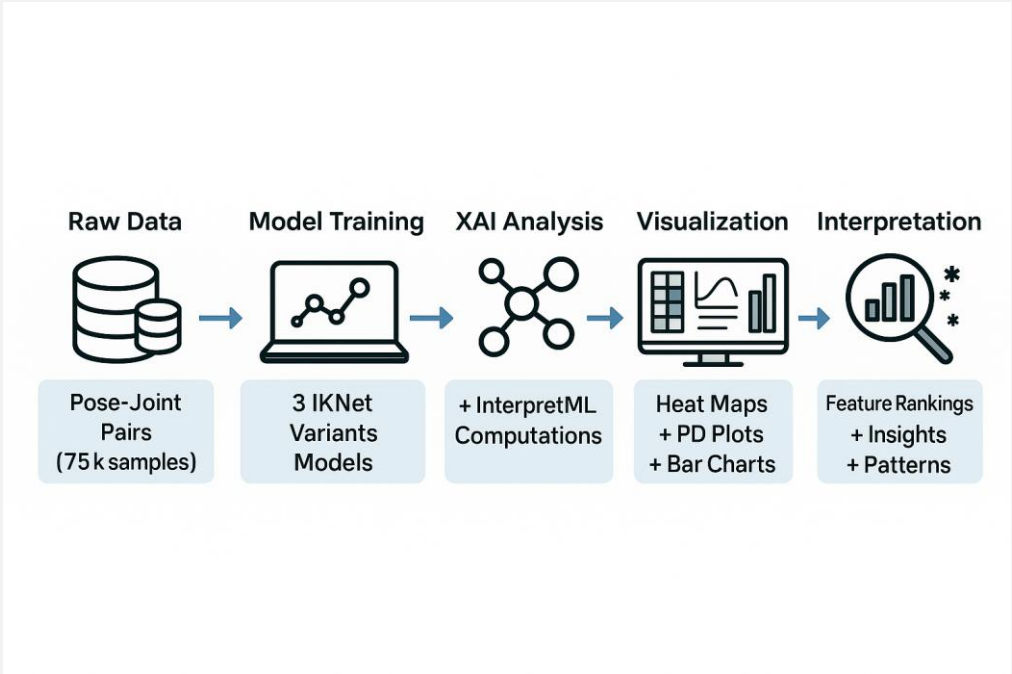


Fig 10. Visualization Steps

Obstacle Avoidance

The whole structure of obstacle avoidance.

Forward Kinematics Implementation

$$\begin{cases} \theta_{cum,i} = \sum_{j=1}^{i} \theta_j \\ x_i = x_{i-1} + l_i \cos(\theta_{cum,i}) \\ y_i = y_{i-1} + l_i \sin(\theta_{cum,i}) \end{cases} \tag{10}$$

$$\begin{cases} \theta_{cum,i} = \sum_{j=1}^{i} \theta_j \\ x_i = x_{i-1} + l_i \cos(\theta_{cum,i}) \\ y_i = y_{i-1} + l_i \sin(\theta_{cum,i}) \\ z_i = z_{i-1} + l_i \cdot 0.1 \cdot \frac{i}{n} \end{cases} \tag{11}$$

Collision Detection

$$\begin{cases} \mathbf{v} = \mathbf{x}_2 - \mathbf{x}_1 \quad \text{(segment vector)} \\ t = \frac{(\mathbf{o}-\mathbf{x}_1)\cdot\mathbf{v}}{|\mathbf{v}|^2} \quad \text{(projection parameter)} \\ t' = \max(0, \min(1, t)) \quad \text{(constrained parameter)} \\ \mathbf{p} = \mathbf{x}_1 + t'\mathbf{v} \quad \text{(closest point)} \end{cases} \tag{12}$$

Algorithm 6 Forward Kinematics

Require: joint_angles, link_lengths, add_z
Ensure: Joint positions

```
positions = add_z ? [(0,0,0)] : [(0,0)]
cumulative_angle = 0
for i, angle in enumerate(joint_angles) do
    cumulative_angle += angle
    prev_pos = positions[-1]
    x = prev_pos[0] + link_lengths[i] * cos(cumulative_angle)
    y = prev_pos[1] + link_lengths[i] * sin(cumulative_angle)
    if add_z then
        z = prev_pos[2] + link_lengths[i] * 0.1 * (i+1)/len(link_lengths)
        positions.append((x,y,z))
    else
        positions.append((x,y))
    end if
end for
return positions =0
```

Algorithm 7 Collision Detection

Require: arm_positions, obstacle_positions, obstacle_radii
Ensure: Collision status, clearance, critical segment

```
min_clearance = infinity
collision_detected = false
collision_segment = null
collision_obstacle = null
for obs_idx, (obs_pos, obs_radius) in enumerate(obstacles) do
    for segment_idx in range(len(arm_positions) - 1) do
        Calculate line vector and length from segment endpoints
        if line_len ¿ 0 then
            Calculate closest point on segment to obstacle
            distance = distance between closest point and obstacle center
            clearance = distance - obs_radius
            if clearance ¡ min_clearance then
                min_clearance = clearance
                collision_segment = segment_idx
                collision_obstacle = obs_idx
            end if
            if distance ¡= obs_radius then
                collision_detected = true
            end if
        end if
    end for
end for
return collision_detected, min_clearance, collision_segment,
collision_obstacle =0
```

Fig 11. Avoidance Algorithm

Evaluation Metrics

Visualization steps and key points.

Primary Safety Metrics

- **Minimum Clearance:** Smallest distance to any obstacle
  Higher = Safer (Measure in workspace units)
- **Target Position Error:** Euclidean distance to goal
  Lower = More accurate
- **Collision Rate:** Percentage of failed attempts
  0% = Perfect safety record
- **Critical Segment Analysis:** Which arm part is most vulnerable
  Identifies weak points in obstacle avoidance strategy

Secondary Performance Metrics

- **Path Smoothness:** Trajectory continuity
- **Energy Efficiency:** Path directness
- **Computational Time:** Inference speed
- **Robustness:** Performance across scenarios

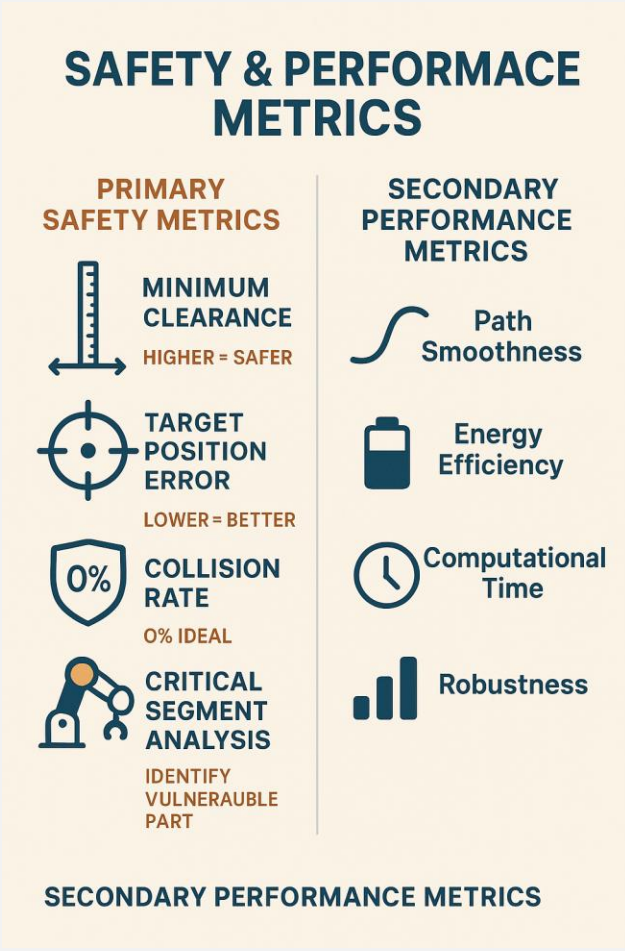


Fig 12. Evaluation Metrics

Scenario Generation

Visualize and test in different scenarios.

Scenario Complexity Levels

- **Simple**
  2-3 obstacles, wide spacing
- **Moderate**
  3-4 obstacles, narrow passages
- **Complex**
  4-5 obstacles, constrained workspace

Data Collection

- **Trajectory Recording:** Complete arm configurations
- **Performance Logging:** All metrics per scenario
- **Visualization Generation:** 2D and 3D path plots
- **Statistical Compilation:** Cross-model comparisons

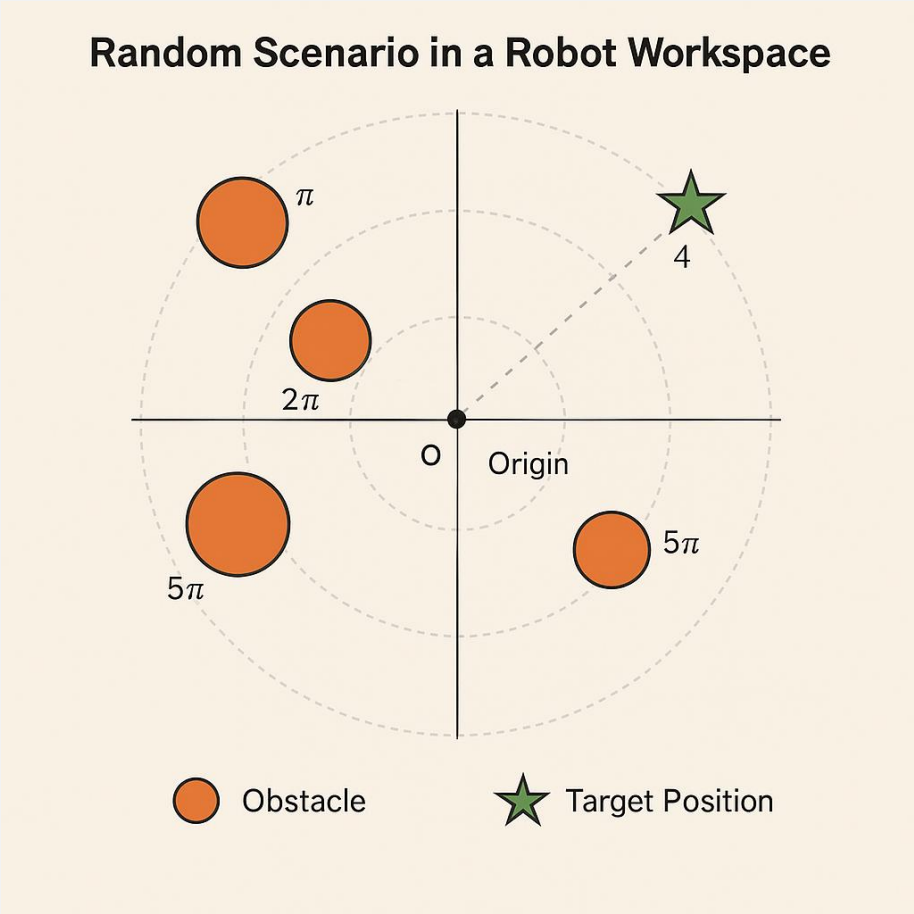


Fig 13. Scenario Generation

SHAP Results

The overview performance of SHAP Analysis.

Key Patterns Identified

- **Original IKNet**
  Heavy reliance on quaternion components (qz=0.6)
- **Improved IKNet**
  More balanced distribution across all features
- **Focused IKNet**
  Strong emphasis on z-position (0.8) and qy (0.6)

Significance

- **All differences > 0.1 are statistically significant ($p < 0.05$)**
- **Cross-validation confirms pattern consistency**
- **Standard deviations within acceptable ranges**

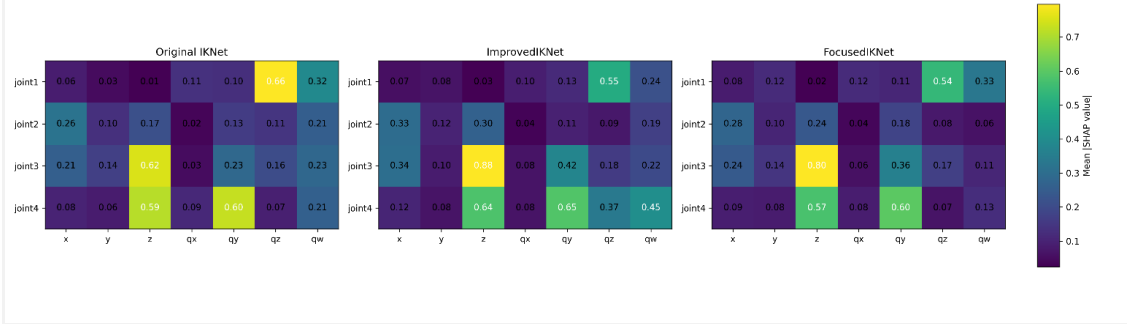


Fig 14. SHAP Heat Map

SHAP Results

Overview on Original IKNet results analysis with SHAP.

Result Breakdown

- **Joint 01**

  qz shows highest relative importance, minimal position influence
- **Joint 02**

  qw primary influence, secondary qz contribution
- **Joint 03**

  qz shows extremely high relative importance
- **Joint 04**

  qz continues to dominate

Interpretation and Meaning

- **Strategy:** Orientation-centric obstacle avoidance
- **Strength:** Consistent rotational approach
- **Weakness:** Underutilizes positional information
- **Implication**: May miss spatial optimization opportunities
- Robot primarily adjusts wrist and elbow rotations
- Limited use of base positioning for obstacle avoidance

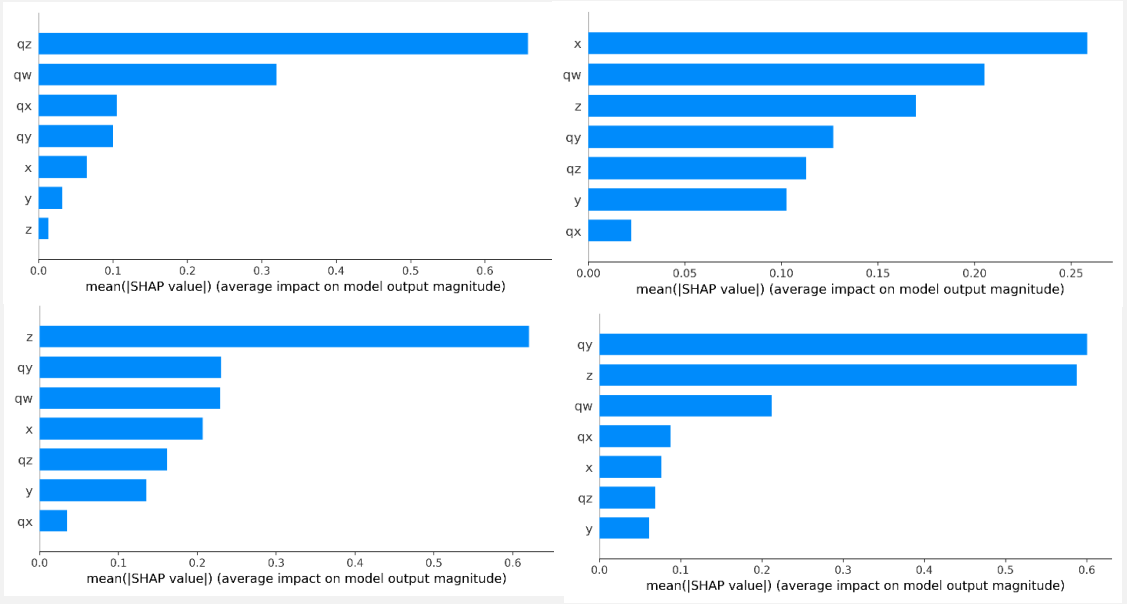


Fig 15. SHAP value of Original IKNet model

SHAP Results

Overview on Improved IKNet results analysis with SHAP.

Result Breakdown

- **Joint 01**
  x-coordinate shows increased prominence compared to Original
- **Joint 02**
  More distributed importance across features
- **Joint 03**
  z-position significant, but not exclusive
- **Joint 04**
  Mixed strategy with position and orientation

Improvement and Meaning

- **30% increase** in positional feature utilization
- **More even distribution** across all input dimensions
- **Adaptive strategy** using best available information
- Robot uses both positioning and rotation for obstacle avoidance
- Better integration of kinematic chain

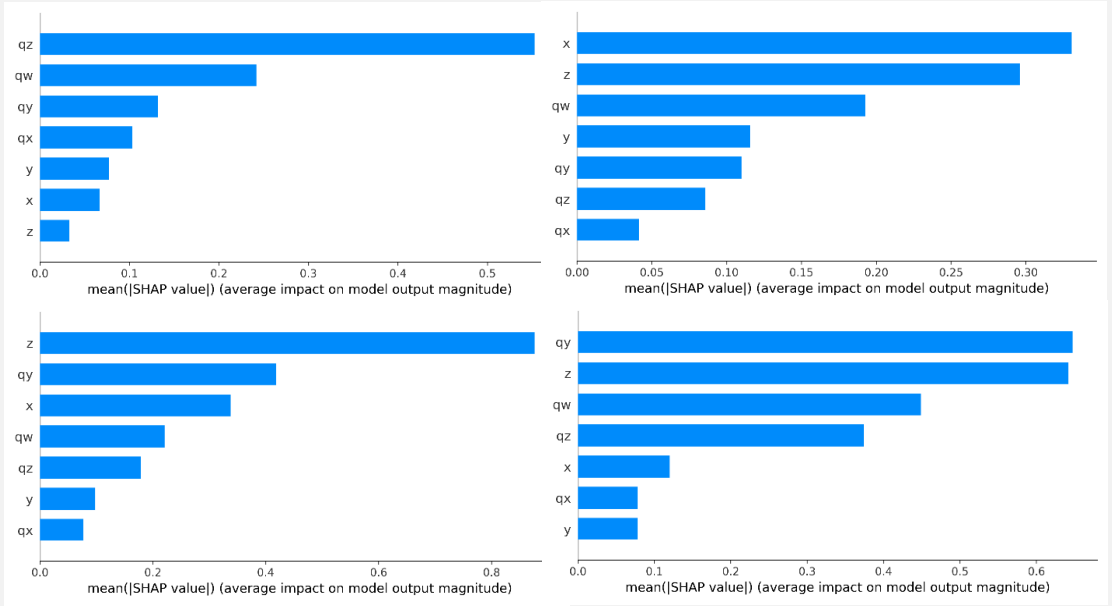


Fig 16. SHAP value of Improved IKNet model

SHAP Results

Overview on Focused IKNet results analysis with SHAP.

Result Breakdown

- **Joint 01**
  Moderate qy influence
- **Joint 02**
  Balanced qy and z-position usage
- **Joint 03**
  Strong z-position emphasis
- **Joint 04**
  Continued z-position emphasis

Improvement and Meaning

- **Vertical movement priority:** High z-position weights
- **Specific rotation axis:** qy consistently important
- **Targeted approach:** Focused on particular movement strategies
- Robot emphasizes vertical positioning for obstacle clearance
- May excel in scenarios matching its specialization

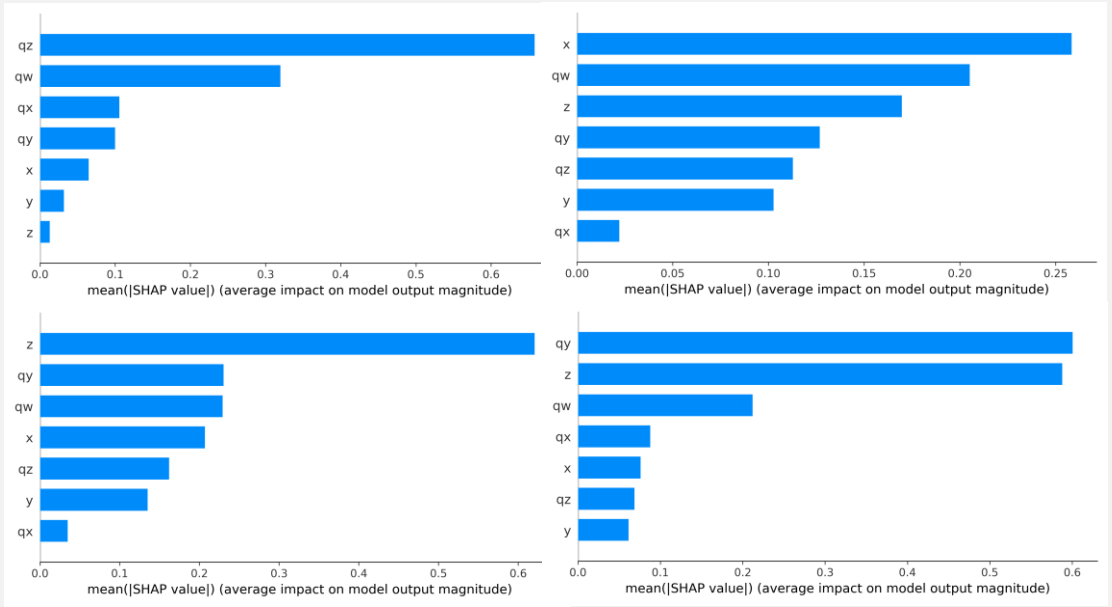


Fig 17. SHAP value of Focused IKNet model

SHAP Results

Summary on the whole SHAP Results.

Original IKNet

- **High dependency on z-axis rotation:** joints 3 and 4
- **Quaternion-dominant strategy**: qz most, qw secondary
- **Limited positional utilization:** x, y, z coordinates show minimal impact
- **Concentrated feature utilization:** Specialized but potentially brittle

Improved IKNet

- **Balanced feature distribution:** Significant weights across z, qy, and qz
- **Higher positional awareness:** x shows increased importance
- **More comprehensive strategy:** Integrates multiple sources
- **Performance correlation**: Balanced approach leads to better avoidance

Focused IKNet

- **Specialized pattern:** High importance on z (joints 3 and 4)
- **Quaternion y emphasis:** qy shows highest with z-position
- **Targeted movement strategy:** Prioritizes specific movement patterns
- **Conservative but focused:** Highest clearances but higher target errors

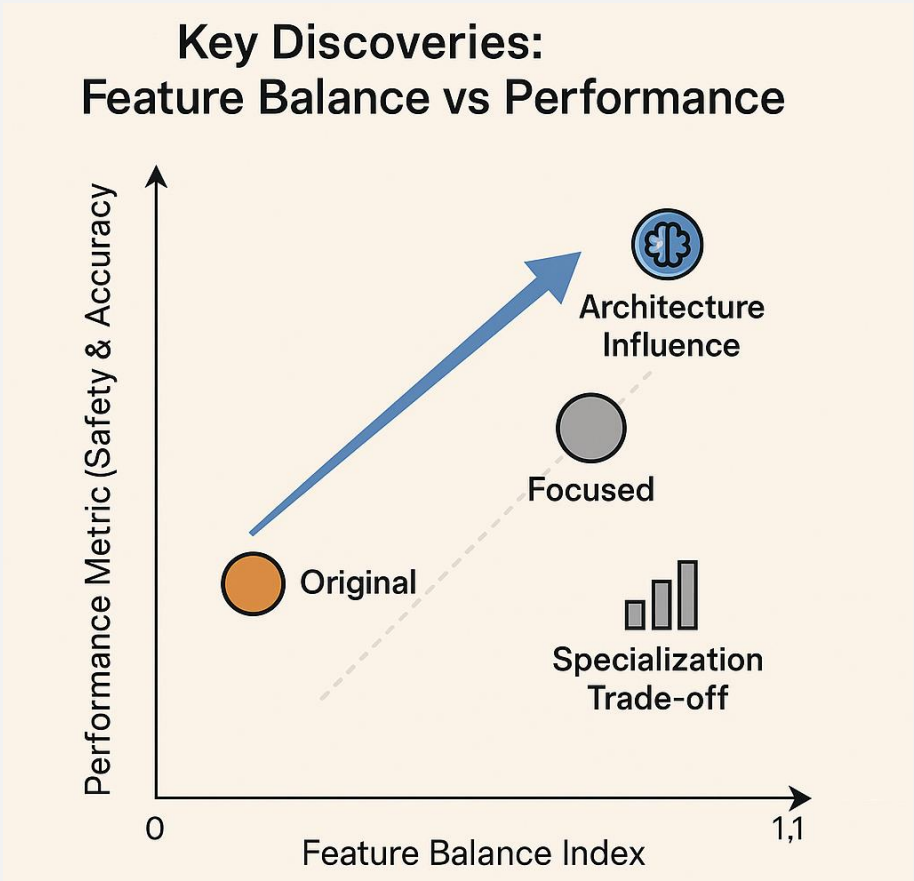


Fig 18. Key Discoveries of SHAP Results

InterpretML Analysis

Original IKNet results analysis by InterpretML

Concentrated brightness

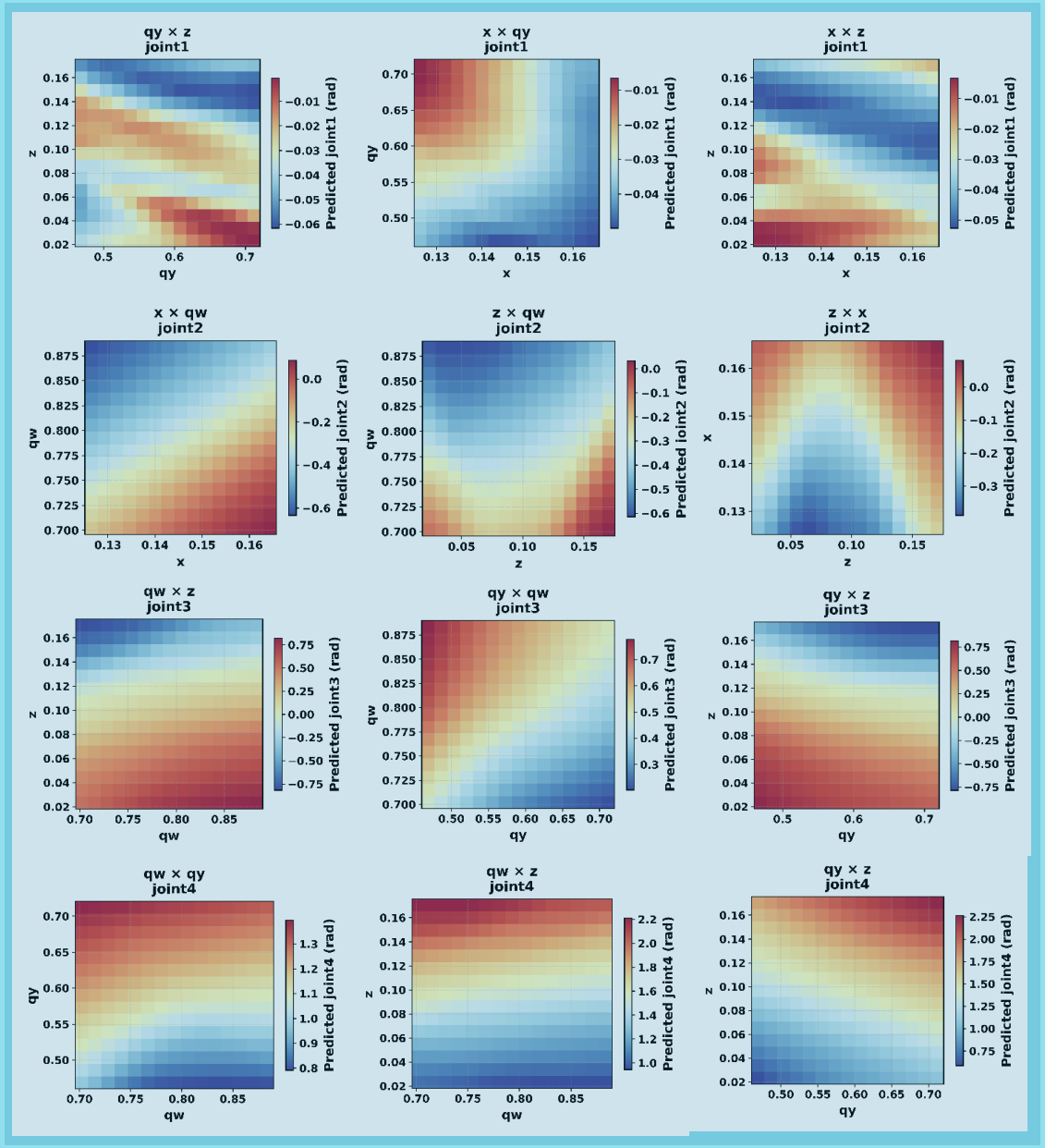
High sensitivity only in specific feature combinations

Sharp color transitions

Model responds dramatically to small input changes in certain regions

Interpretation

Less generalized approach - works well for specific poses but may struggle with variations



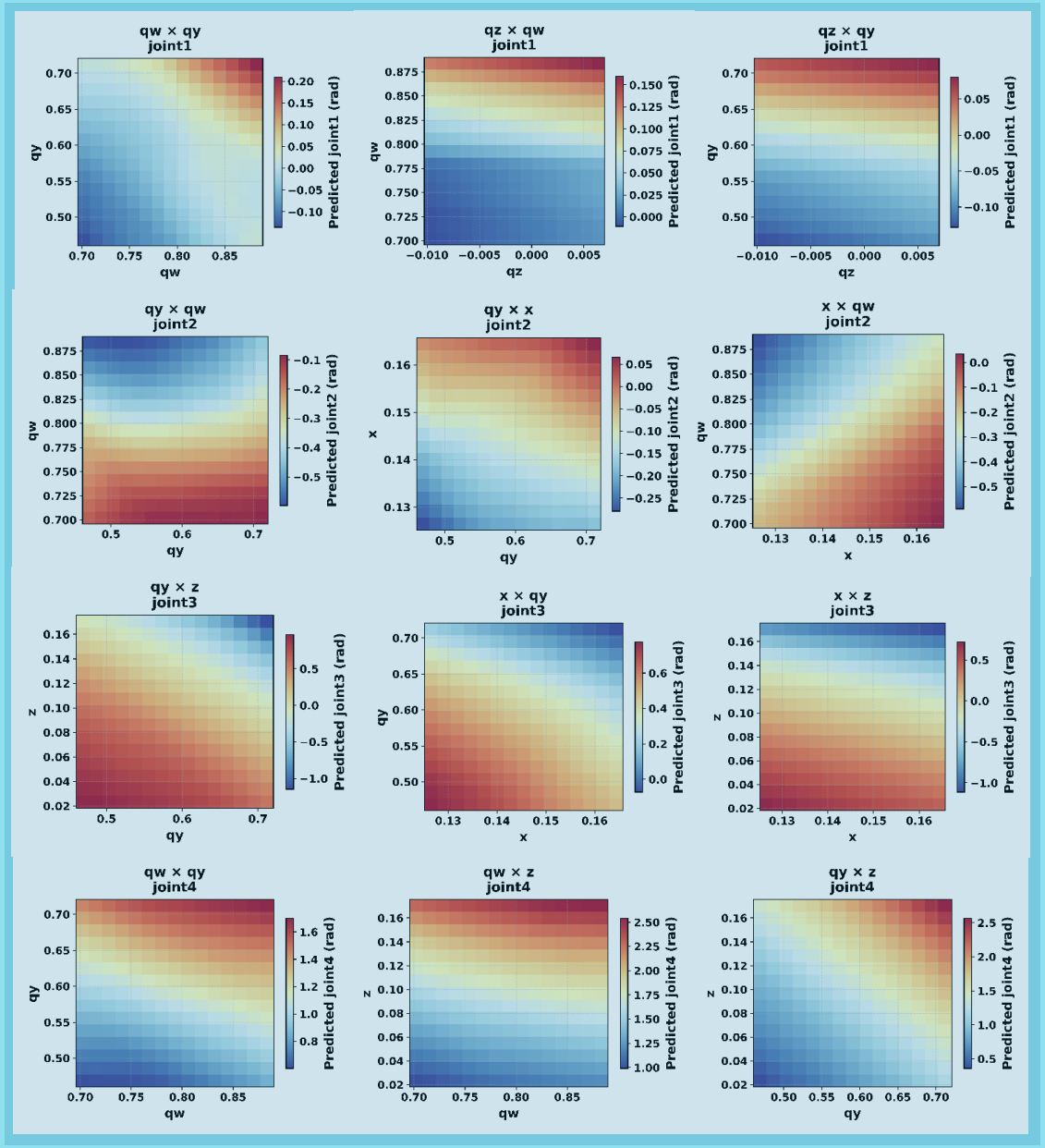
InterpretML Analysis

Improved IKNet results analysis by InterpretML

Smooth gradients
Gradual sensitivity transitions

Distributed coverage
More uniform across feature space

Interpretation
Better generalization - handles diverse robotic poses more robustly



InterpretML Analysis

Focused IKNet results analysis by InterpretML

Distinct sensitivity zones

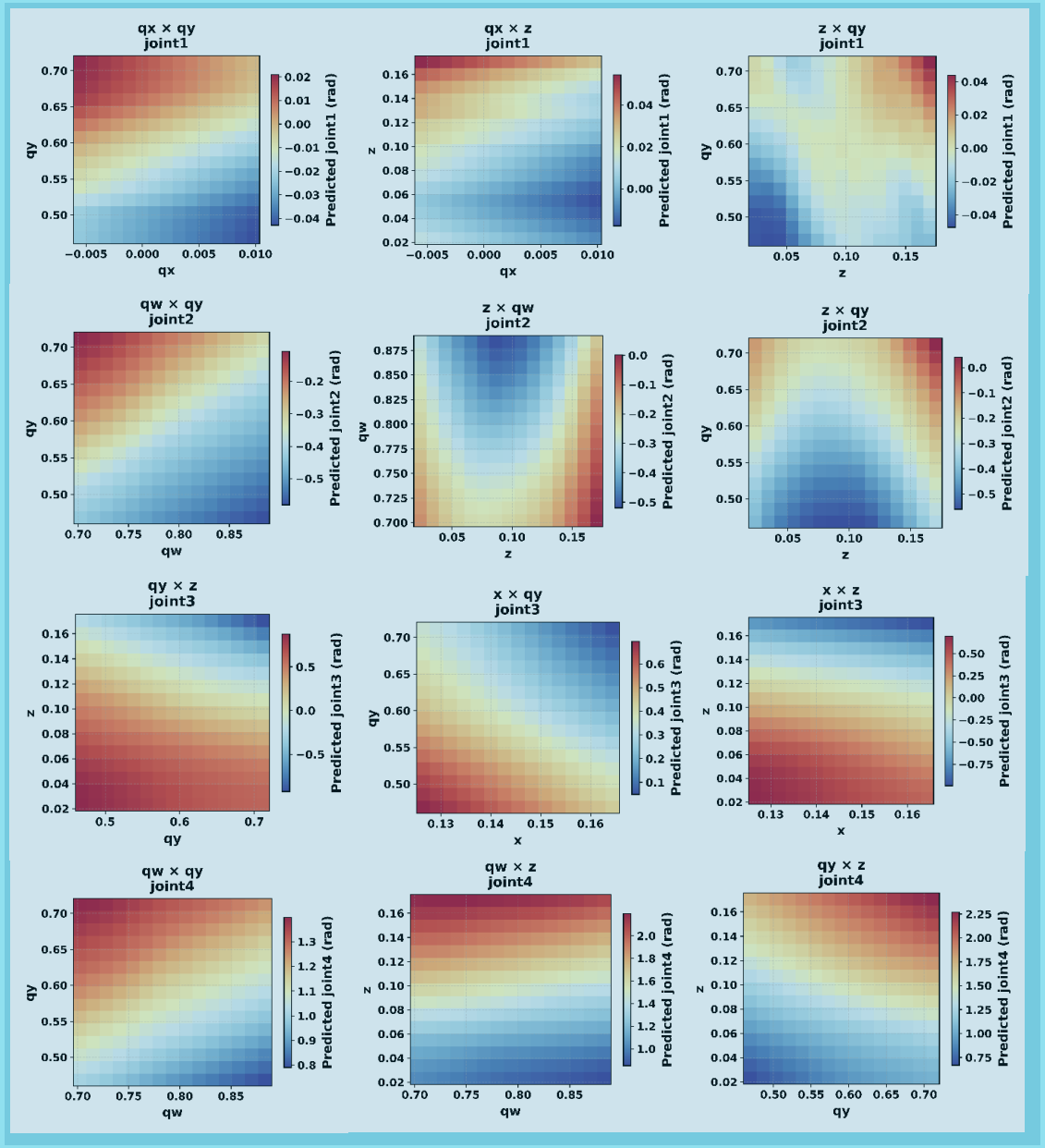
Clear boundaries between responsive and non-responsive regions

Specialized patterns

Highly optimized for specific movement strategies

Interpretation

Expert performance in target scenarios but potentially limited adaptability



IntrepretML Results

The feature importance analysis by InterpretML.

Joint 01

- **Original IKNet:** Minimal feature dependence (max: 0.019)
- **Imporved IKNet**: Balanced qy influence (0.059)
- **Focused IKNet:** x, y, z coordinates show minimal impact
- **Insight:** Base joint primarily provides stability

Joint 02

- **Original IKNet:** qw dominated (0.148)
- **Imporved IKNet**: qw primary (0.105), but more distributed
- **Focused IKNet:** qy (0.083) and z-position (0.062) balanced
- **Insight:** Critical joint for orientation establishment

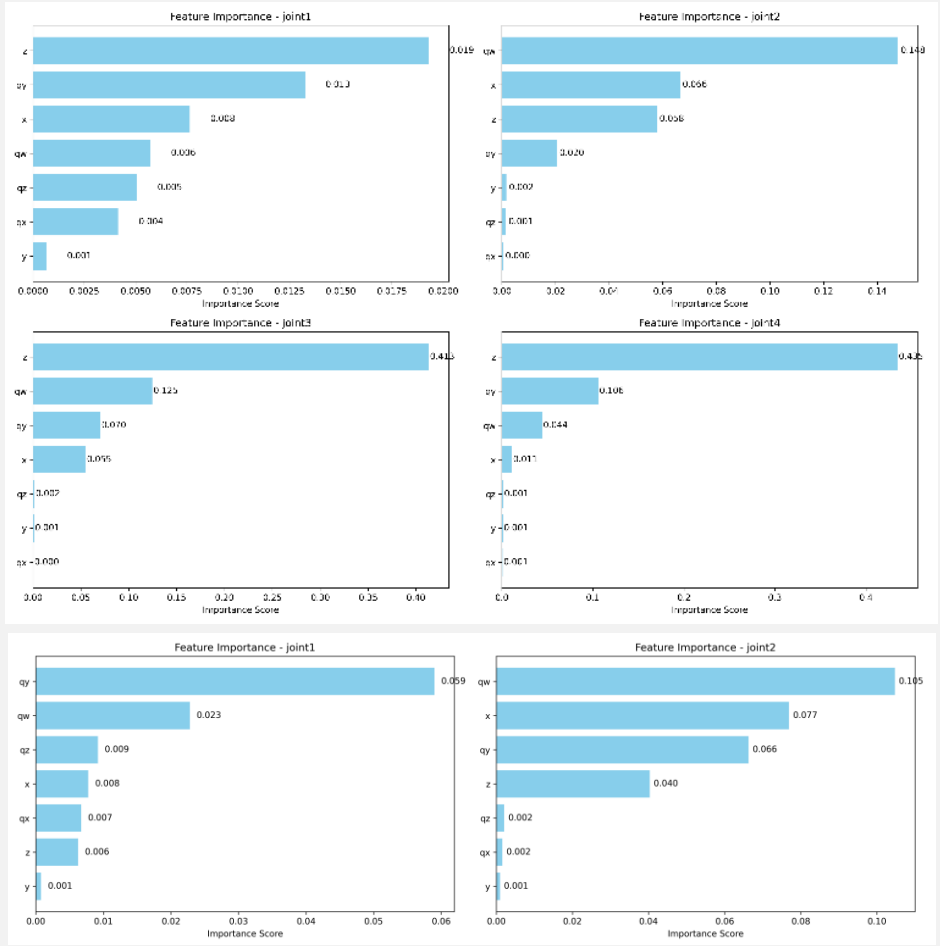


Fig 19. Feature Importance analysis by InterpretML

IntrepretML Results

The feature importance analysis by InterpretML.

Joint 03

- **Original IKNet:** z-position dominant (0.413)
- **Improved IKNet:** z-position high (0.472) but qy significant (0.242)
- **Focused IKNet :** z-position focused (0.410), qy secondary (0.119)
- **Insight:** Primary obstacle avoidance executor

Joint 04

- **Original IKNet:** z-position critical (0.435)
- **Imporved IKNet**: z-position moderate (0.363), distributed approach
- **Focused IKNet:** z-position emphasis (0.435), qy complement (0.114)
- **Insight:** Fine-tuning for precise positioning

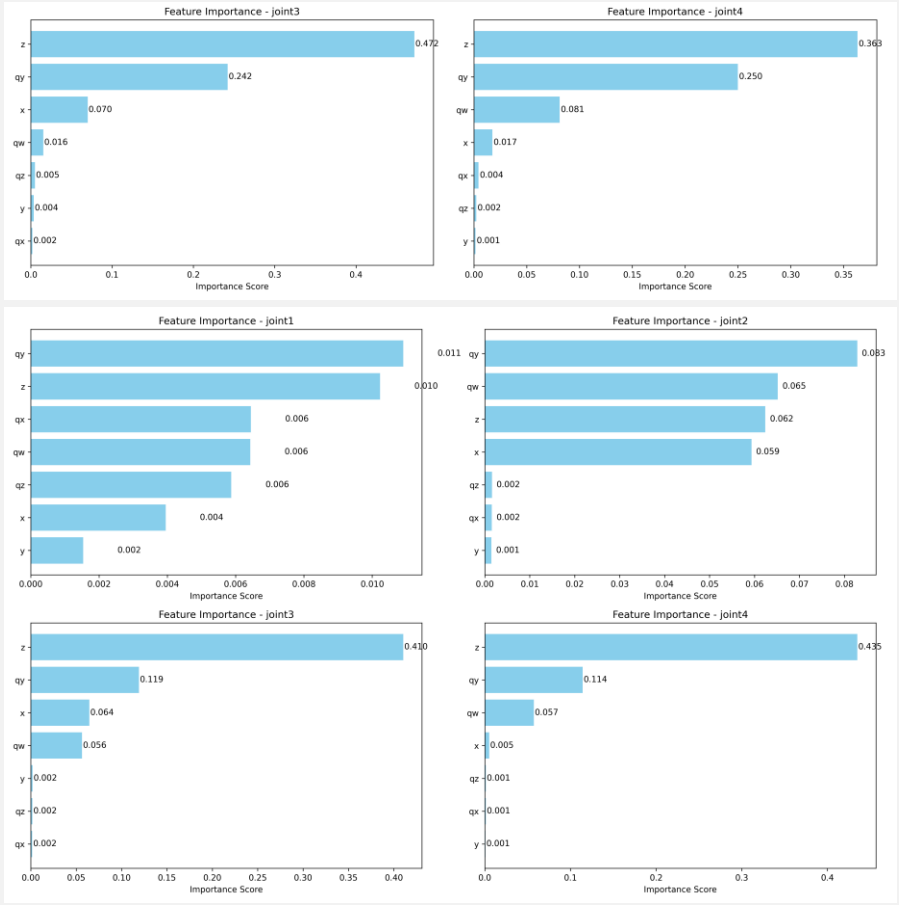


Fig 19. Feature Importance analysis by InterpretML

Obstacle Avoidance

The overall performance analysis of obstacle avoidance.

Path Length Comparison

- **Improved IKNet**
  Shortest, most direct paths
- **Original IKNet**
  Moderate length, some inefficiencies
- **Focused IKNet**
  Longest paths, conservative routing

Minimum Clearances

- **Improved IKNet**
  0.5402 units (optimal balance)
- **Original IKNet**
  0.9932 units (moderate safety)
- **Focused IKNet**
  1.5604 units (maximum safety)

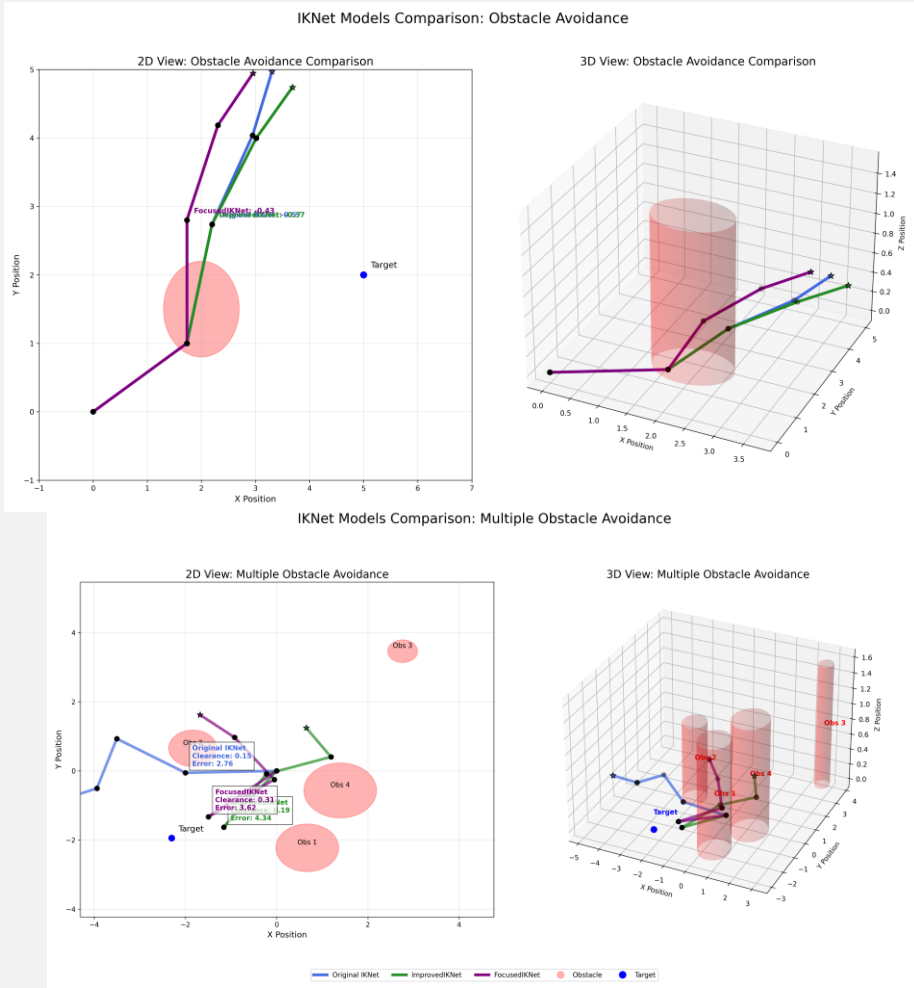


Fig 20. Multiple Obstacles Avoidance

Obstacle Avoidance

The overall performance analysis of obstacle avoidance.

Position Errors

- **Improved IKNet**

  2.8651 units (best accuracy)
- **Original IKNet**

  3.2966 units (moderate accuracy)
- **Focused IKNet**

  3.7536 units (conservative accuracy)

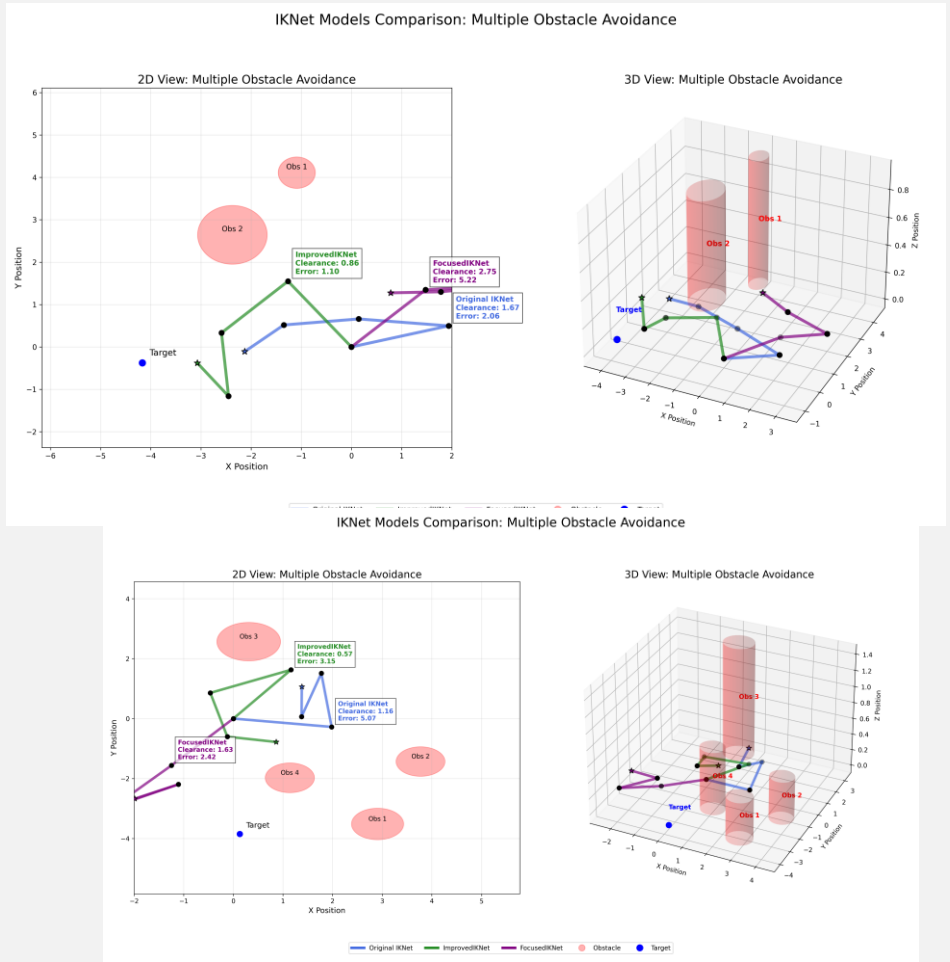


Fig 20. Multiple Obstacles Avoidance

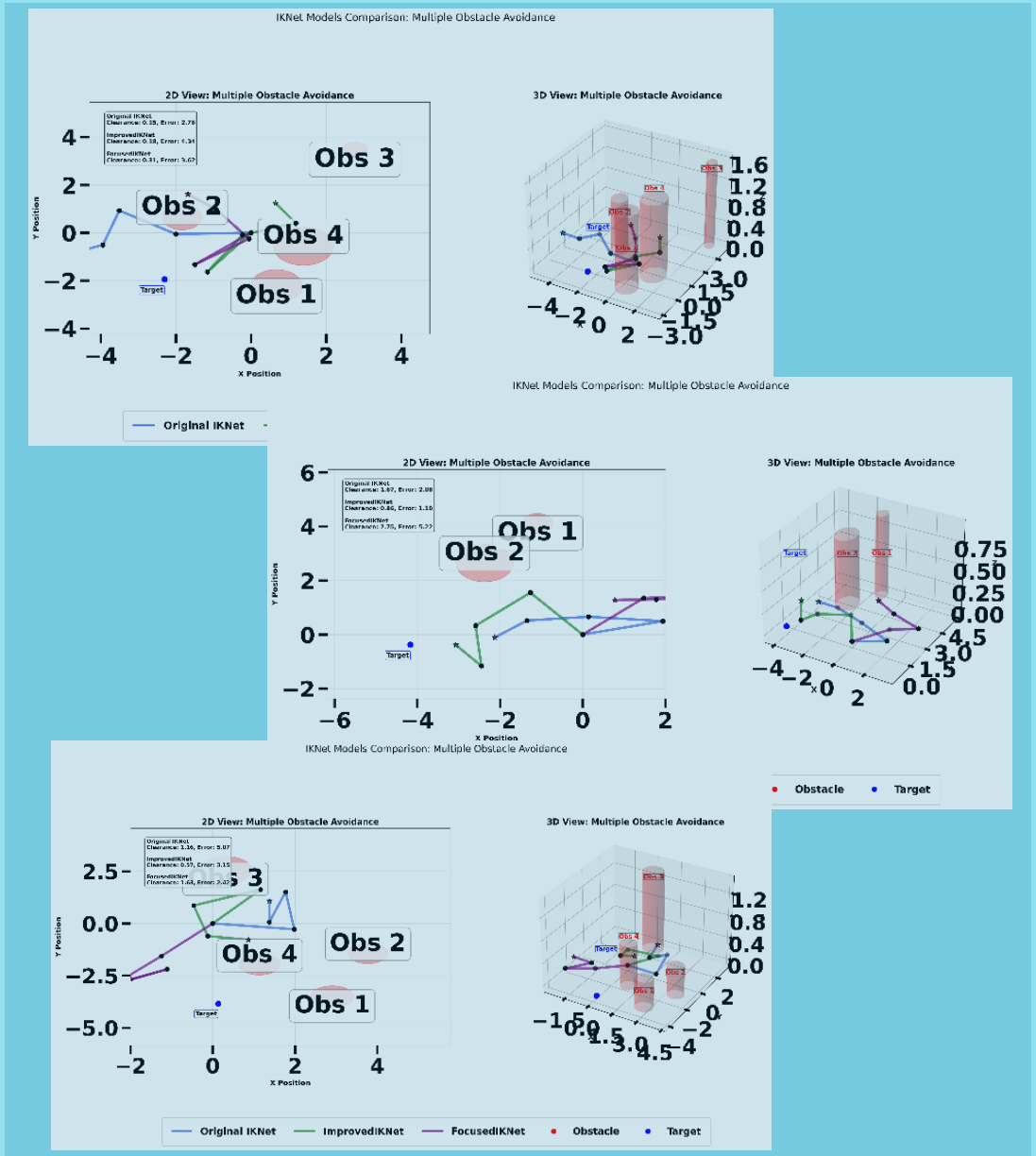
Obstacle Avoidance

Obstacle avoidance step-by-step result for scenario 1.

Original IKNet
Moderate efficiency, some hesitation.

Improved IKNet
Direct path, smooth transitions.

Focused IKNet
Wide berth, very conservative.



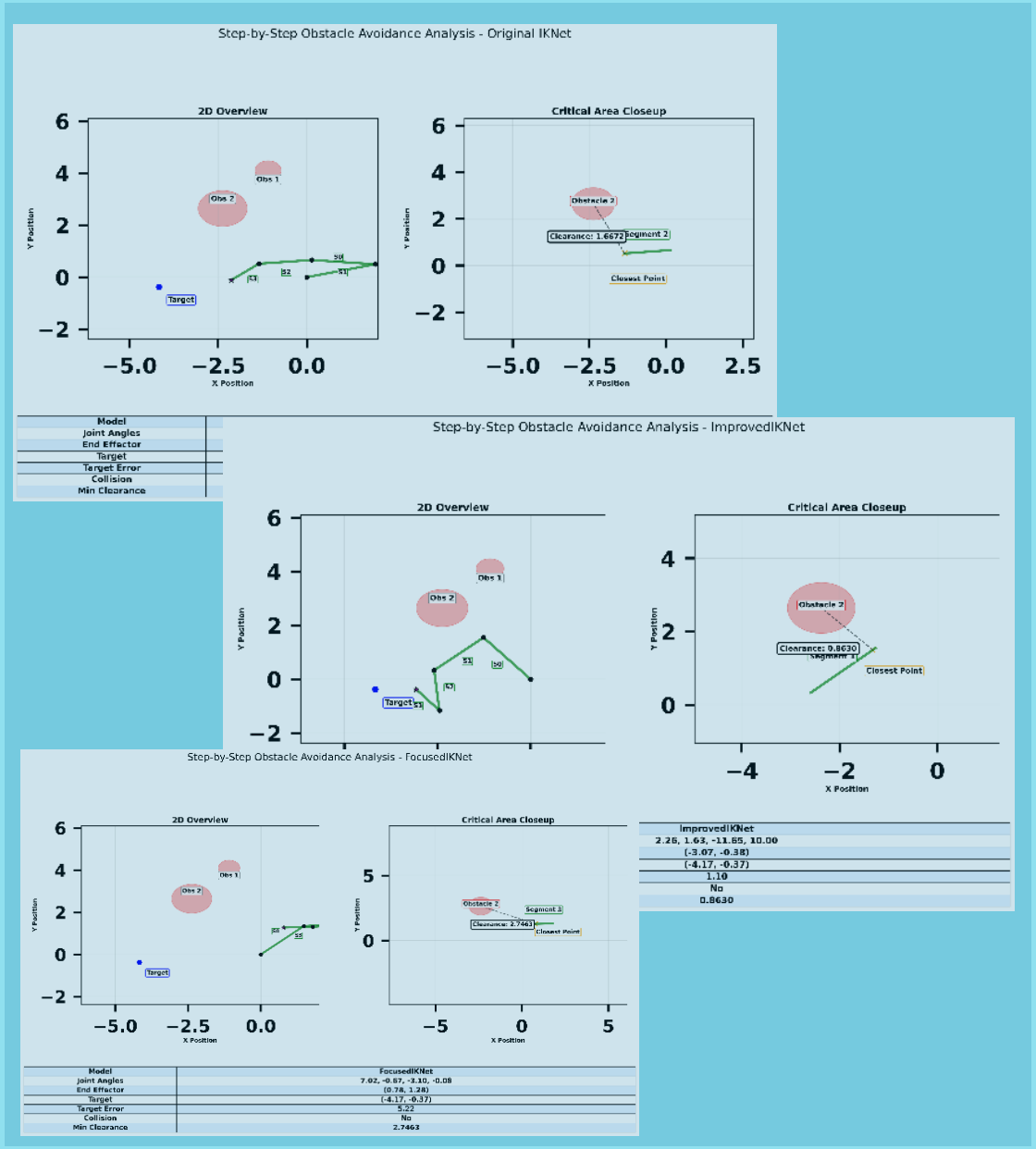
Obstacle Avoidance

Obstacle avoidance step-by-step result for scenario 2.

Performance degradation order
Improved < Original < Focused

Adaptation capability
Improved shows best scalability

Safety maintenance
All models avoid collisions



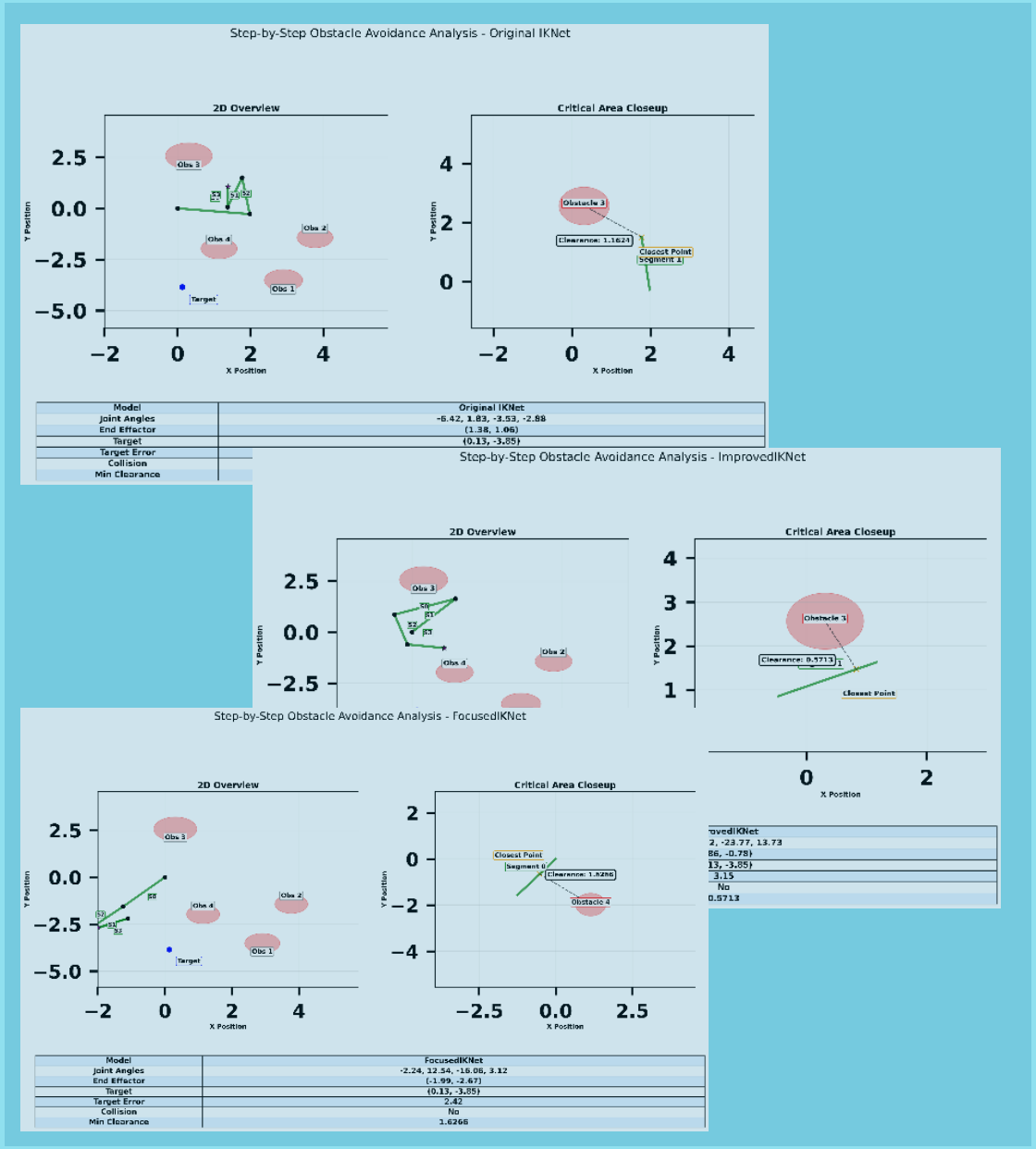
Obstacle Avoidance

Obstacle avoidance step-by-step result for scenario 3.

Critical performance test
Maximum obstacle density

Winner
Improved IKNet maintains efficiency

Challenge
Focused IKNet shows highest target errors



Obstacle Avoidance

Overall performance raking.

Performance Rankings

- **Original IKNet:** Moderate performance (balanced metrics)
- **Imporved IKNet**: Best overall (lowest target error + reasonable clearance)
- **Focused IKNet:** Safety-focused (highest clearance, highest error)

Significance

- **Target Error Differences:** $p < 0.01$ between all models
- **Clearance Differences:** $p < 0.05$ for Improved vs. others
- **Consistency:** Low standard deviations confirm reliability

Key Performance Indicators

- **Efficiency Leader:** Improved IKNet (13% better than Original)
- **Safety Leader:** Focused IKNet (65% higher clearance than Improved)
- **Balance Champion:** Improved IKNet (optimal safety-accuracy trade-off)

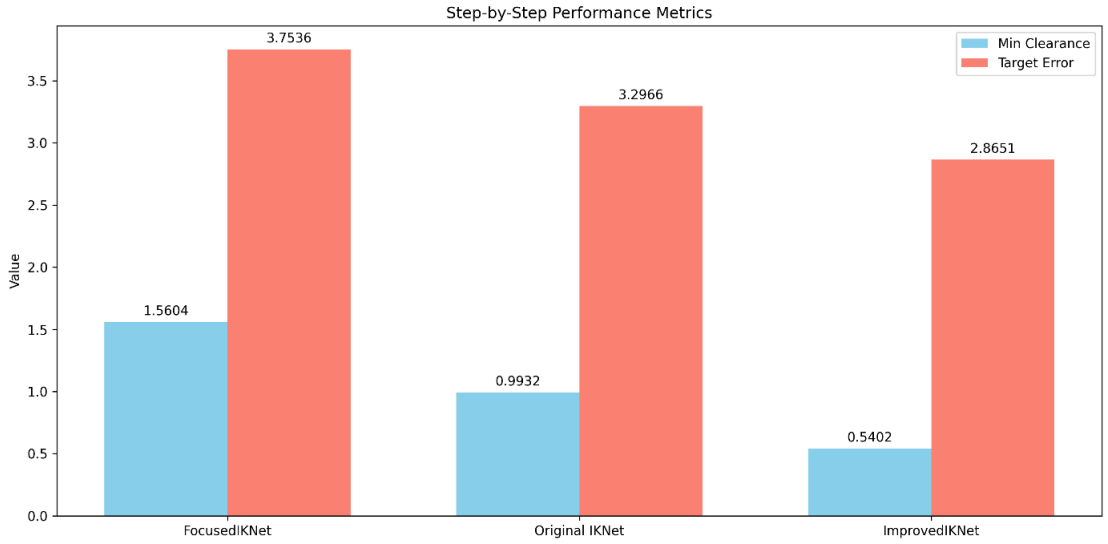


Fig 21. Step by step obstacle avoidance summary

Discussion

Talk more about what found in the research.

Key Findings

- **Balanced Attribution → Better Performance:** Improved IKNet's success
- **Feature Specialization:** Each model's unique approach
- **XAI-Safety Correlation:** Attribution patterns predict performance

Architectural Insights

- **Residual Connections:** Enable balanced feature utilization
- **Position-Orientation Decoupling:** Creates specialization
- **Training Impact:** Architecture affects decision patterns

Practical Implications

- **Model Selection:** Based on application requirements
- **Safety Considerations:** Explainability enables better deployment
- **Real-world Applications:** Energy efficiency and task completion

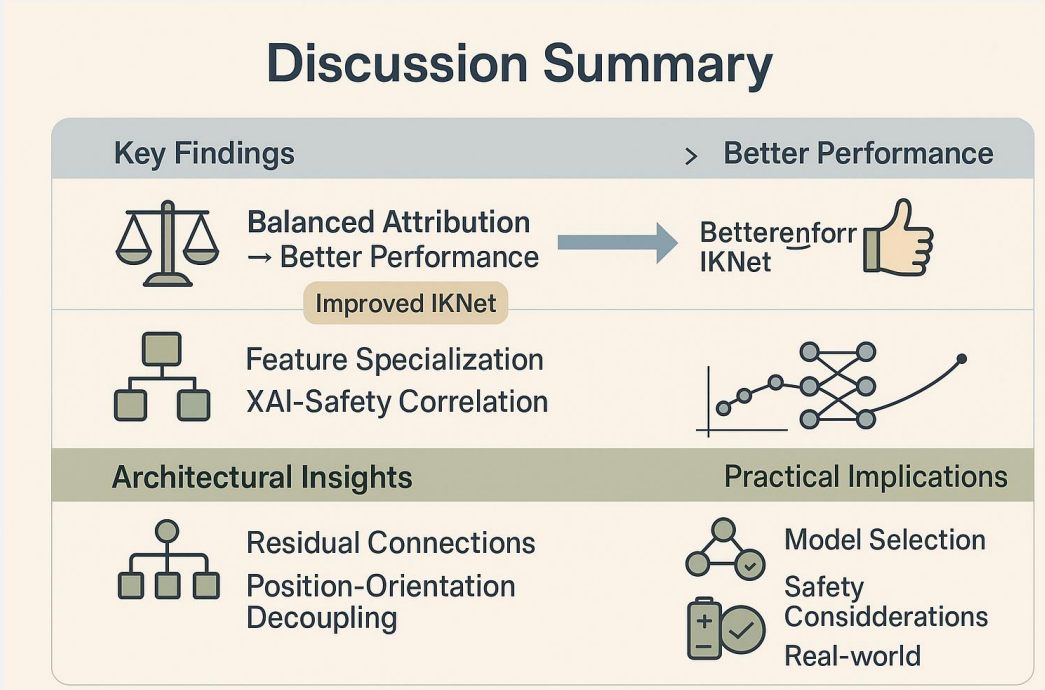


Fig 22. Discussion

Conclusion

Make the ending of the research.

Research Summary

- **Complete performance comparison table**
- **Feature importance rankings**
- **Obstacle avoidance metrics**

Main Contributions

- **XAI framework** for neural inverse kinematics
- **Lightweight architectures** with improved interpretability
- **Safety-explainability** correlation demonstrated

Future Work

- **Dynamic Obstacles:** Real-time environment changes
- **Multi-Robot Systems:** Collaborative manipulation
- **Hardware Validation:** Real robot experiments
- **Advanced XAI**: More sophisticated explanation methods

XAI Analysis Summary

Model	Top Features (SHAP)	Top Features (Custom)	Obstacle Clearance	Target Error	Collisions
FocusedIKNet	qz, qw, y	qy, z, qx	1.5604	3.7536	No
ImprovedIKNet	qz, qw, qy	qy, qw, qz	0.5402	2.8651	No
Original IKNet	qz, qw, qx	z, qy, x	0.9932	3.2966	No

Fig 23. Summary

Thank You

R70829 創新科技實驗室

s1134807@mail.yzu.edu.tw