Synthetic Data-Driven Mismatch Detection for Updating BIM-Based Digital Twins from a Single Viewpoint

PEPIJN HUNDEPOOL

26-06-2025

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

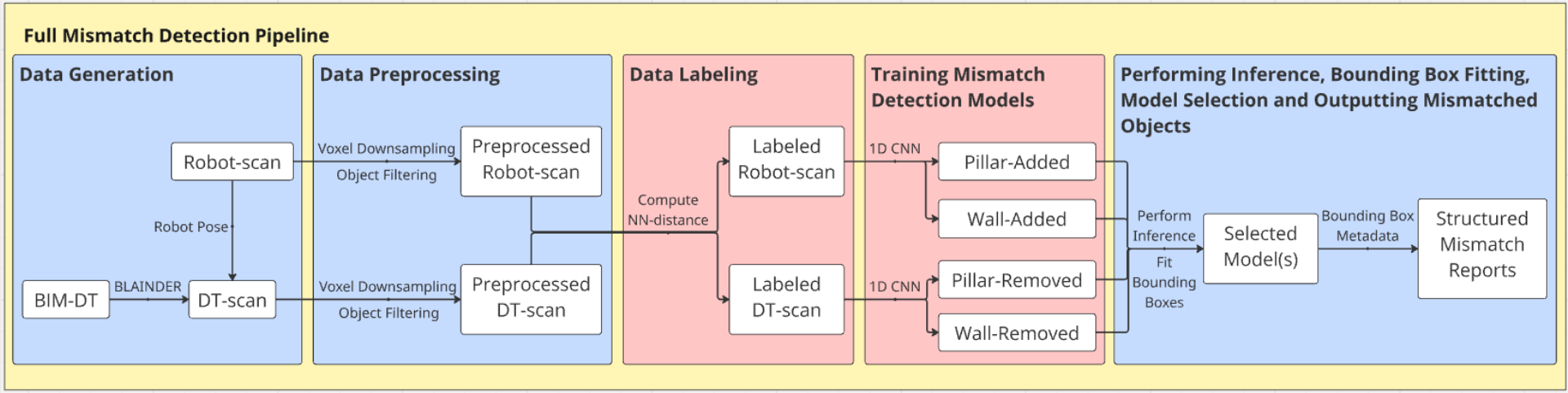
Introduction

- BIM
- Digital Twin
- Buildings evolve
- Mobile robots
- Mismatched structural elements
- Single viewpoint

Contents

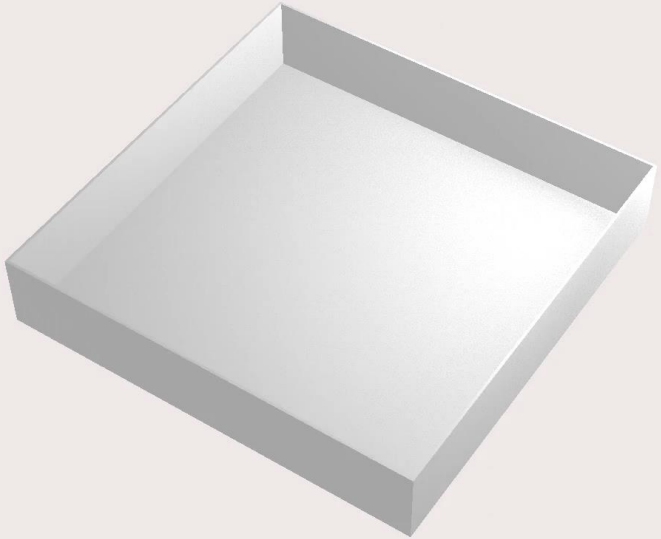
- Methodology
  - Data generation
  - Preprocessing
  - Mismatch detection models
- Results
  - Individual model performance
  - Ablation studies
  - Semi-real-world validation
  - Multi-mismatch handling
- Conclusion

Phase 1 – Data Generation



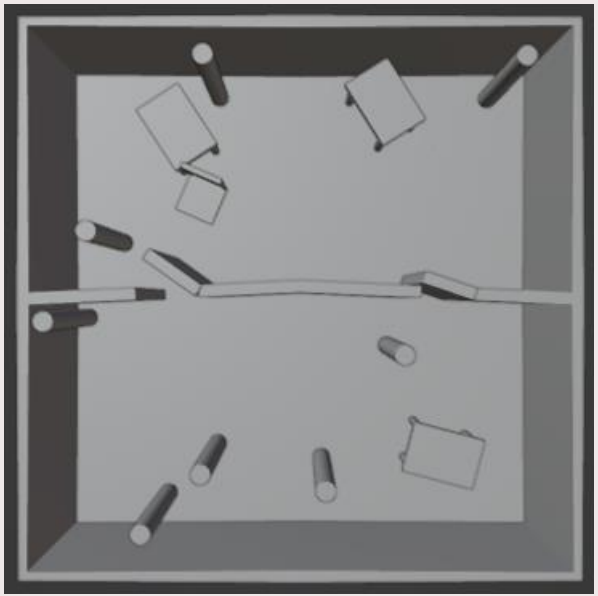
Room Generation

- Parametric dataset generator
- Object set
- Random room layouts
- Serve as digital twins

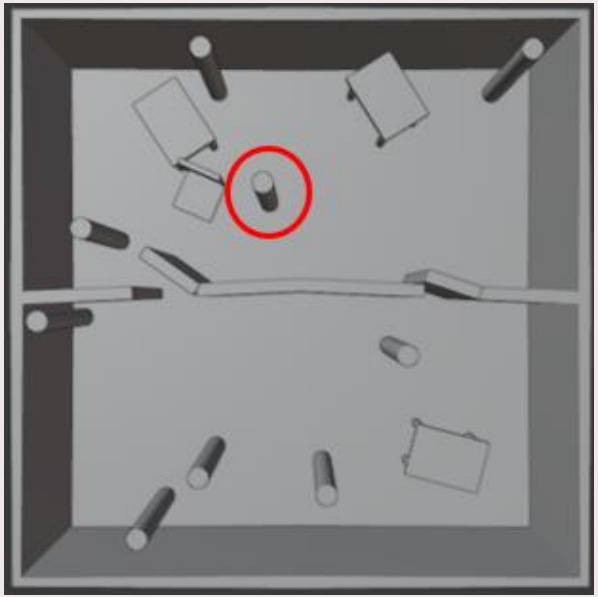


Parametric Dataset Generator developed by Pim van den Akker in: *"A Parametric Dataset Generator For Updating Digital Twins Through 2D LiDAR Perceptual Data",* (2024).

Scenario: Pillar Added

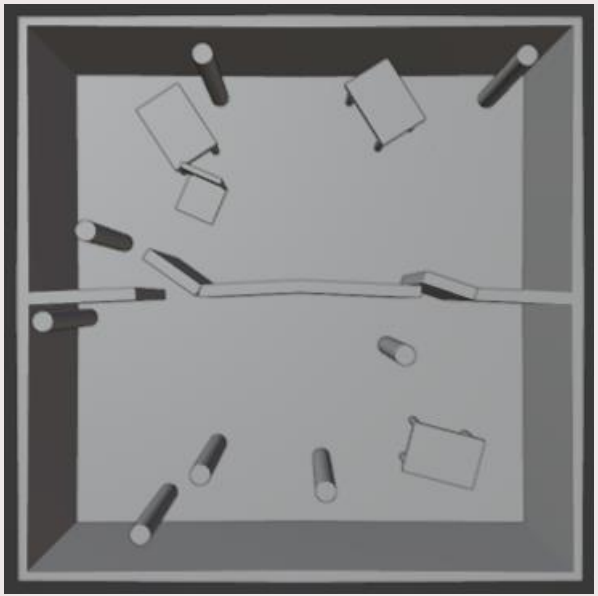


Original

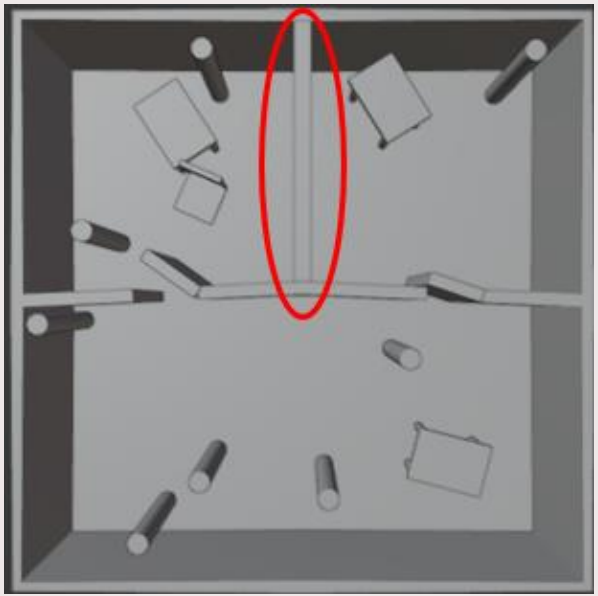


Modified

Scenario: Wall Added

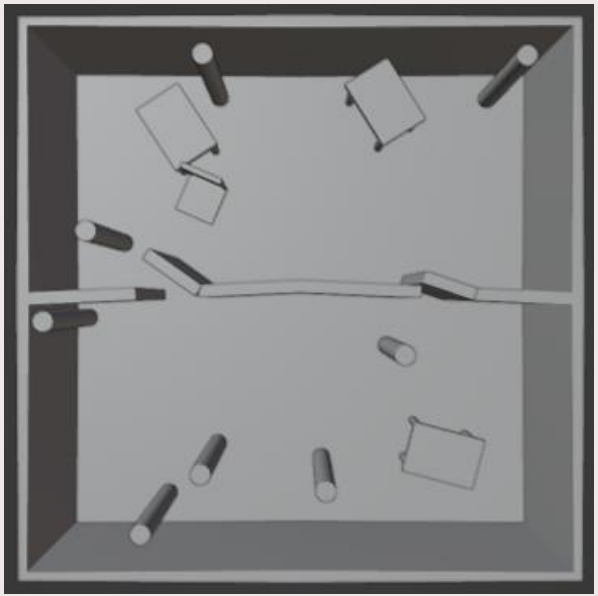


Original

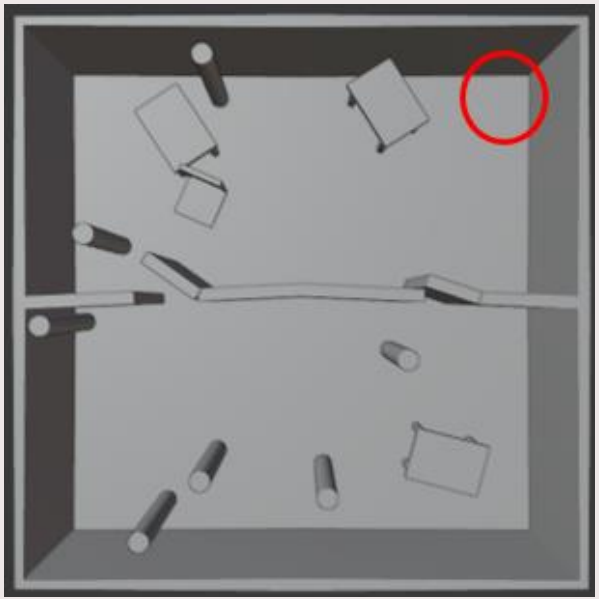


Modified

Scenario: Pillar Removed

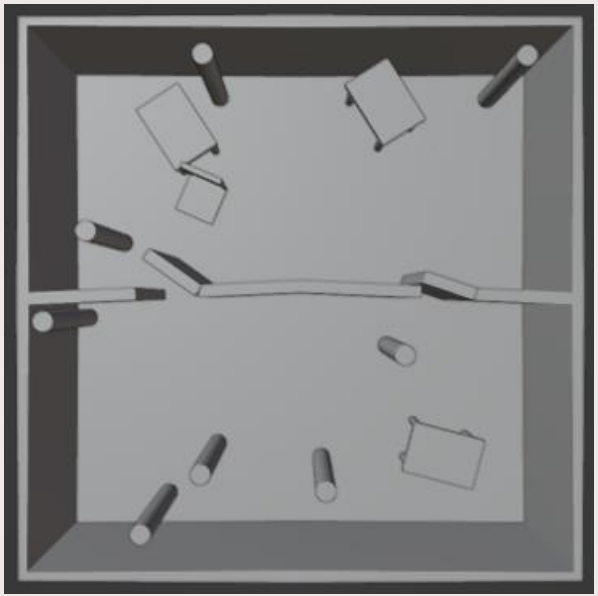


Original

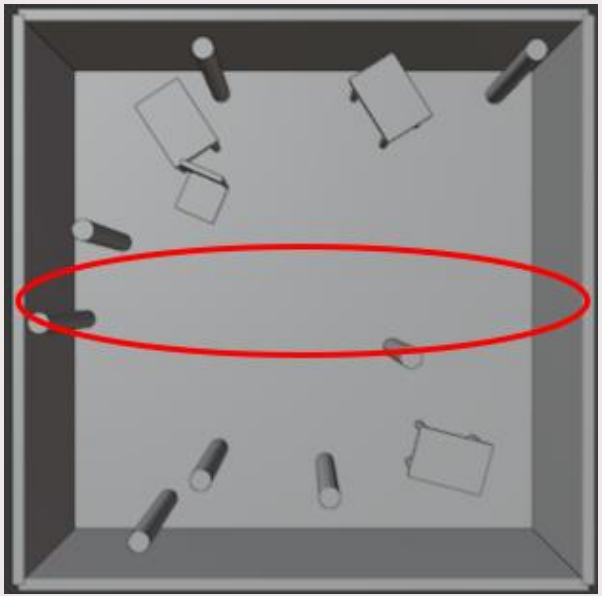


Modified

Scenario: Wall Removed



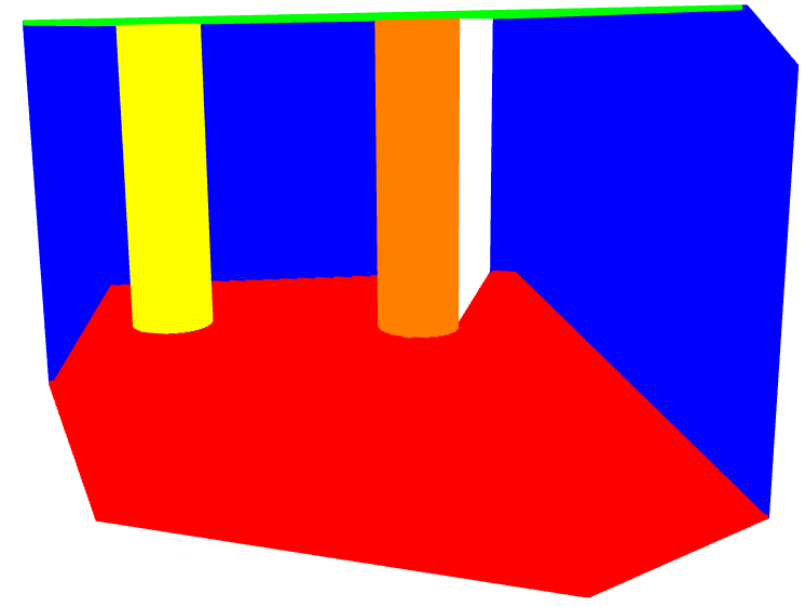
Original



Modified

3D Sensor Simulation

- BLAINDER
- Virtual RGB-D camera
- Geometry and semantics
- Direct instance-level grouping

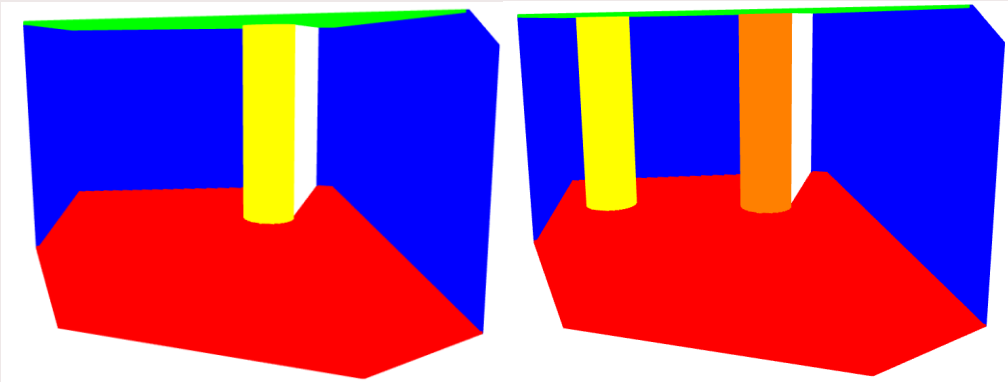


Scan example. Objects are coloured by categoryID.

TU/e

Mismatches and Scenario Control

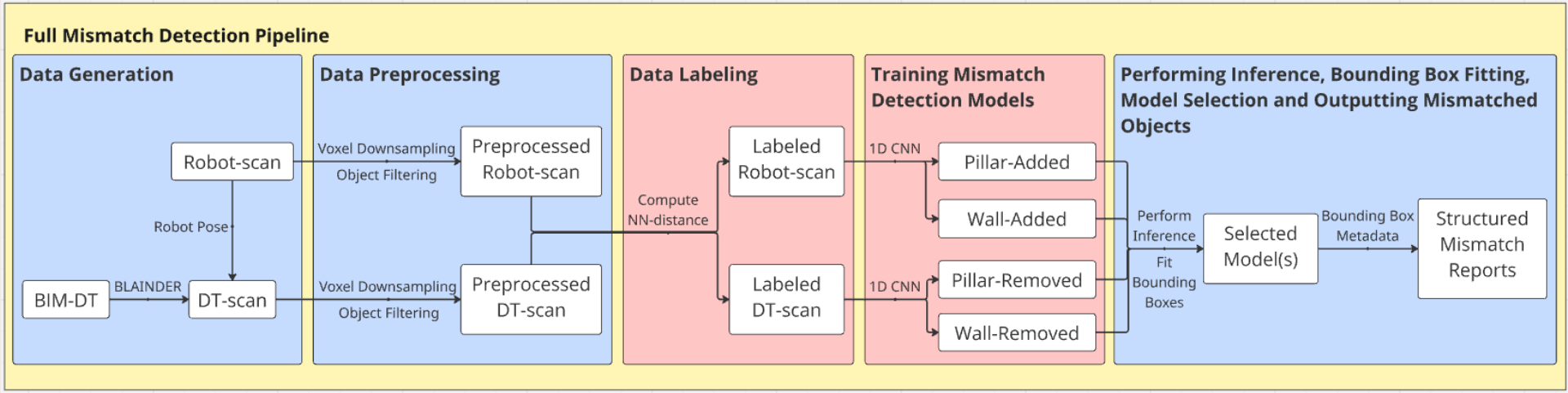
- DT scan
- Mismatches
- Robot scan
- Occluding objects
- Pre-aligned scenes



DT scan

Robot scan (without occluding objects)

Phase 2 – Data Preprocessing

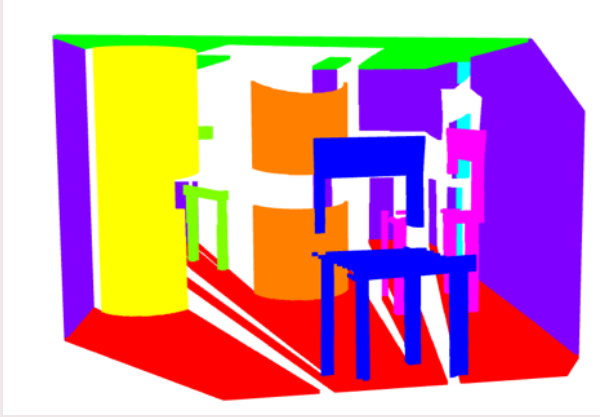


Data Preprocessing

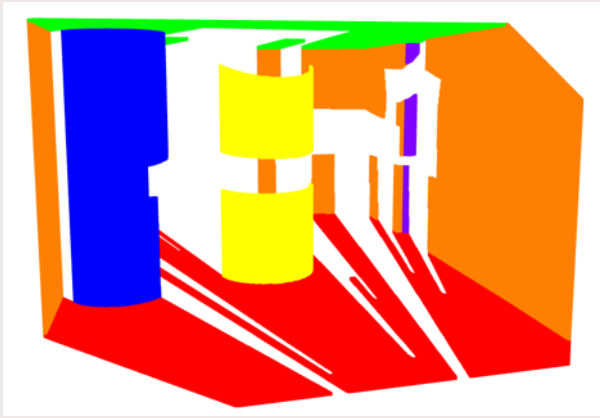
- Object filtering
- Downsampling
- Label generation
- Data augmentation
- Formatting

Object Filtering

- Non-structural elements
- Filter robot scan
- CategoryID & height
- Assume access to semantics



Raw scan

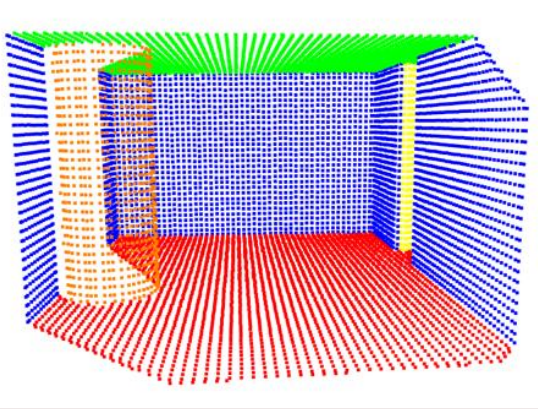


Object-filtered scan

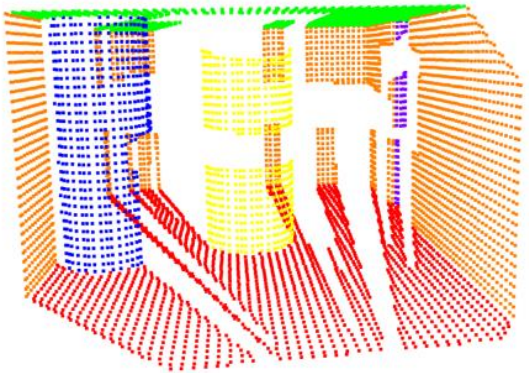
TU/e

Downsampling

- Uniform point density
- Reduce memory usage
- Voxel downsampling
- Smoothing effect

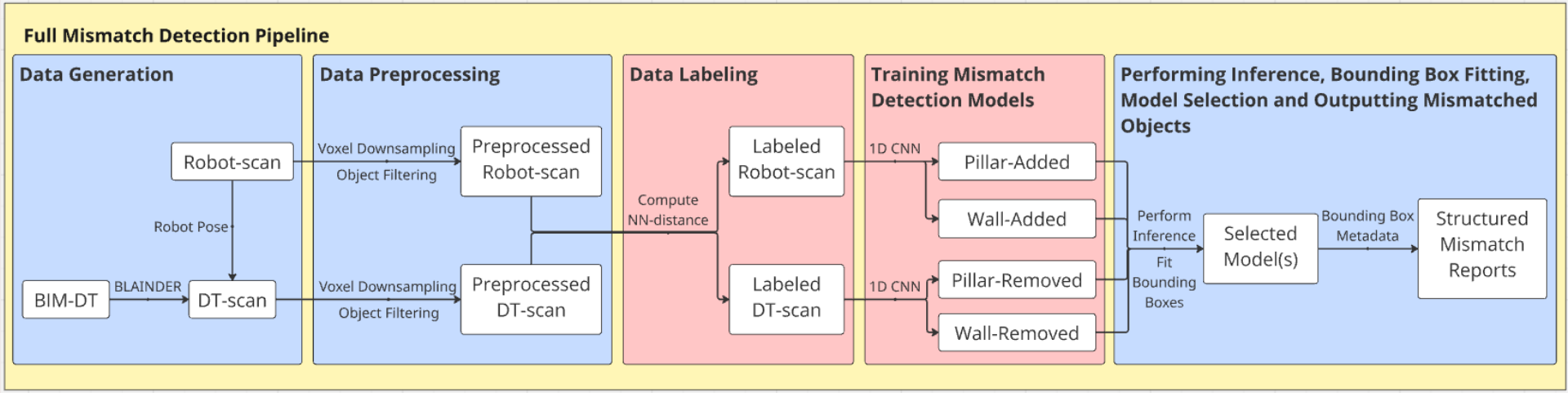


DT scan downsampled



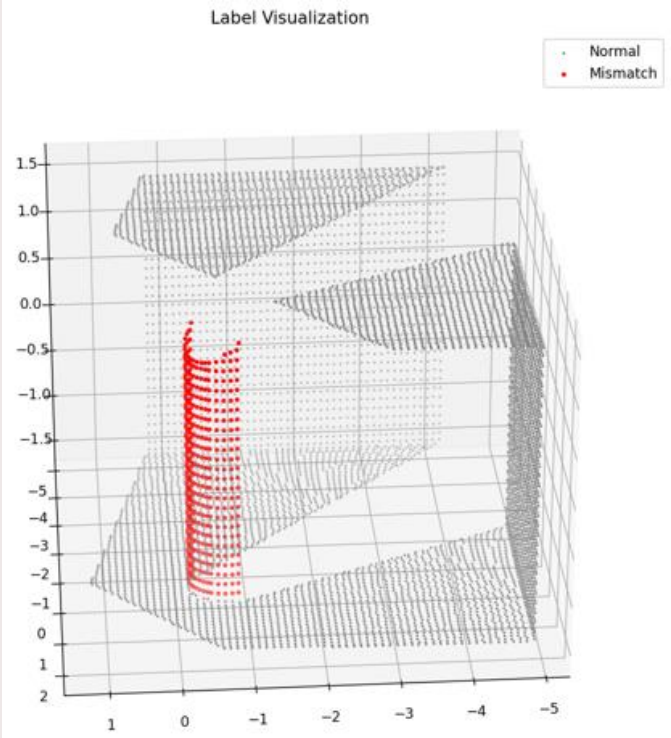
Robot scan downsampled

Phase 3 – Data Labeling



Label Generation & Filtering

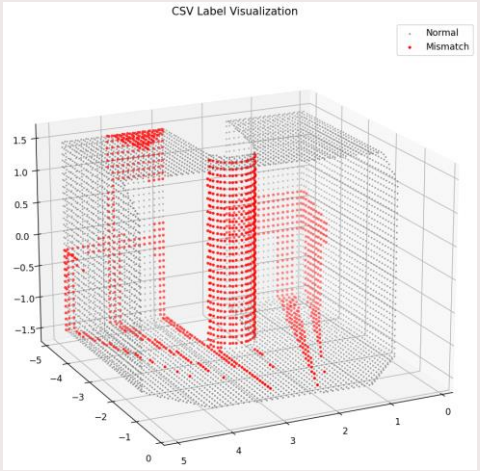
- Accurate per-point labels
- Nearest-neighbour distance
- Mapping from distance to likelihood
- Scenario-dependent
- Unchanged cases



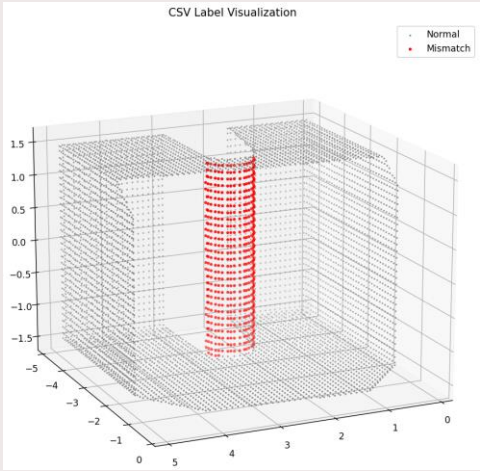
Generated labels for a pillar-added scenario

Label Generation & Filtering

- Occlusion shadows
- False mismatches
- Nearest robot point
- DT scans have no occlusions
- Cluster-based filtering



Generated labels

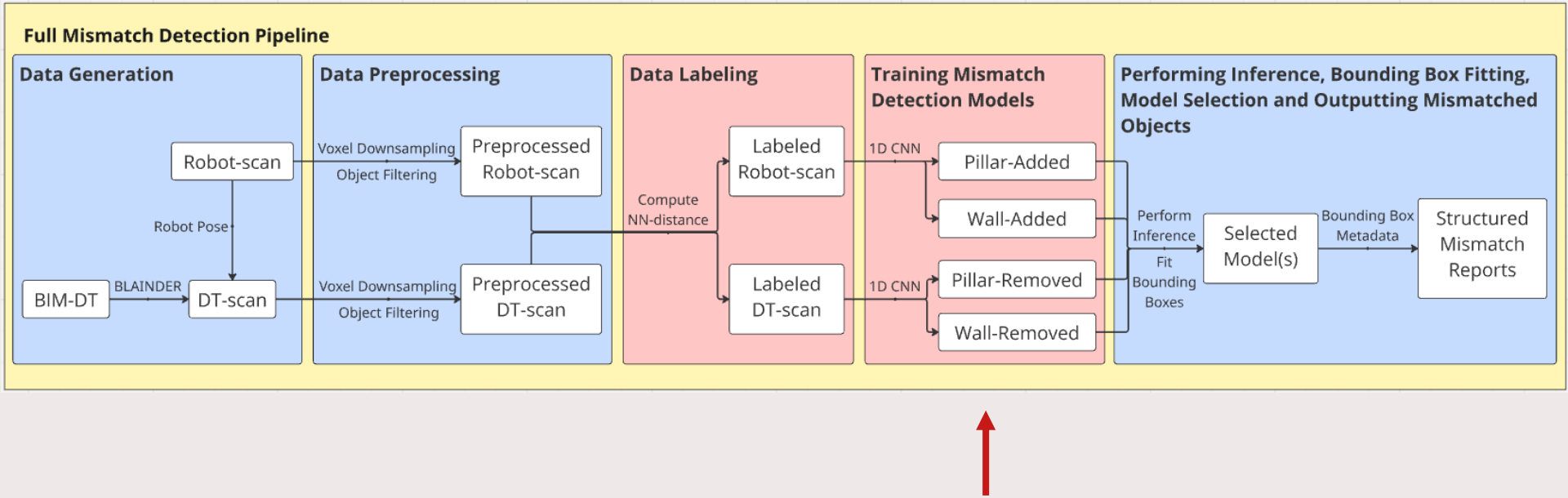


Filtered labels

Data Augmentation

- Prevent overfitting to absolute coordinates
- Pair-wise spatial augmentations
- Random rotation and/or translation

Phase 4 – Training Mismatch Detection Models



Mismatch Detection Neural Network Architecture

- 1D CNN
- Per-point classification
- Nearest-neighbour distance as a feature
- Mismatch probability

Input Representation

- Feature vector
- Paired with a label file
- Unchanged samples

$$\mathbf{X}_i = (x_i, y_i, z_i, f_i)$$

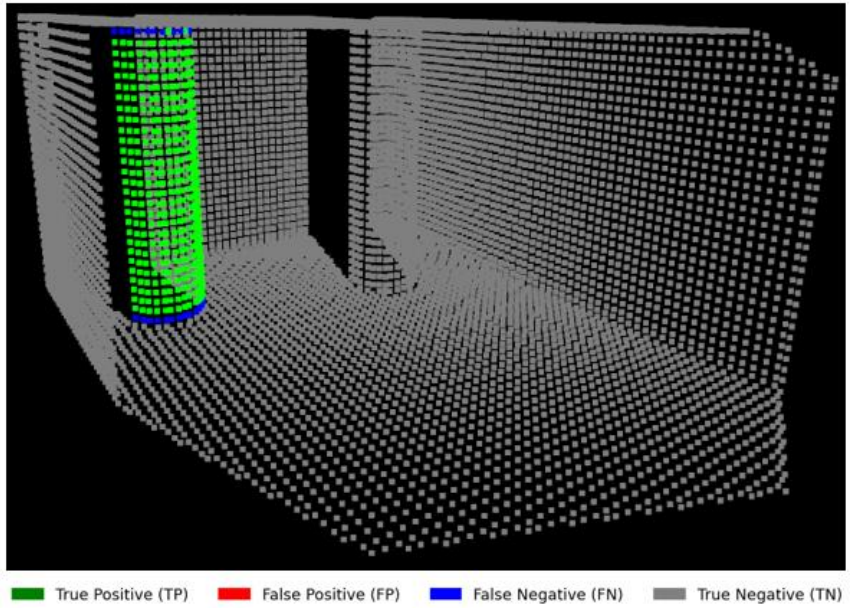
Testing

1. Testing each specialized model on controlled single-mismatch scenarios.
2. A set of ablation studies to evaluate pipeline components
3. Selecting the best specialized mismatch detection model for test cases where mismatch type is unknown beforehand.
4. Validation on scans derived from real BIM (IFC) models.
5. Testing each specialized model on controlled multi-mismatch scenarios.

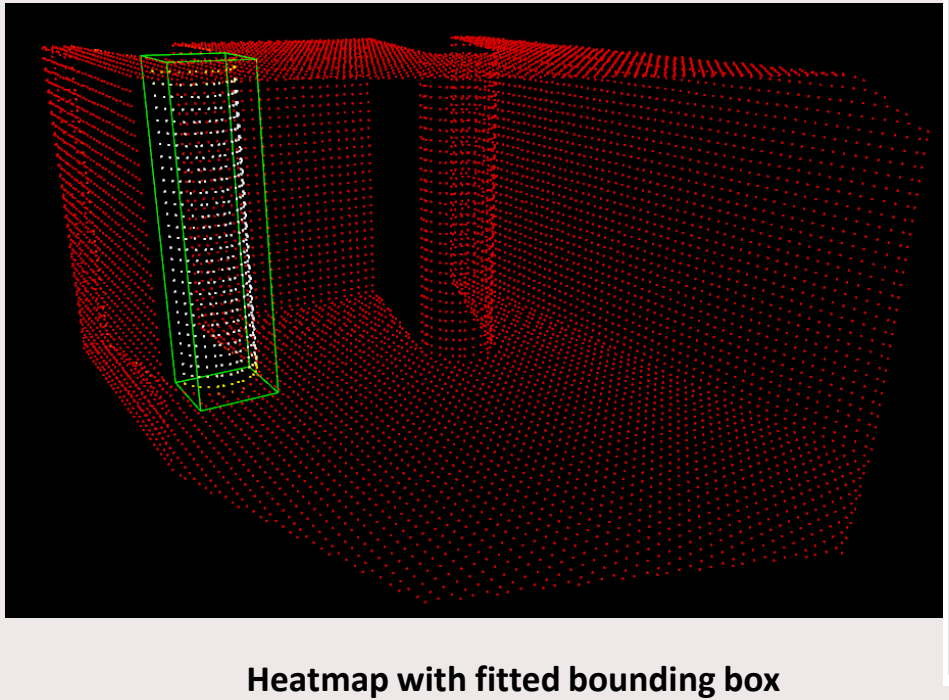
Scenario-Specific Evaluation on Specialized Models

- Single mismatched object
- Scan selection based on mismatch type
- Inference
- Thresholded outputs
- 95% recall as operating point
- FPR, precision, IoU, AUC

Pillar-Added Model – Clean Case

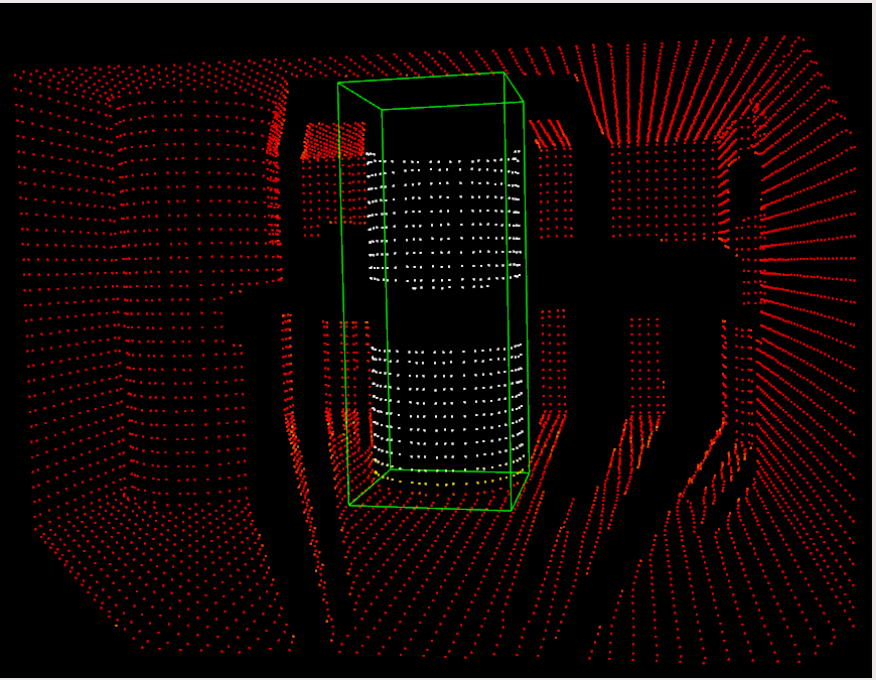


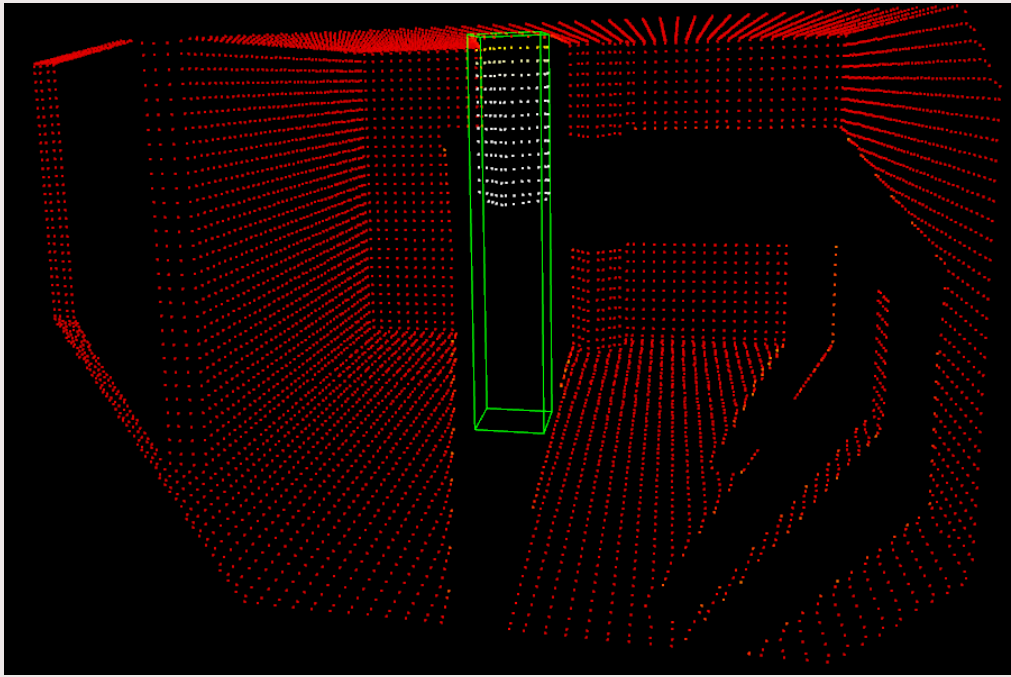
Classification map



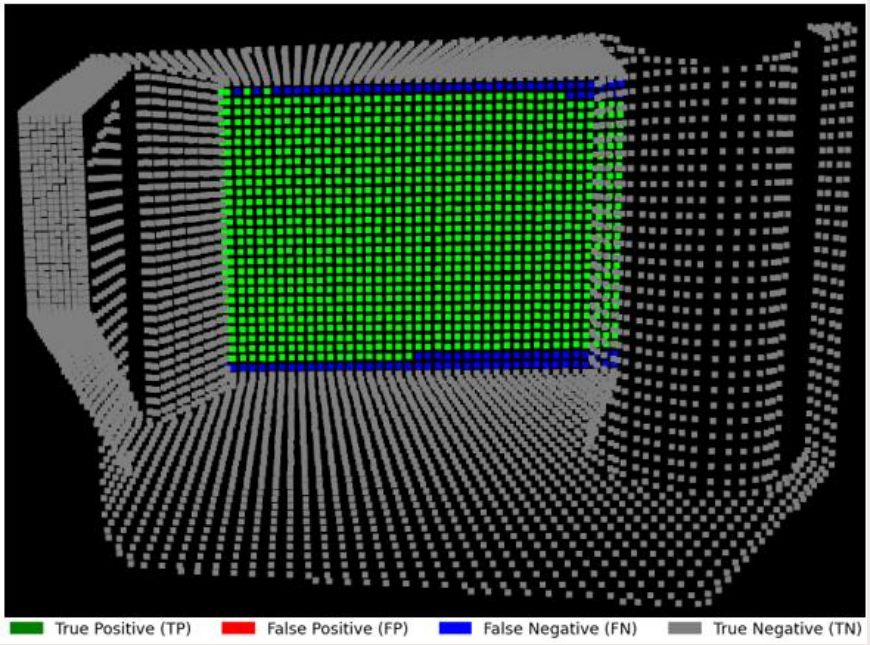
Heatmap with fitted bounding box

Pillar-Added Model – Occluded Case

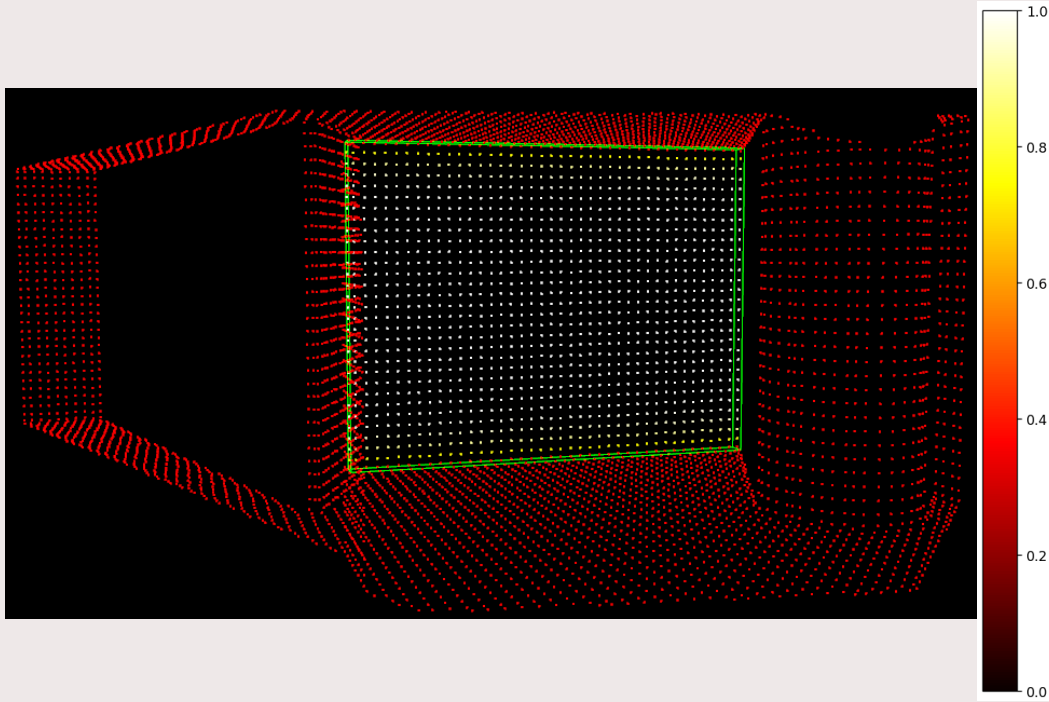




Wall-Added Model – Clean Case

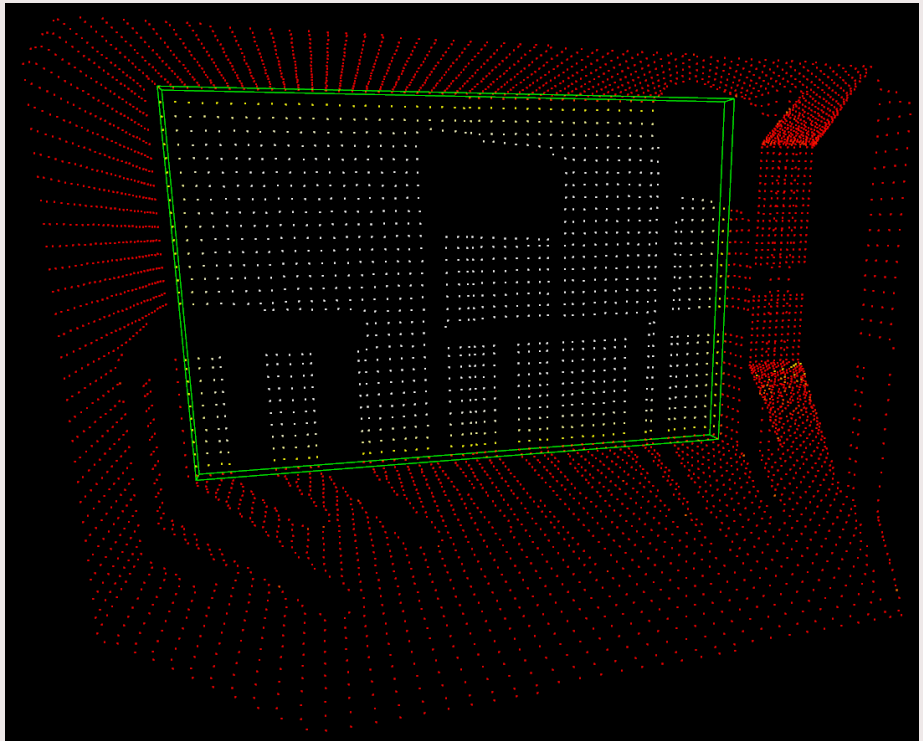


Classification map

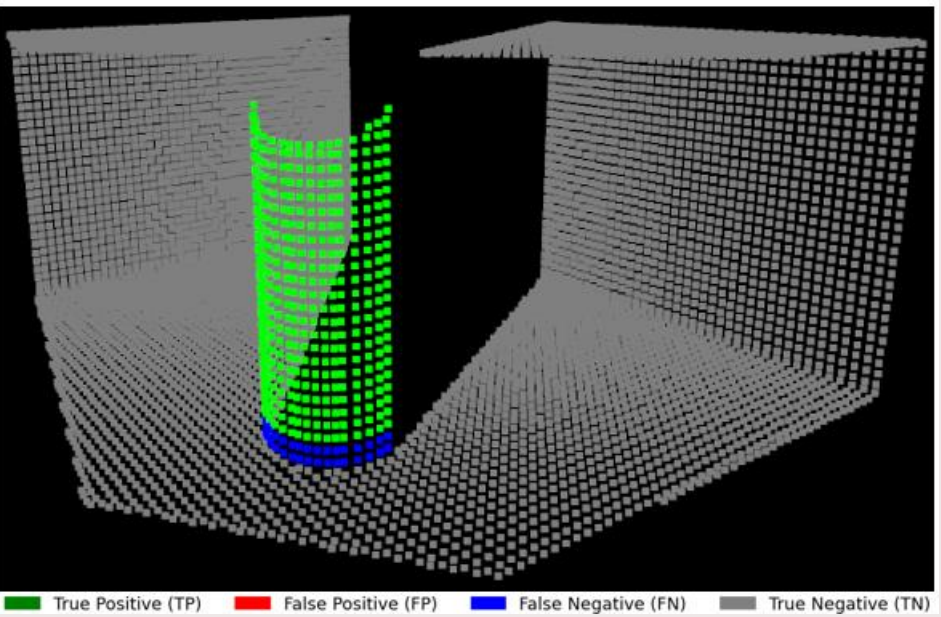


Heatmap with fitted bounding box

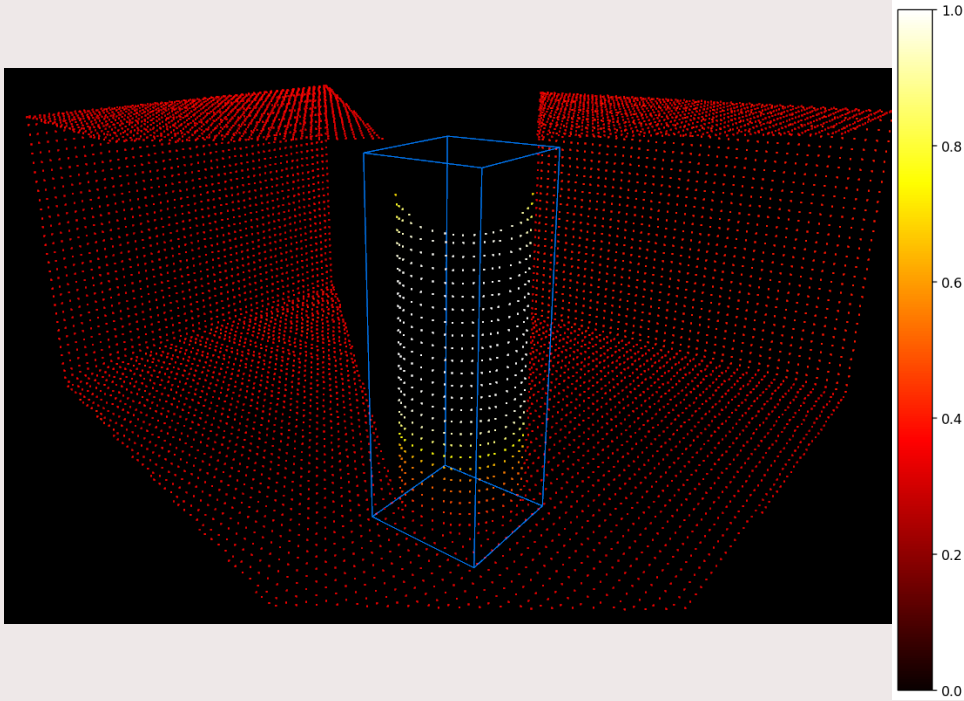
Wall-Added Model – Occluded Case



Pillar-Removed Model – Clean Case

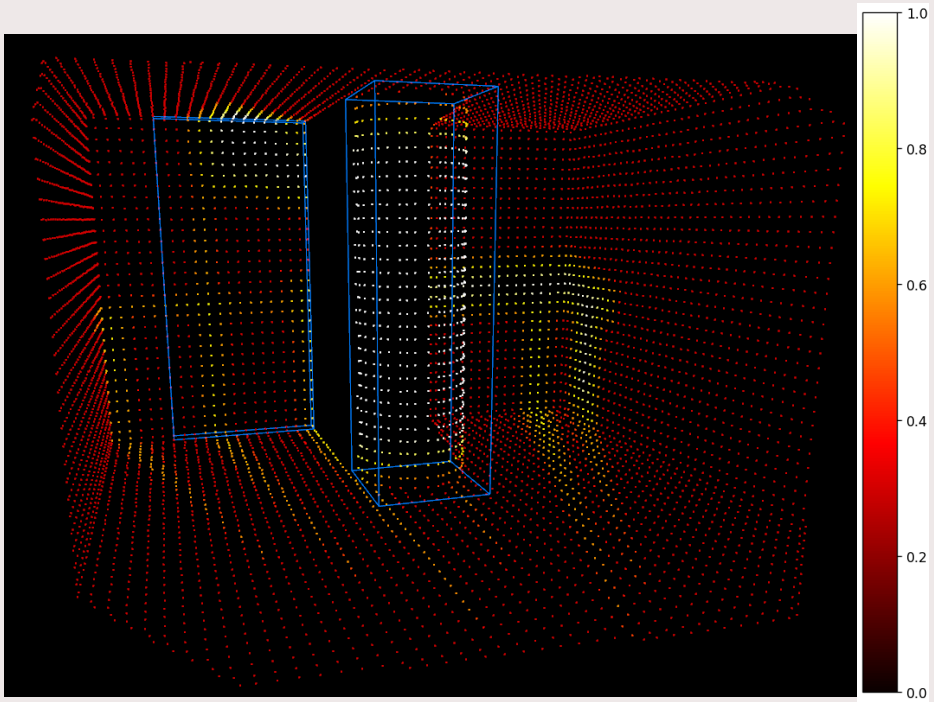


Classification map

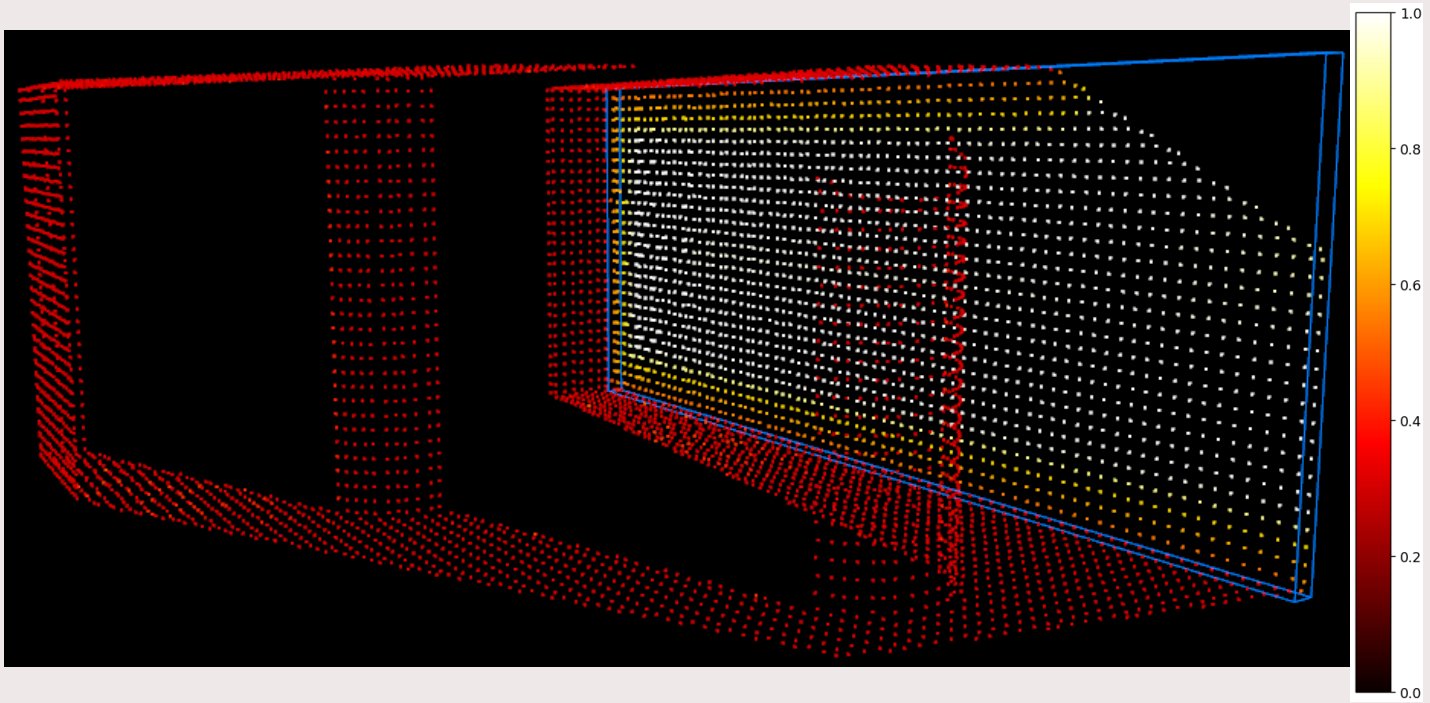


Heatmap

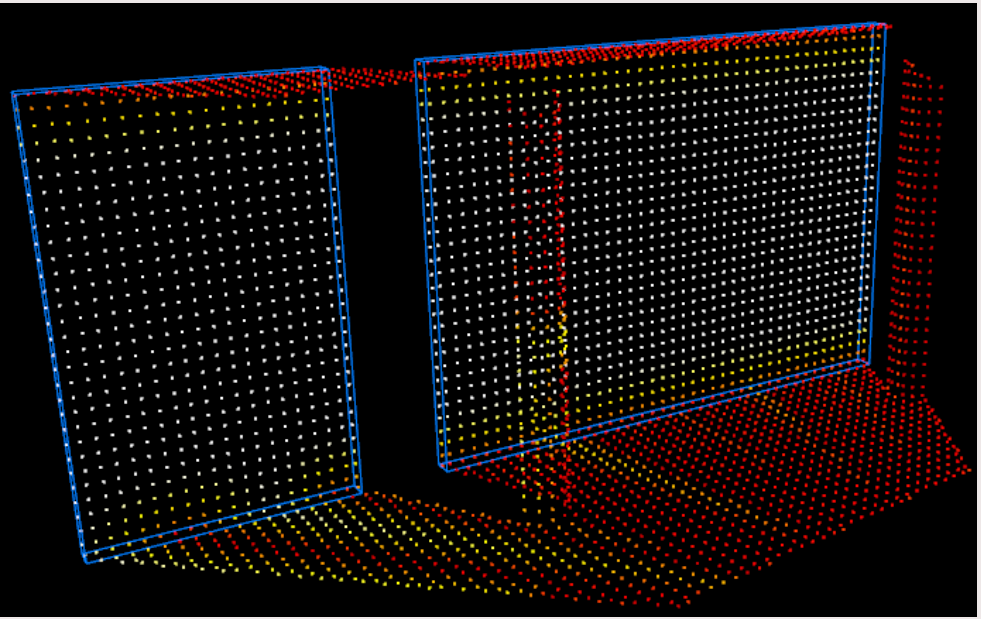
Pillar-Removed Model – Occluded Case

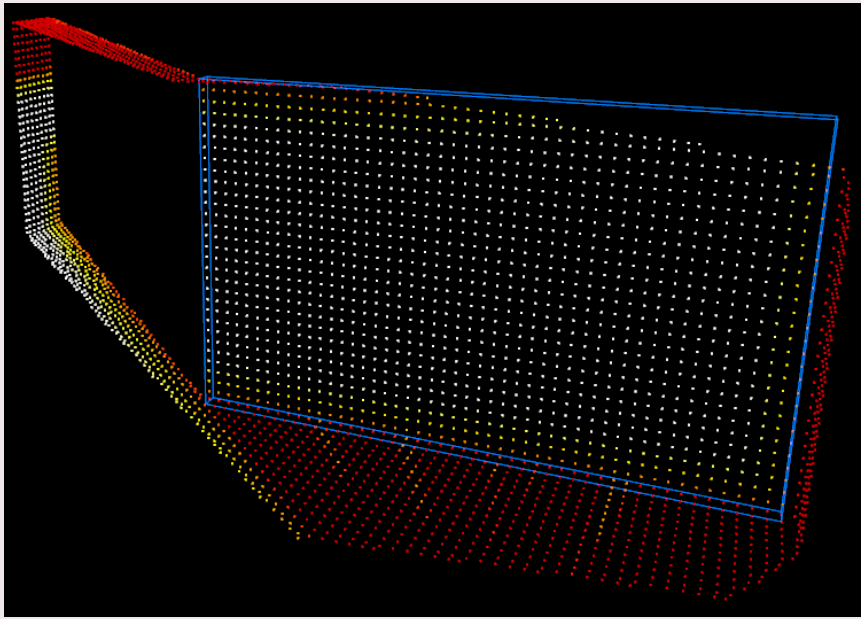


Wall-Removed Model – Clean Case



Wall-Removed Model – Occluded Case

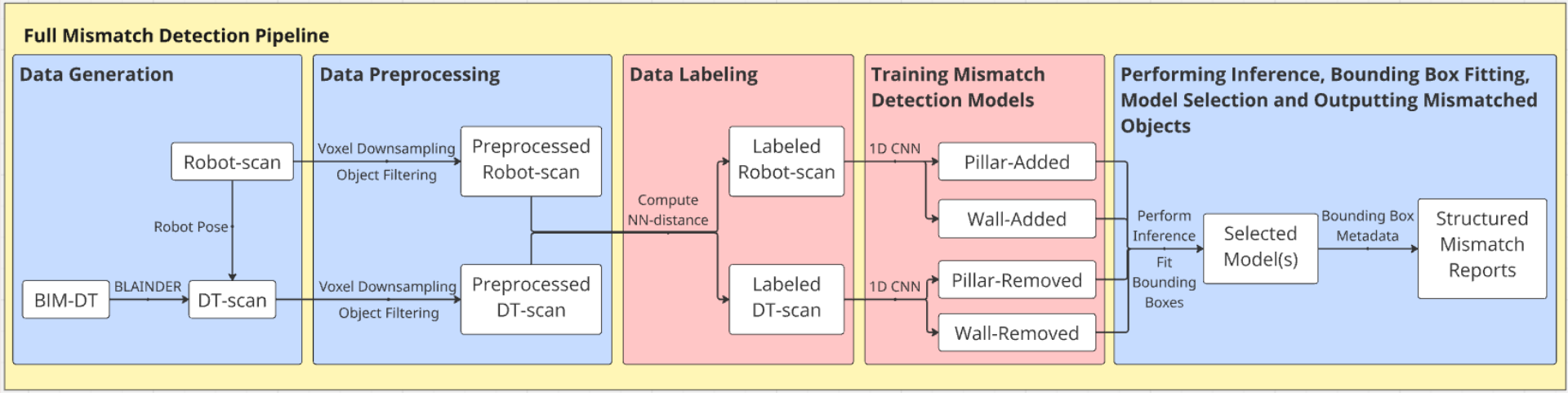




Ablation Studies

- NN-distance is essential
- Coordinates provide spatial context
- Mixed data is needed
- Scan alignment is required
- Data augmentation only introduces noise
- Sensor noise barely hurts performance

Phase 5 – Model Selection and Mismatch Reporting



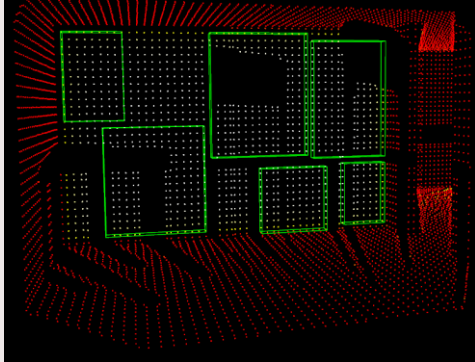
Full Pipeline Model Selection

- Run all models in parallel
- Fixed threshold
- Model selected if bounding box is fitted
- Clusters of mismatched points
- Bounding box shape is guided by model type

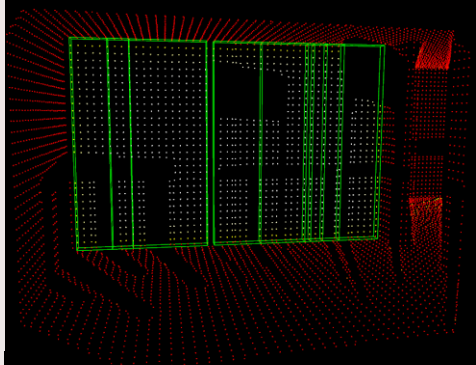
Full Pipeline Model Selection

- Manhattan world assumption
- Vertical stretching
- Assumed wall thickness
- Duplicate boxes
- Merge overlapping wall boxes
- Trim all floor and ceiling points

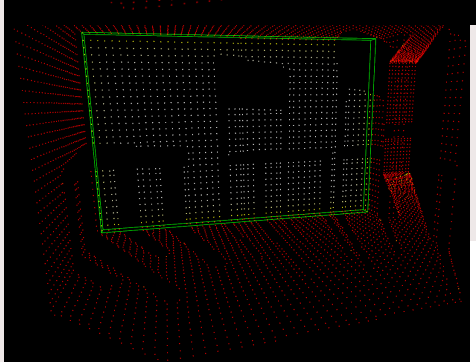
First boxes



Vertically stretched



Merged



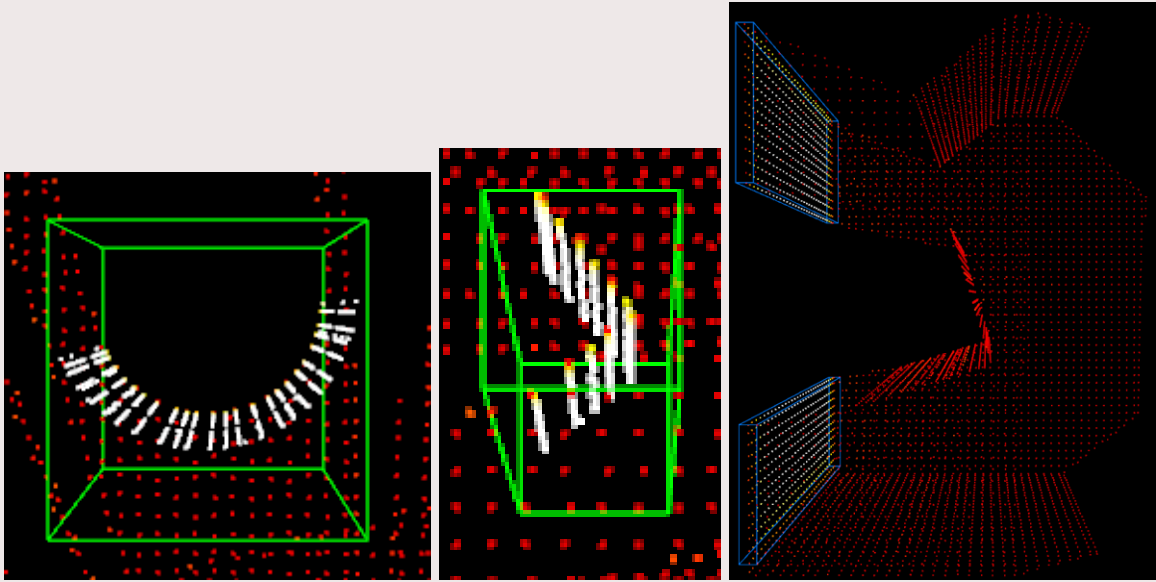
Full Pipeline Model Selection

*"A **pillar** has been **added** at **[-0.3 -0.4 0.]** with dimensions **[0.34 0.34 3.]**."*

- Actual DT-updating is not included
- Overlay boxes onto prior DT for validation
- Leverage DT for shape completion
- DT is not ground truth!

Evaluation of Full Pipeline with Model Selection

- Visibility constraints
- Geometrically reasonable
- Offset pillar center
- Offset pillar orientation
- Partial wall boxes

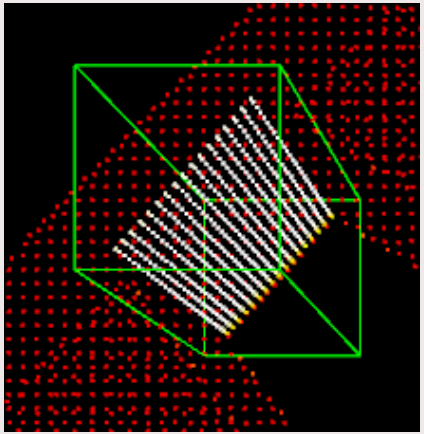


Offset center **Imperfect orientation** **Partial boxes**

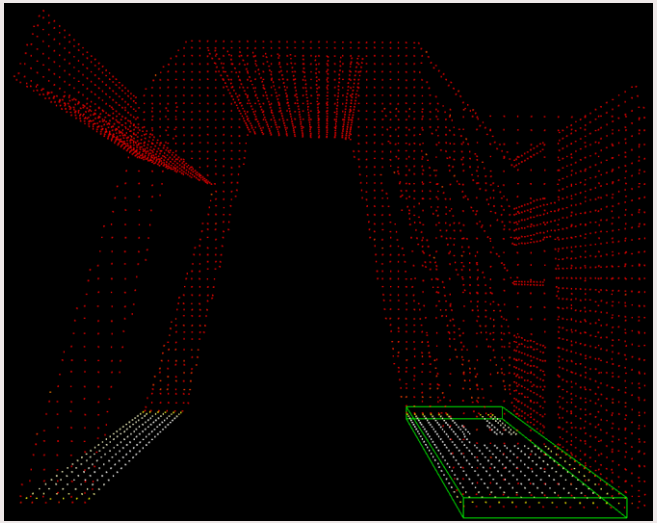
TU/e

Evaluation of Full Pipeline with Model Selection

- Shadow artefact box
- Wall-looking pillar
- Split wall



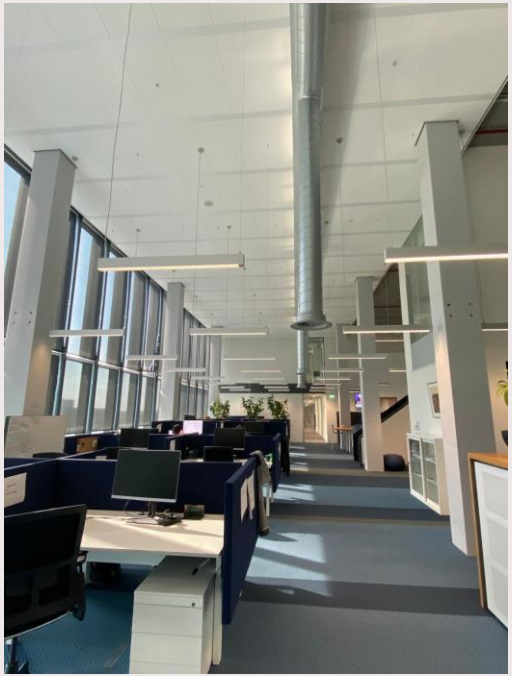
Square pillar with one visible side



Split wall with only one box

Semi-Real-World Validation

- IFC model
- Real deployment



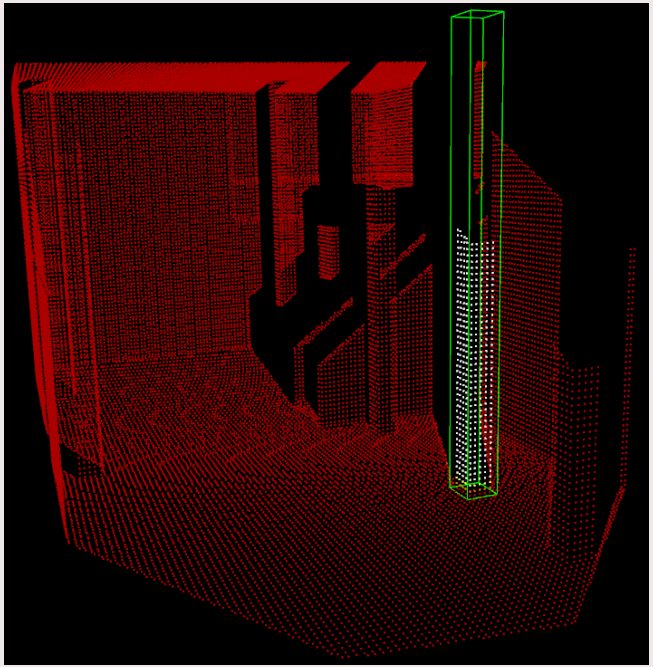
Atlas-8 real world

Atlas-8 BIM

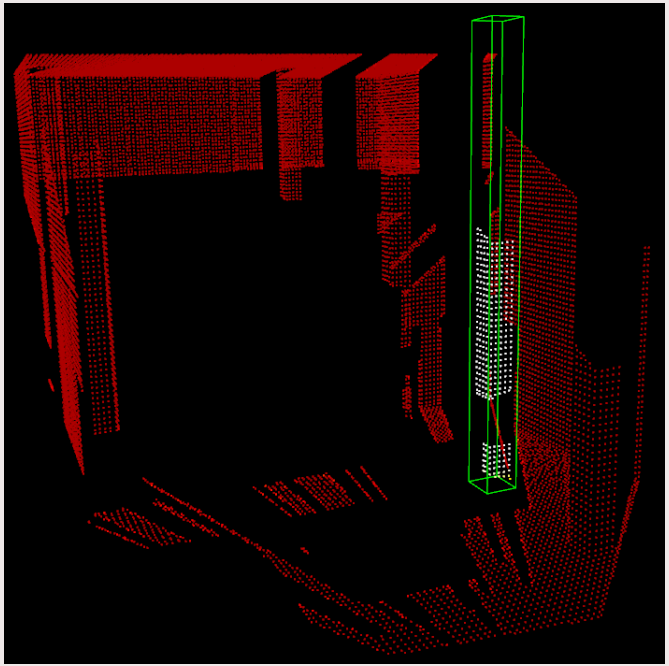
Semi-Real-World Validation Results

- Initial failure
- Global coordinates as features
- Retrained all models using only NN feature
- All models succeed

Semi-Real-World Validation – Pillar-Added Model

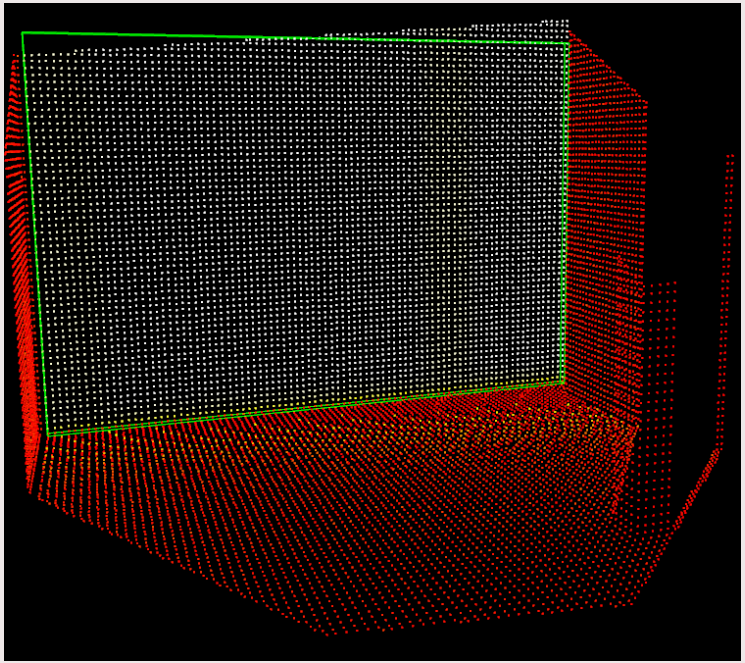


Clean case

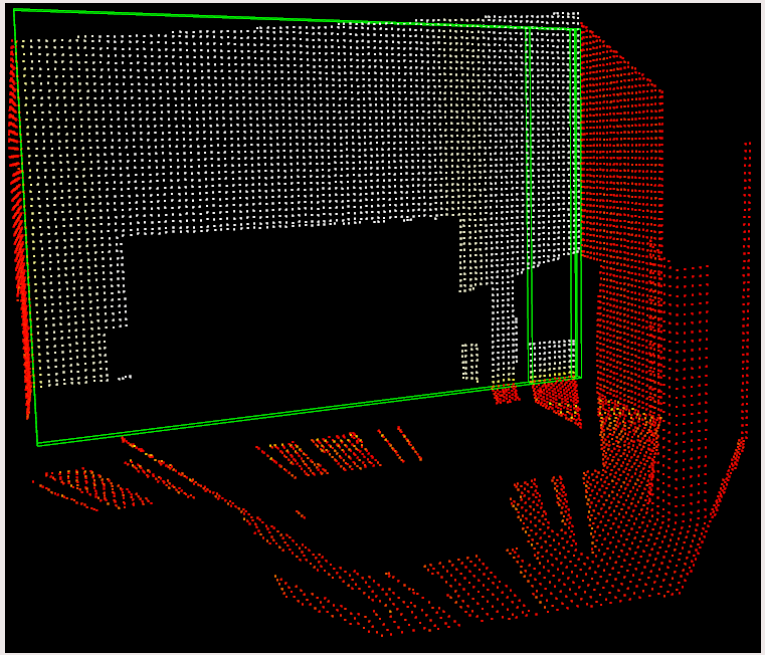


Occluded case

Semi-Real-World Validation – Wall-Added Model

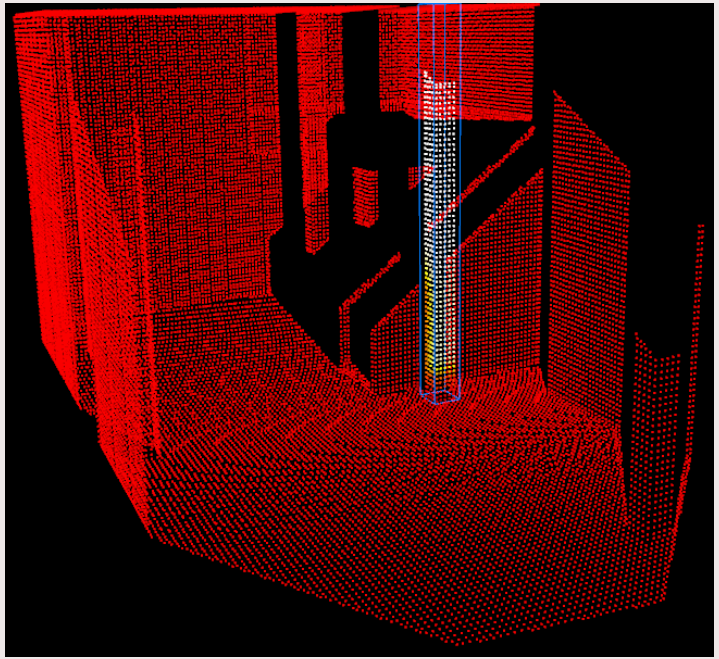


Clean case

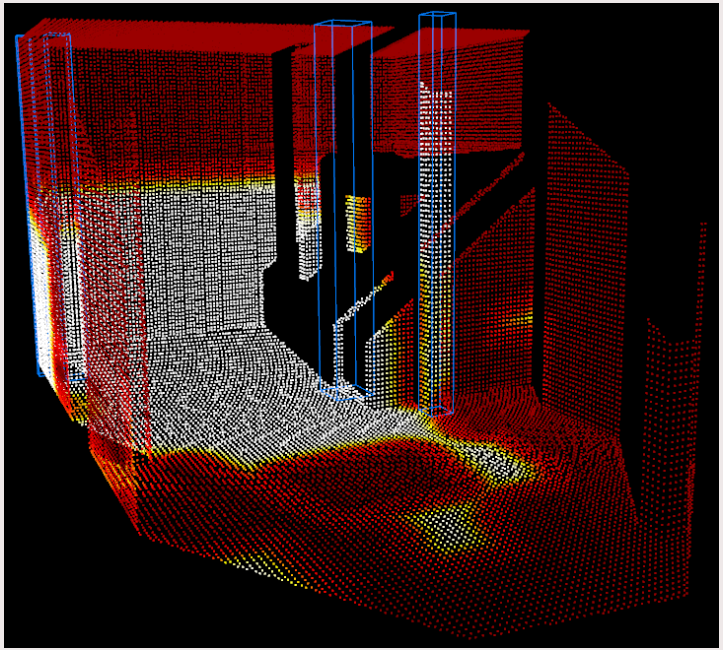


Occluded case

Semi-Real-World Validation – Pillar-Removed Model

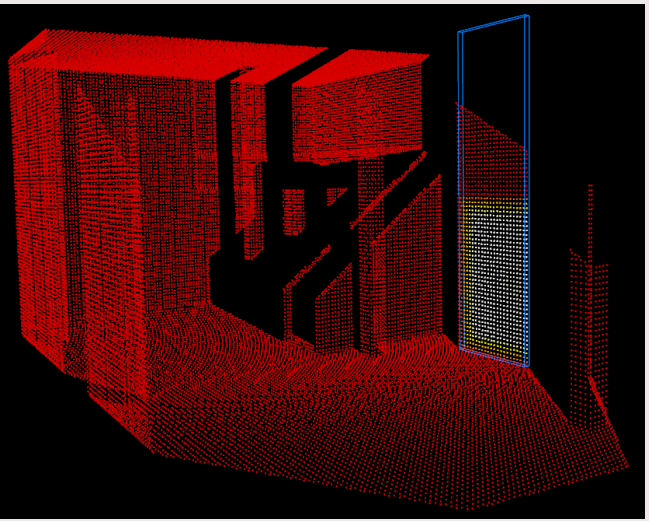


Clean case

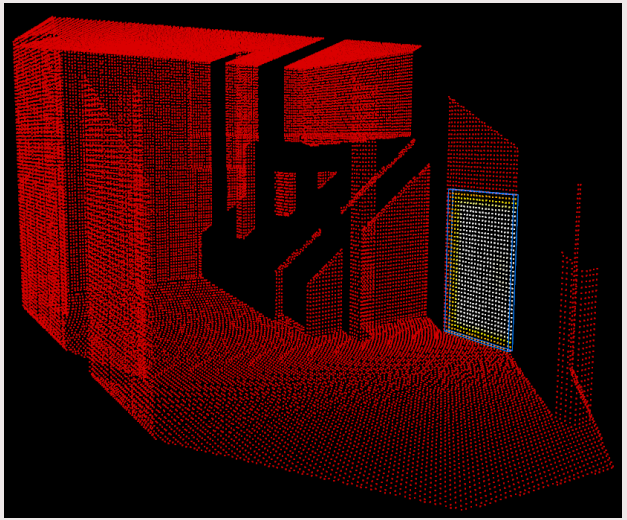


Occluded case

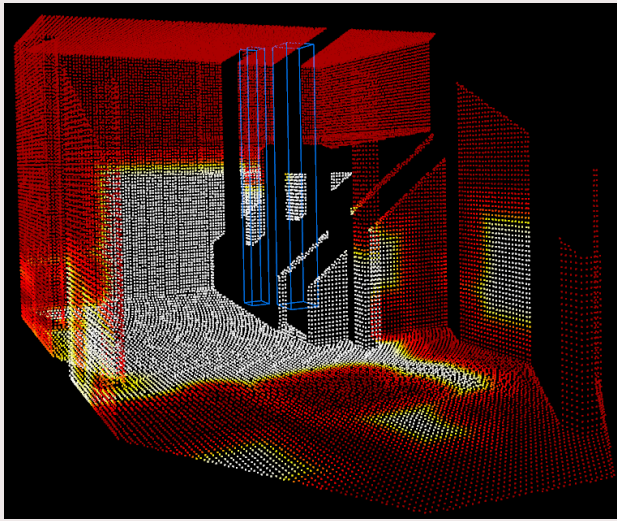
Semi-Real-World Validation – Pillar-Removed Model



Clean case with vertically stretched box



Clean case without vertically stretched box



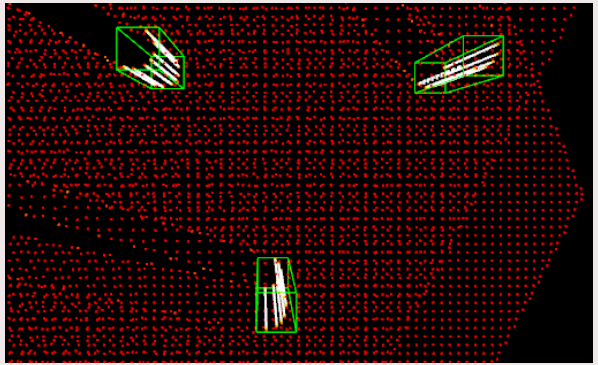
Occluded case

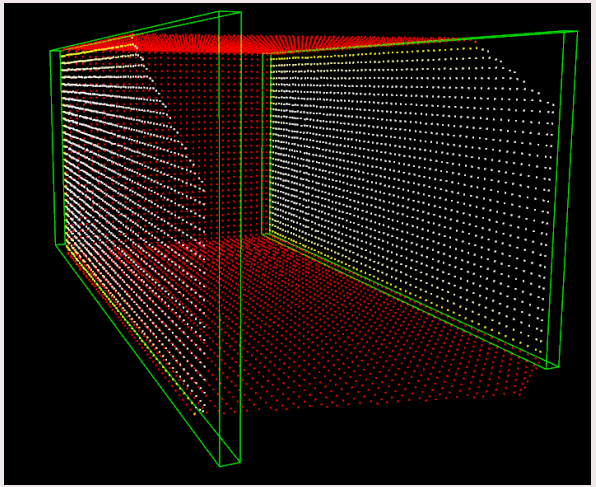
Multi-Mismatch Scenario Evaluation

- Multiple objects of the same type and mismatch(e.g., several removed pillars)
- Multiple objects of different types but the same mismatch (e.g., walls and pillars added)
- Multiple mismatch types across the same object class (e.g., one pillar removed, another added)
- Fully mixed scenes with several types and mismatch categories

Same Object and Mismatch Type - Added Objects

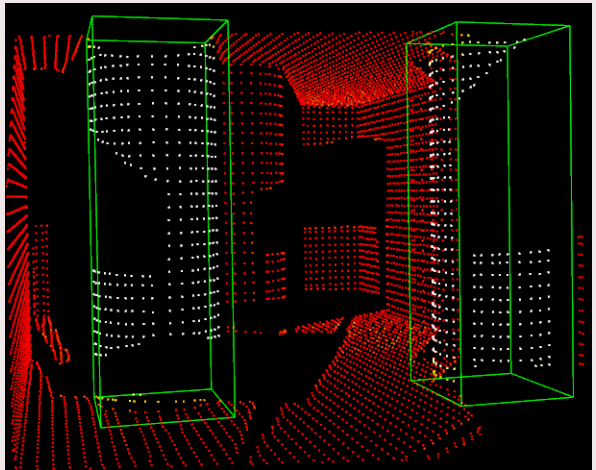
Clean pillars-added case

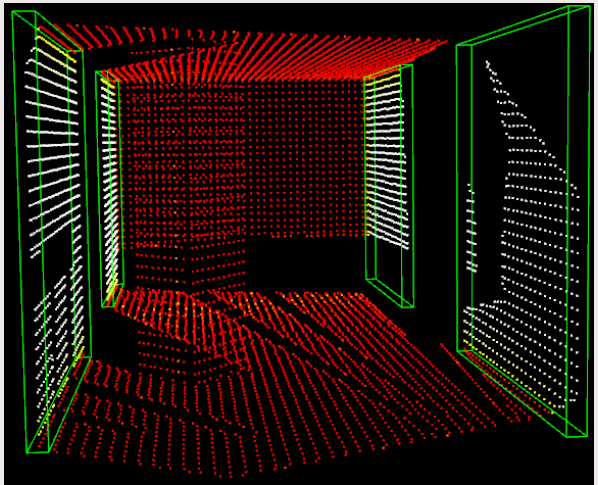




Clean walls-added case

Occluded pillars-added case

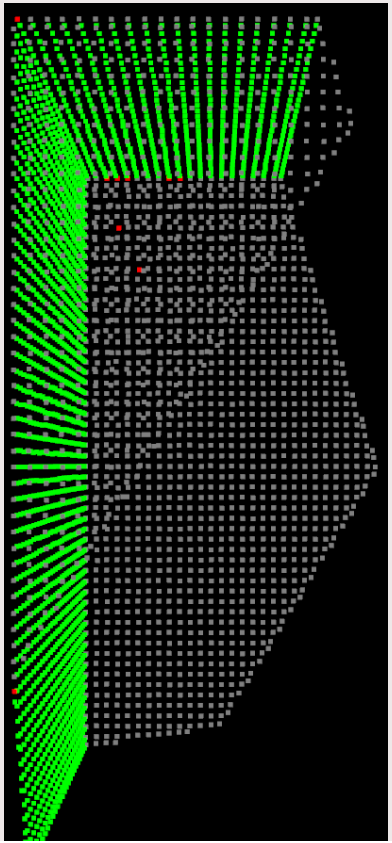




Occluded walls-added case

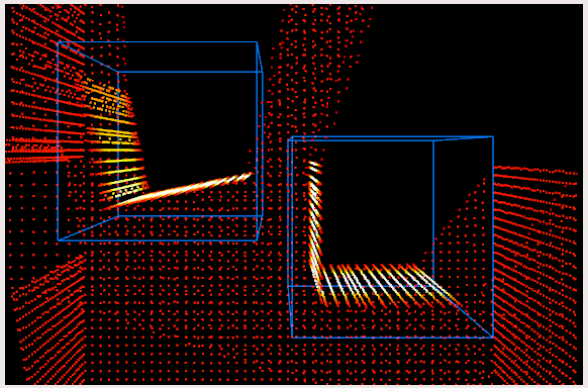
TU/e

Connected Mismatched Walls

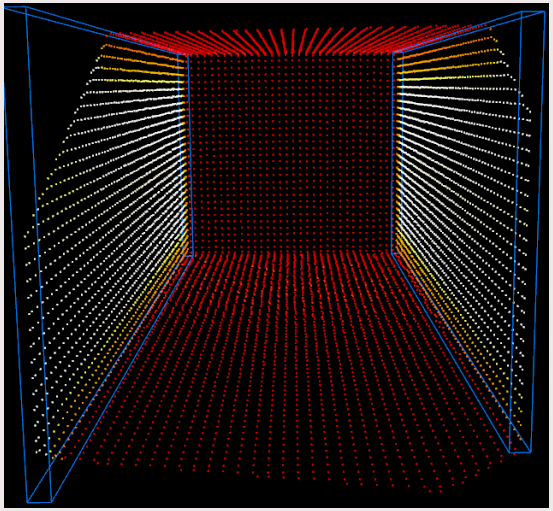


Same Object and Mismatch Type - Removed Objects

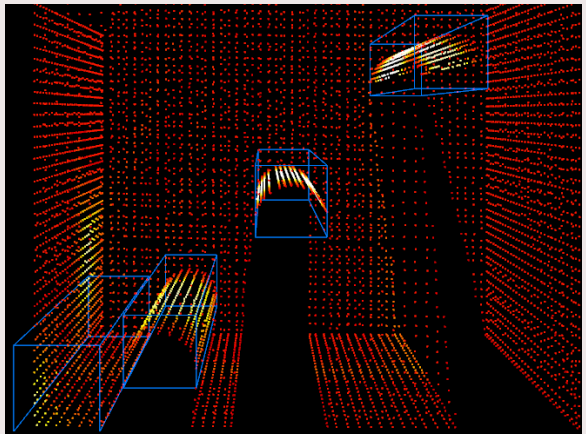
Clean pillars-removed case



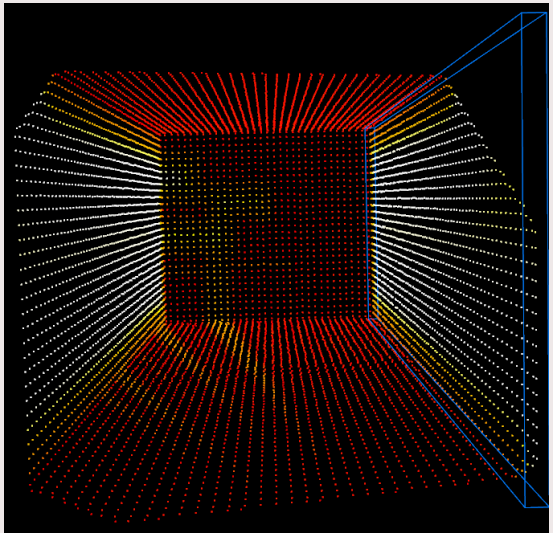
Clean walls-removed case



Occluded pillars-removed case



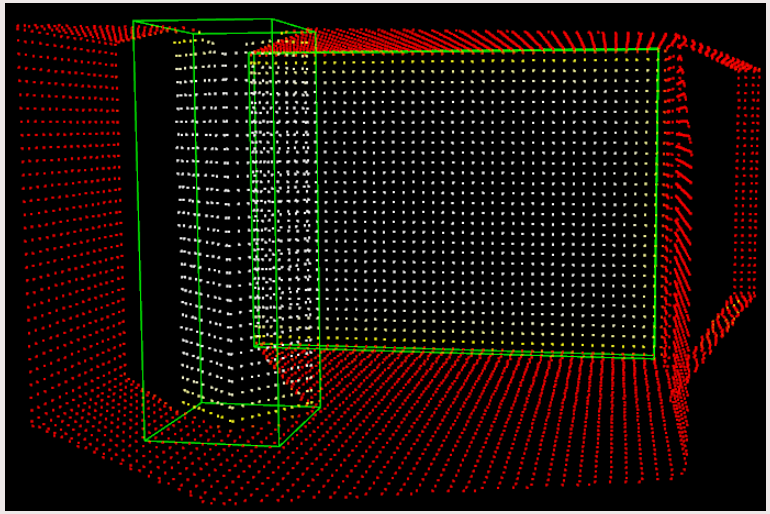
Occluded walls-removed case

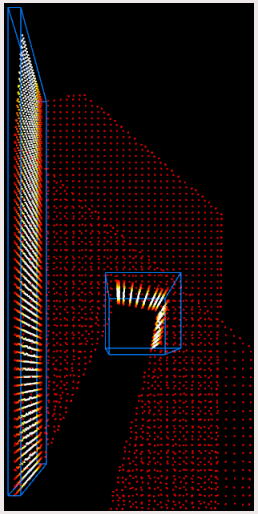


TU/e

Different Object Type and Same Mismatch Type

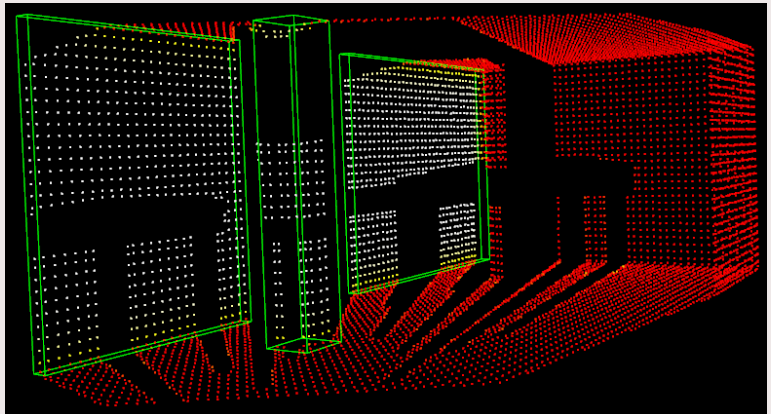
Clean objects-added case

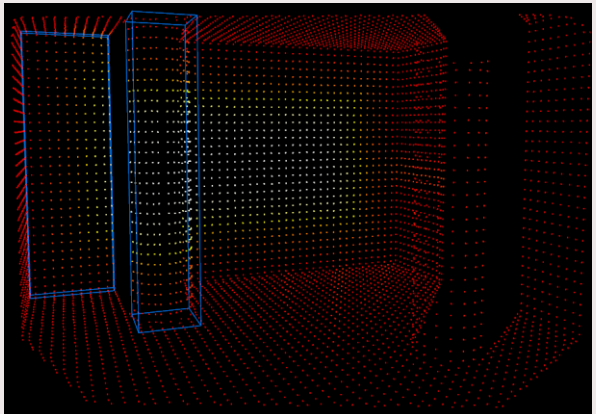




Clean objects-removed case

Occluded objects-added case

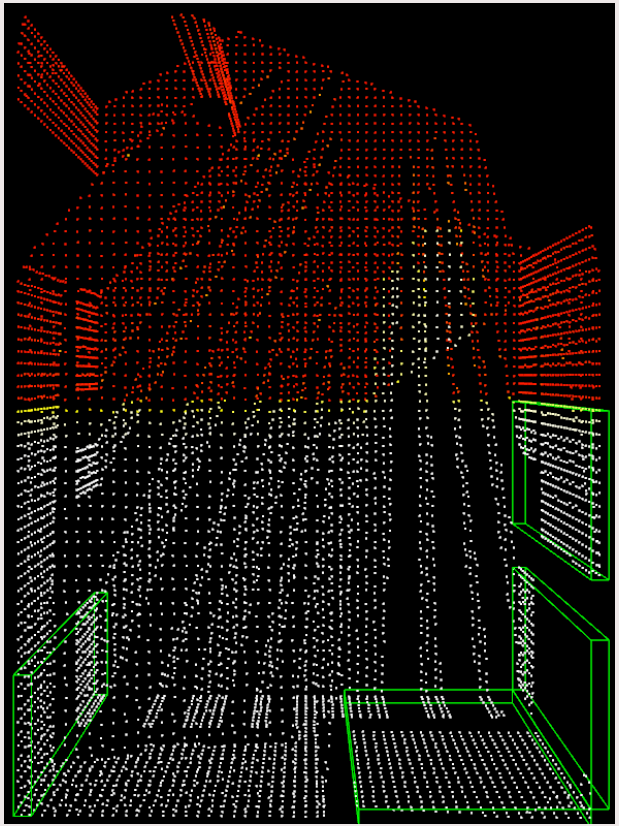




Occluded objects-removed case

TU/e

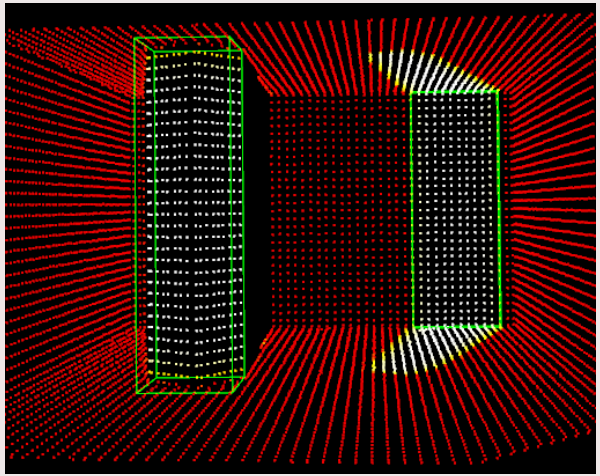
Different Object Type and Same Mismatch Type

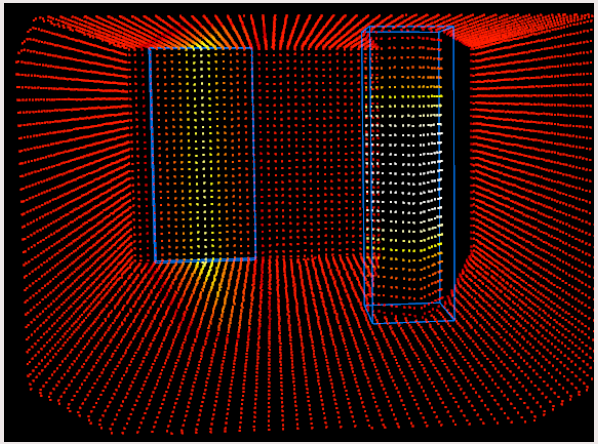


Robot scan – Removed wall reveals “new” room

Same Object and Different Mismatch Type

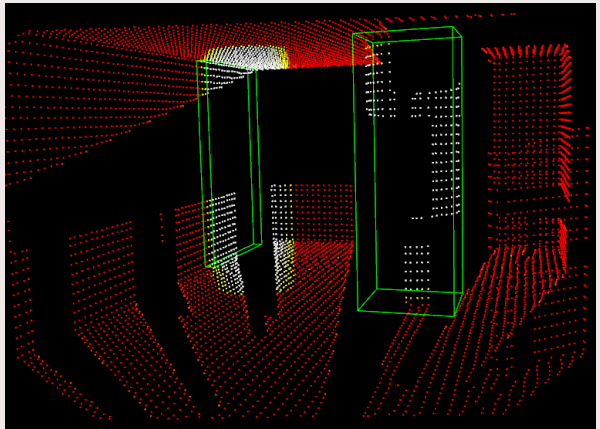
Clean robot scan pillars mismatched

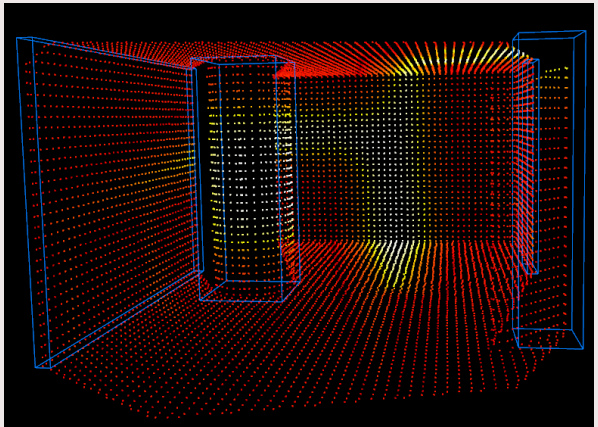




Clean DT scan pillars mismatched

Occluded robot scan pillars mismatched



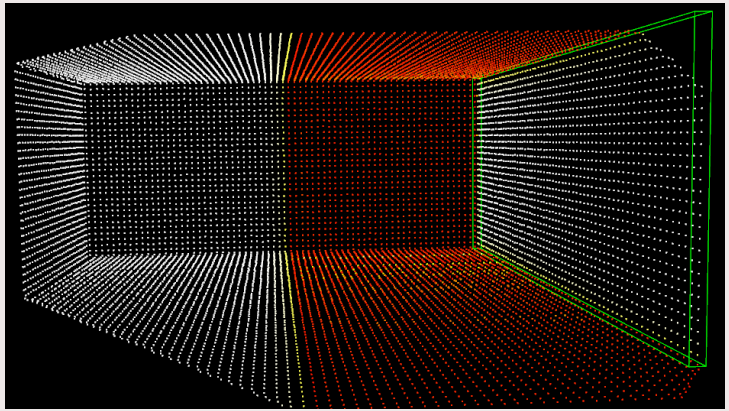


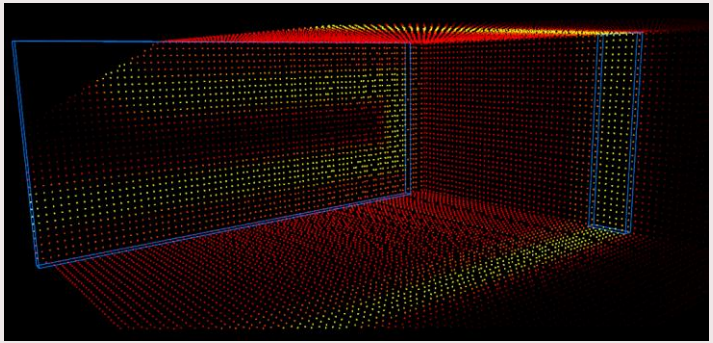
Occluded DT scan pillars mismatched

TU/e

Same Object and Different Mismatch Type

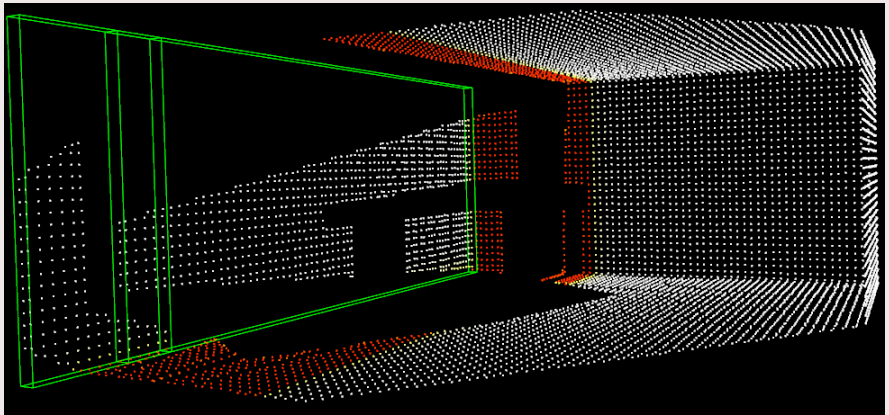
Clean robot scan walls mismatched

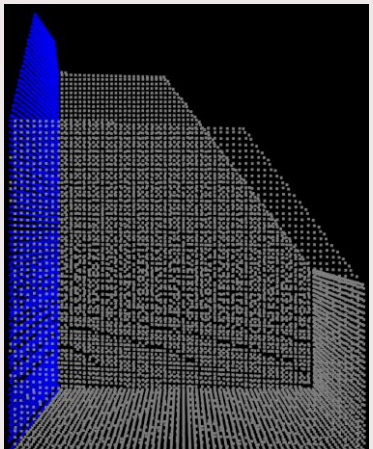




Clean DT scan walls mismatched

Occluded robot scan walls mismatched



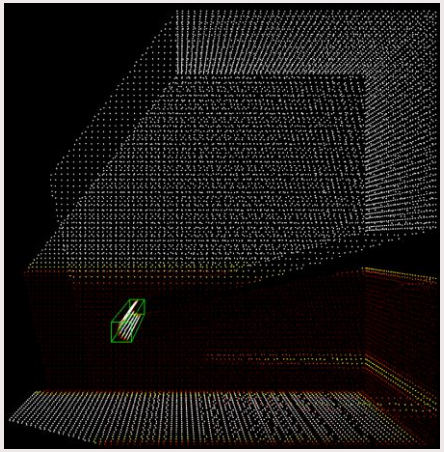


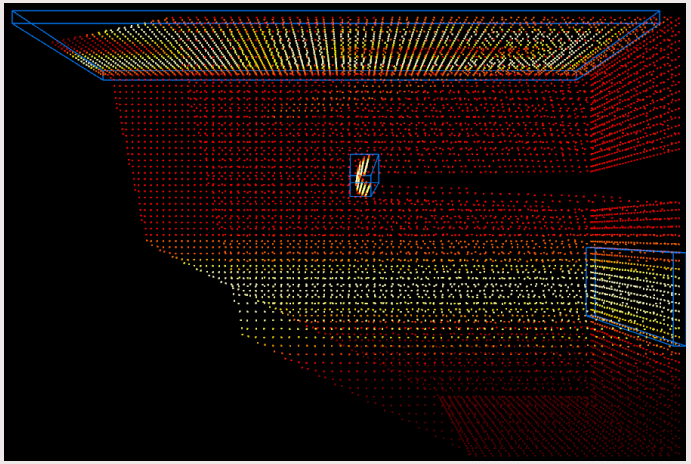
Occluded DT scan walls mismatched

TU/e

Different Object Type and Mismatch Type

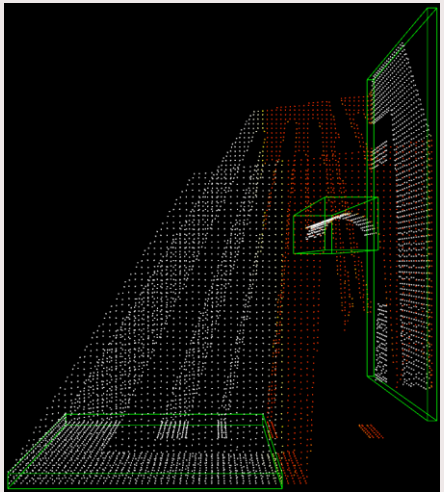
Clean robot scan objects mismatched

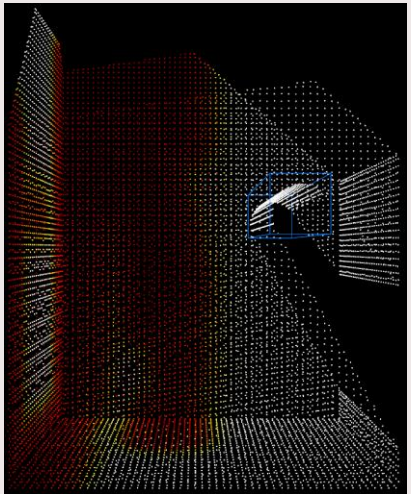




Clean DT scan objects mismatched

Occluded robot scan objects mismatched





Occluded DT scan objects mismatched

TU/e

Conclusion

- Mismatch detection
- Bounding box fitting
- Lightweight
- Multi-view fusion
- Uncertainty scoring system
- Real-world noise
- Direct digital twin updating

Questions?

Appendix - Research Questions

"How can structural mismatches in indoor environments be detected using per-point predictions from 3D scan pairs generated from a single simulated viewpoint?"

- *How can a fully synthetic data pipeline be designed to generate high-quality, labeled 3D scan pairs for training mismatch detection models?*
- *Can simple, lightweight neural network models trained on synthetic data generalize to unseen scenes and identify different types of structural changes?*
- *What is an effective way to validate mismatch detection in scenarios where no ground truth is available?*
- *How can we evaluate and select predictions from multiple specialized models when the true mismatch type is unknown?*
- *How do the trained mismatch detection models generalize to multi-mismatch scenes?*
- *How can we create a bridge from mismatch detection to the updating of a BIM?*

Appendix: Export Format and Data Structure

- BLAINDER .csv files
- Sensor noise initially disabled
- Converted to .npy

Appendix: 1D Convolutional Feature Extraction

- Four 1D convolutional layers
- ReLU
- Batch normalization
- Global max pooling
- Concatenated with local features

Appendix: Global Feature Pooling

- Summarize most dominant features

$$\mathbf{g} = \max_{i=1}^{N} \mathbf{f}_i$$

Appendix: Segmentation Head

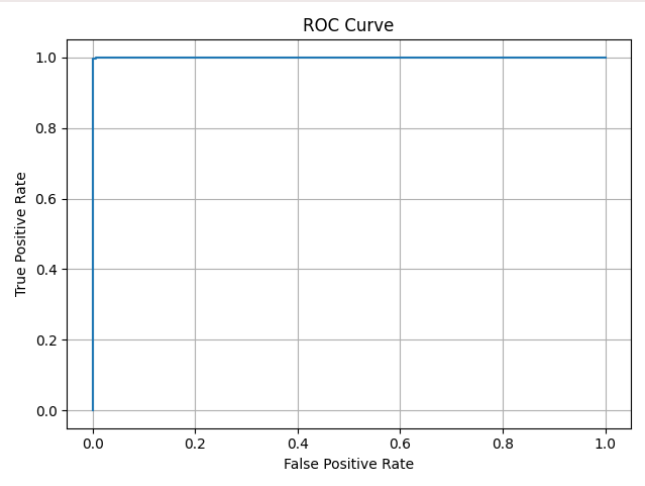
- Per-point classifications
- Sigmoid

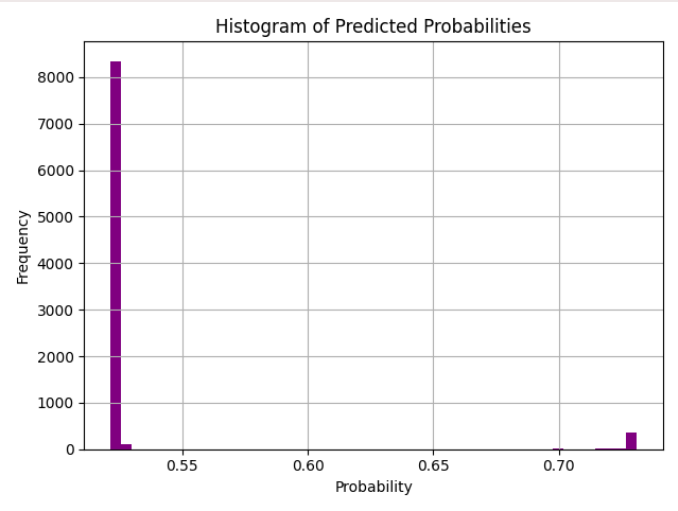
Appendix: Loss Function

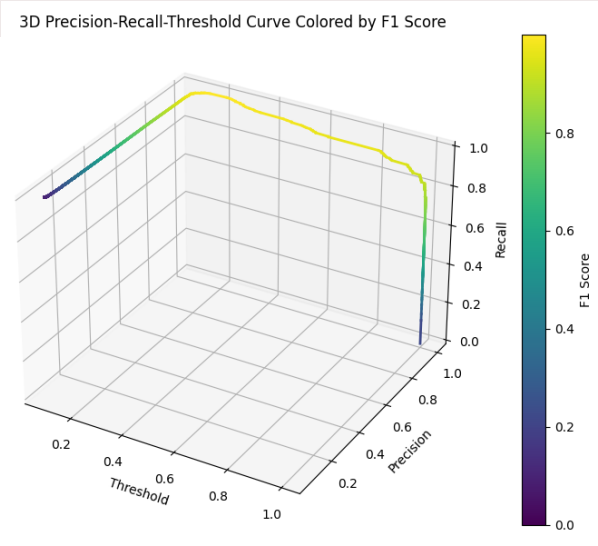
- Binary Focal Loss
- Class imbalance
- α increases the penalty for false negatives
- γ reduces the influence of well-classified examples

$$L = -\alpha(1-\hat{y}_i)^{\gamma} \cdot y_i \log(\hat{y}_i) - (1-\alpha)\hat{y}_i^{\gamma} \cdot (1-y_i) \log(1-\hat{y}_i)$$

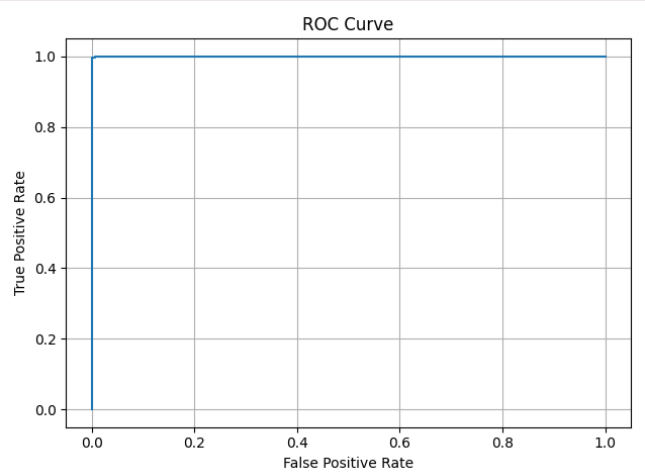
Appendix: Pillar-Added Model – Clean Case

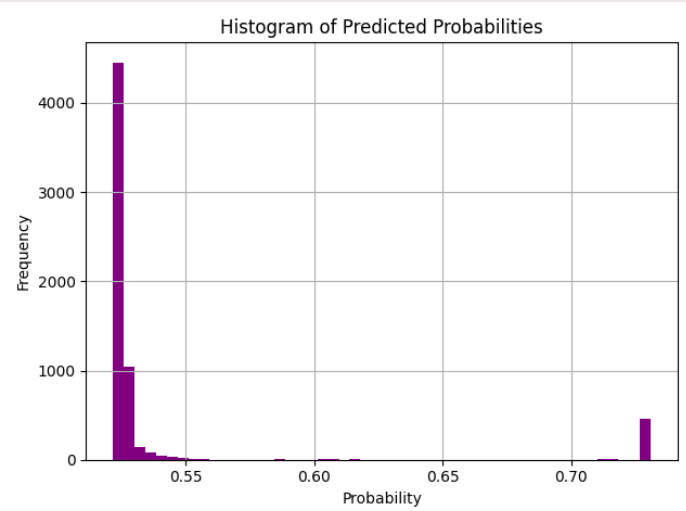


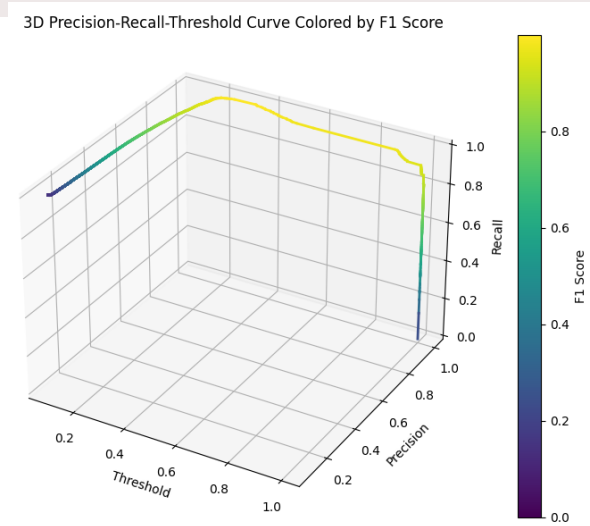




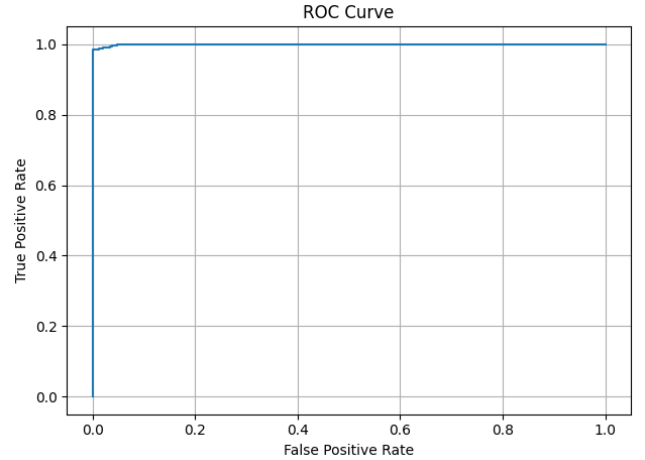
Appendix: Pillar-Added Model – Occluded Case

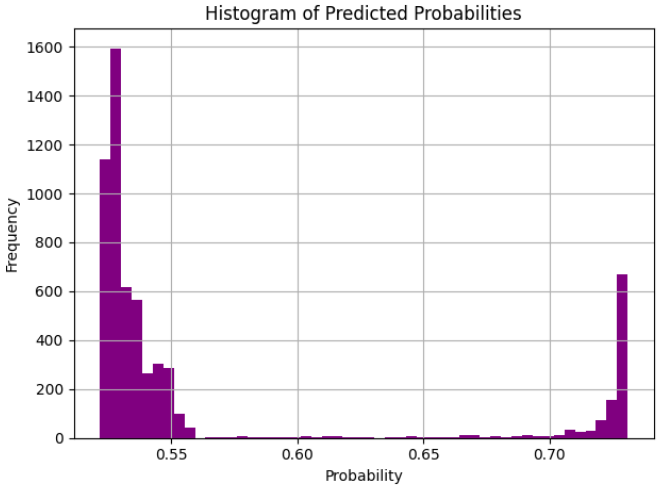


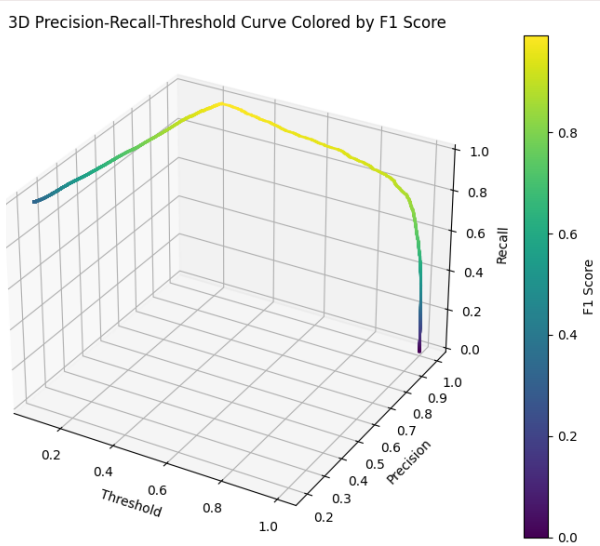




Appendix: Wall-Added Model – Clean Case

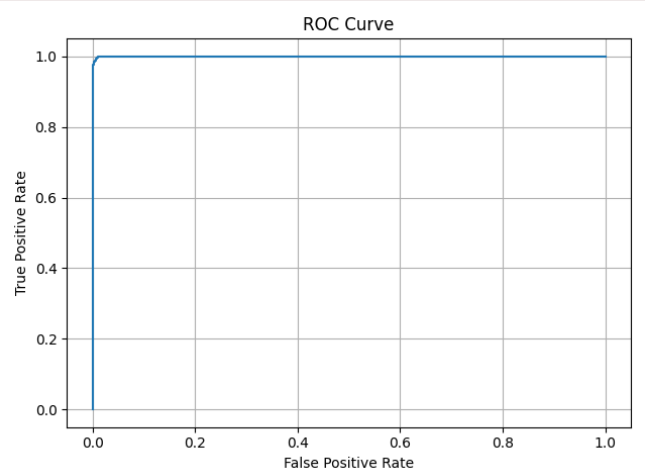


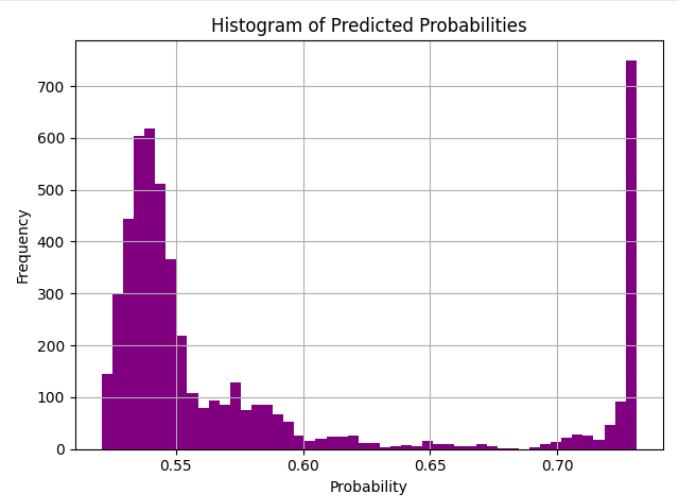


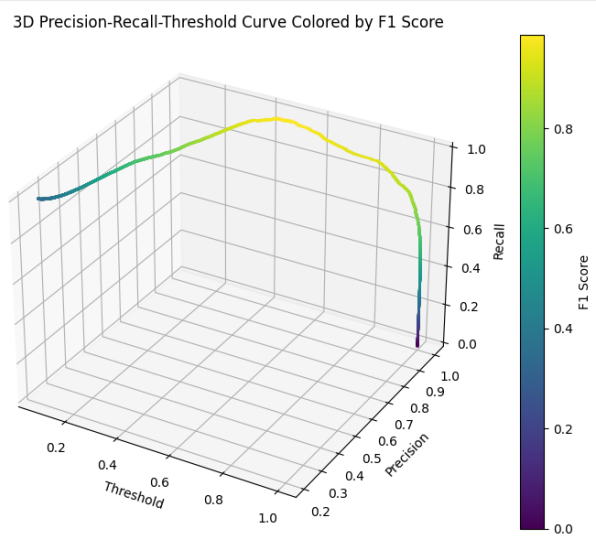


TU/e

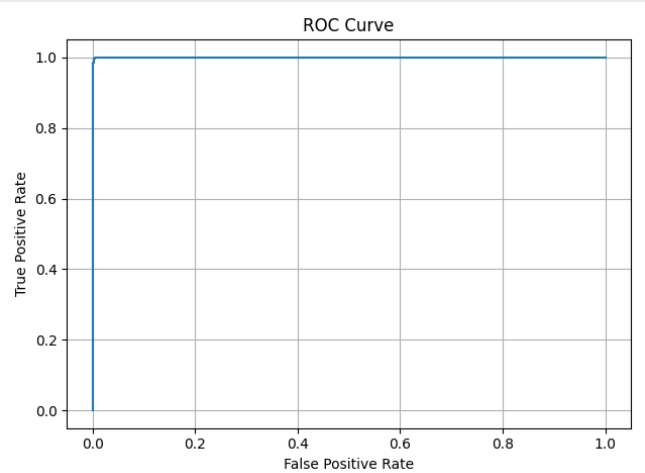
Appendix: Wall-Added Model – Occluded Case

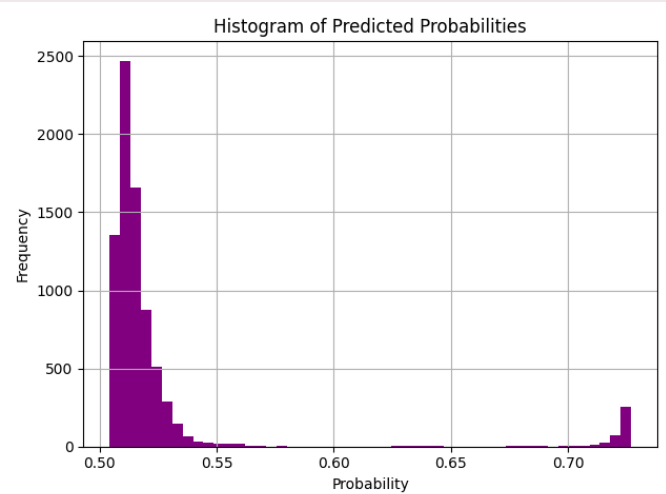


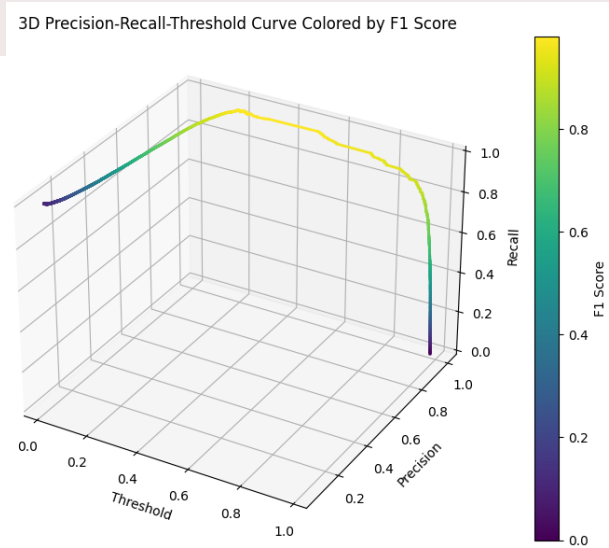




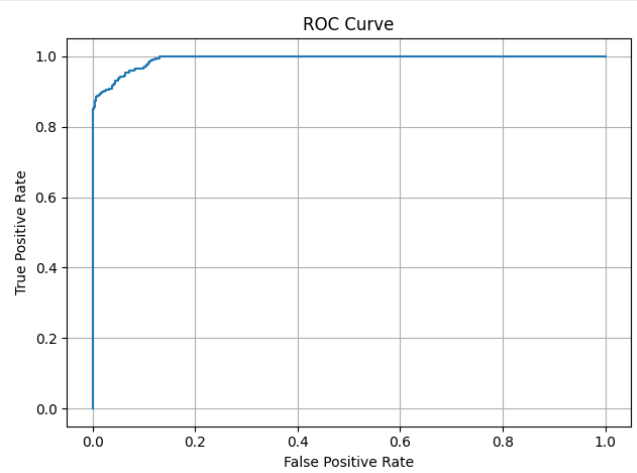
Appendix: Pillar-Removed Model – Clean Case

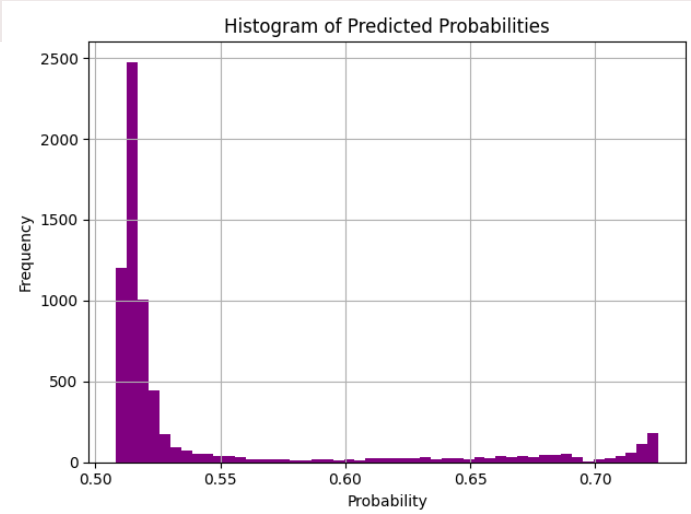


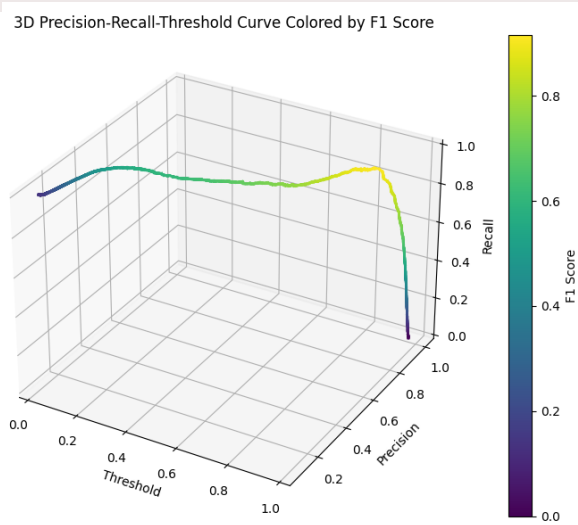




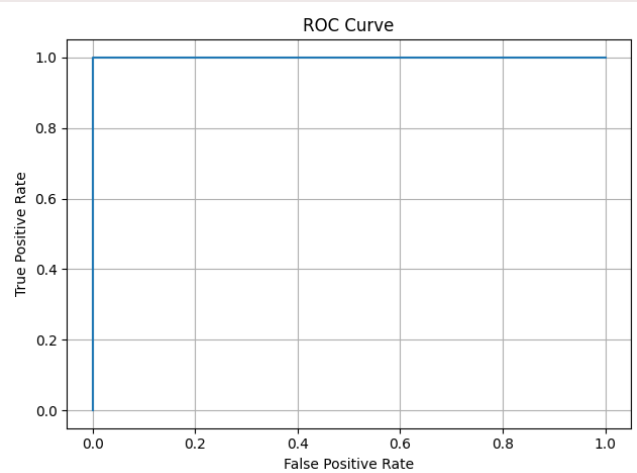
Appendix: Pillar-Removed Model – Occluded Case

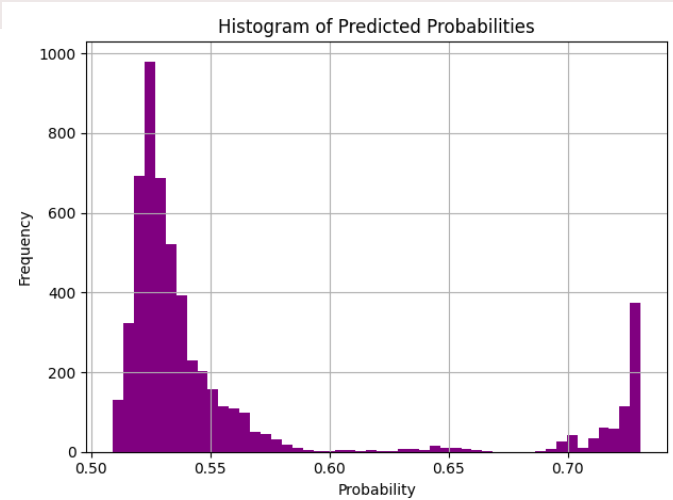


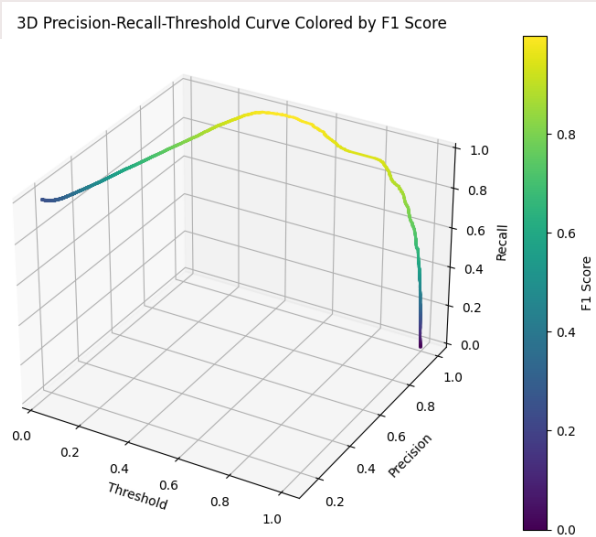




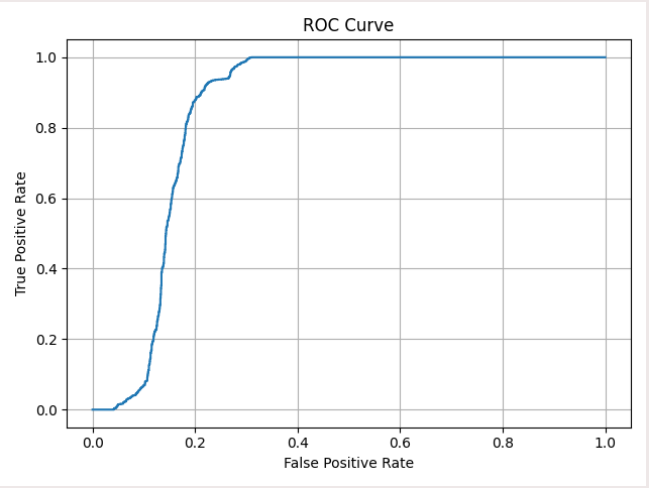
Appendix: Wall-Removed Model – Clean Case

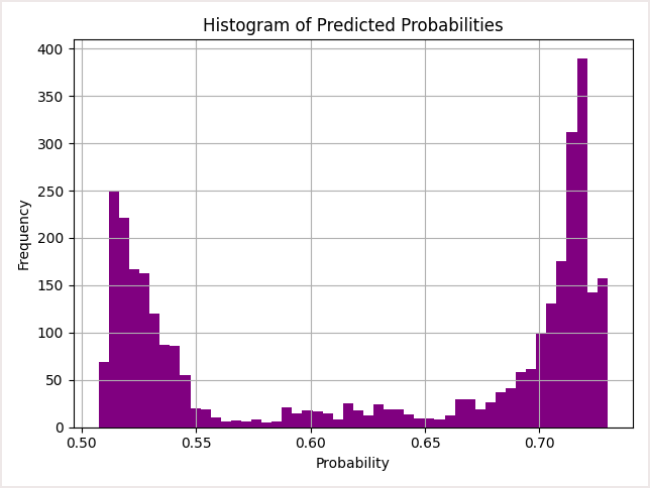


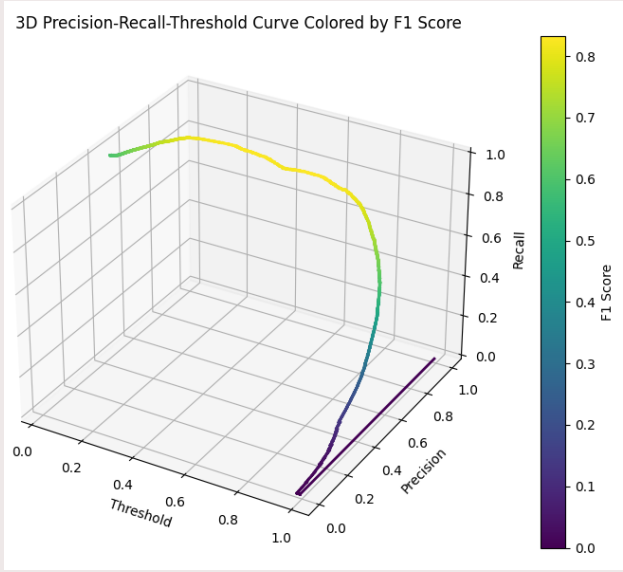




Appendix: Wall-Removed Model – Occluded Case







Appendix: List of Limitations

- Actual digital twin updating has not been implemented
- False bounding boxes around shadow artefacts or the absence of a fitted bounding box around a mismatched object due to shadow artefact infection
- Bounding boxes are only fitted to the visible portion of a mismatched object
- Bounding box overgrowth, where clusters that span multiple nearby structures can result in boxes that are too large to be plausible.
- The bounding box fitting stage remains object-type specific, since different shapes (e.g., pillars vs. walls) can possibly require different geometric assumptions.
- Models trained with global XYZ coordinates tended to overfit to the spatial bounds of the synthetic rooms
- This approach does not score or measure uncertainty between models like the mismatch detection part does
- The challenge of using only a single viewpoint, which makes it hard to detect geometric mismatches when parts of the scene are not visible.
- The experiments did not account for real-world noise, such as surface reflectivity or specific lighting conditions
- Only walls and pillars are considered as mismatched object types in this work
- Only the addition or removal of such objects is implemented and tested
- Per-point classification without explicit object-level context can lead to fragmented or incomplete mismatch predictions

Appendix: List of Future Work Ideas

- Actual updating of a digital twin could be done in future work
- An alternative to using global coordinates as features alongside the nearest-neighbour distance feature could be to use relative coordinates, such as positions centered around the object cluster, room centroid or the 3D sensor.
- Future work could benefit from object-centric models that learn to localize and describe entire mismatched objects more directly, even when only a small portion of the object is visible, reducing reliance on bounding box fitting.
- Update bounding boxes multiple times using multiple scans from different viewpoints, allowing a more complete reconstruction of the object's geometry over time.
- A more principled model selection strategy such as bounding box confidence or adding a learned selector model could improve pipeline robustness.
- One possible future extension to reduce these shadow artefacts is to apply interpolation techniques during preprocessing.
- The inclusion of domain adaptation techniques such as fine-tuning on real-world scans and robust features less sensitive to density and viewpoint.
- Future work could also explore the integration of additional information that is present inside a BIM model, such as material properties and surface colour information, into the mismatch detection pipeline.
- Future work could also deal with a wider variety of occlusion types beyond static furniture like chairs and tables, for example, dynamic objects such as humans may pose different challenges for mismatch detection.