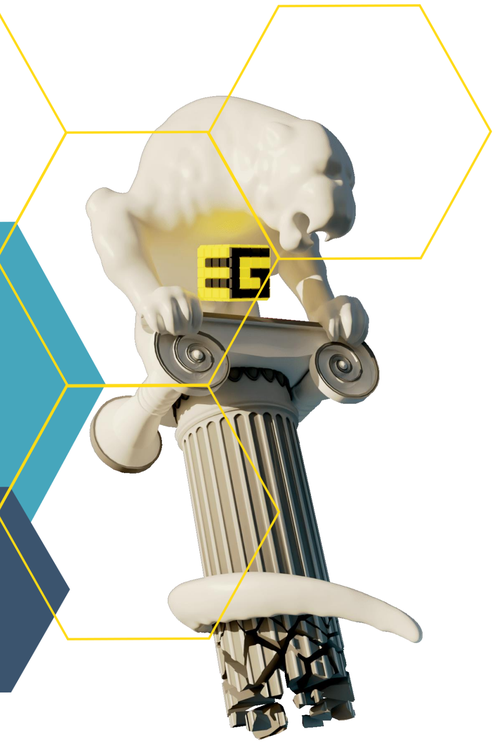
Tetrahedron-Tetrahedron Intersection and Volume Computation Using Neural Networks



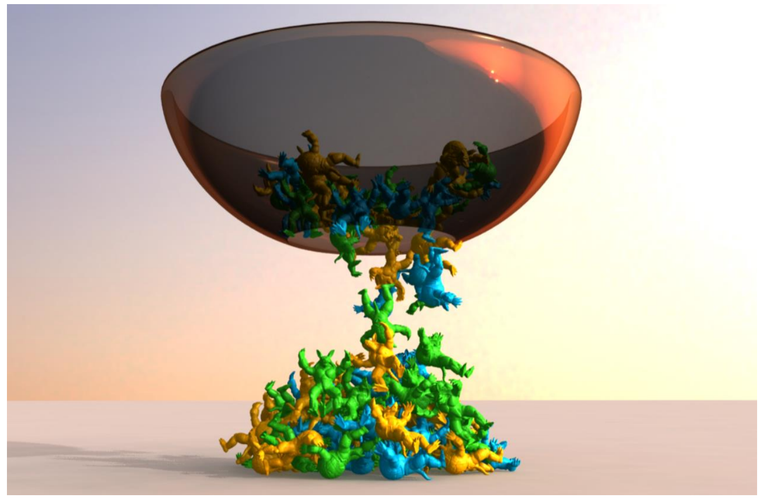
Erendiro Pedro[1], Hermann Meißenhelter[2], Gabriel Zachmann[2]

[1]Institute of Engineering, Polytechnic of Porto, Portugal
[2]University of Bremen, Germany

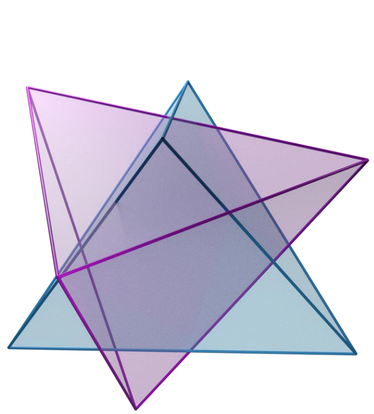
Eurographics 2026, 4-8 May, Aachen, Germany

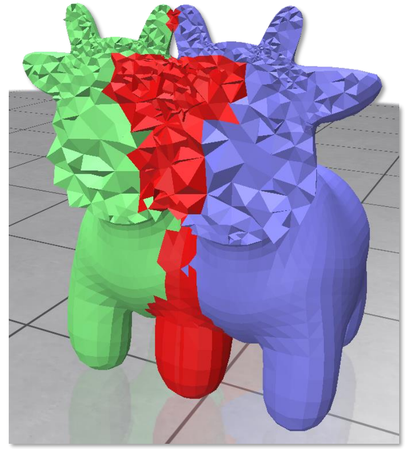
Motivation

- Narrow-phase collision detection is a bottleneck in physics simulation
  - Tetrahedral meshes are common in FEM and deformable simulation
  - May require millions of pairwise checks per frame
- Beyond yes/no collision, many applications need an estimate of intersection extent
- Intersection volume is the “most complicated yet accurate method” for intersection extent [FisherLin01]



Armadillo models, each with 371,000 tetrahedral elements [Bender17]

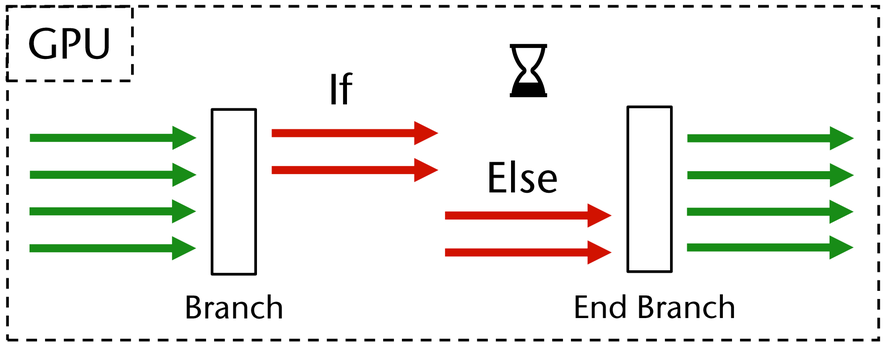




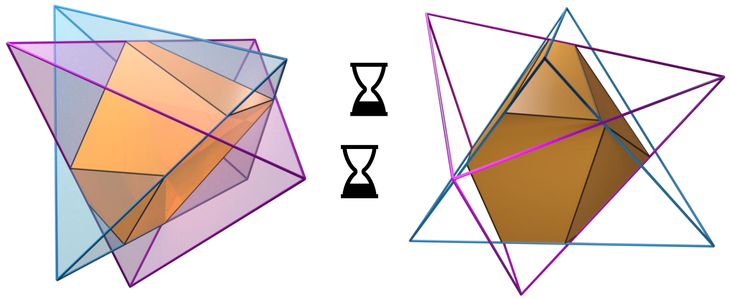
Problem and Gap

- Classical exact methods rely on sequential branching logic
  - This causes thread divergence and poor GPU utilization
- Volume computation is even more expensive than intersection testing
  - Requires constructing the overlap geometry
  - Becomes difficult to use at large scale

```
pair<bool, simplex> gjk(support_func):
...
if dot(cross(normal, C-A), -A) > 0:
    if dot(C-A, -A) > 0:
        simplex = [C,A]
        D = cross(cross(C-A, -A), C-A)
        return
    else:
        simplex = [A]
        D = -A
        return
else
    ...
...
```

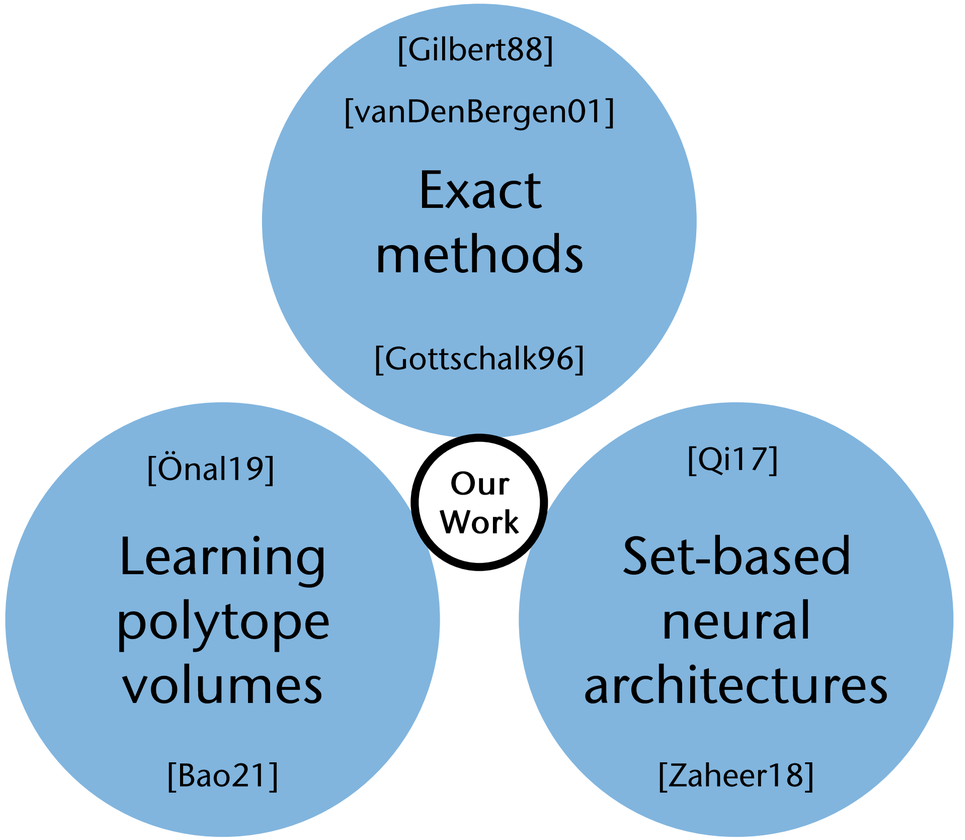


- Question: Can batched neural inference predict tet–tet intersection and overlap volume?



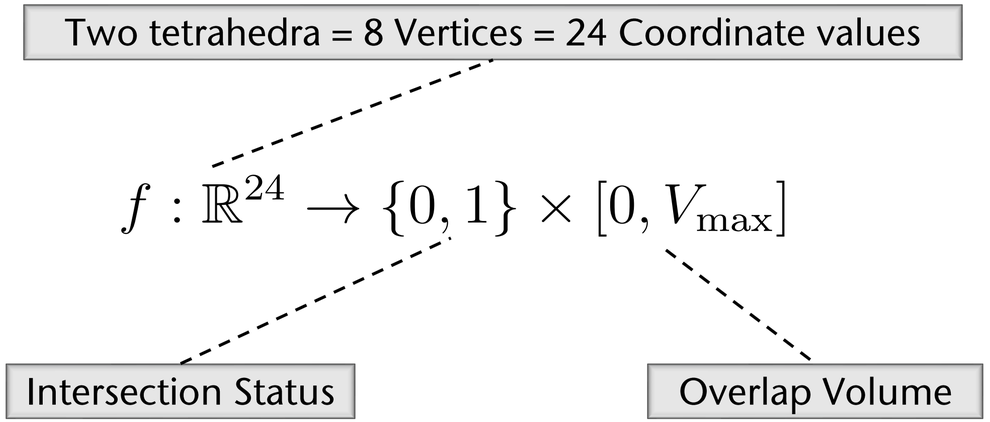
Related Work

- Exact collision methods: GJK, EPA, SAT overlap tests
- Learning for geometry exists, but not directly for this task
- PointNet and DeepSets provide permutation-invariant set processing
- Our work: Joint tetrahedron intersection & overlap volume prediction



Our Contributions & Pipeline

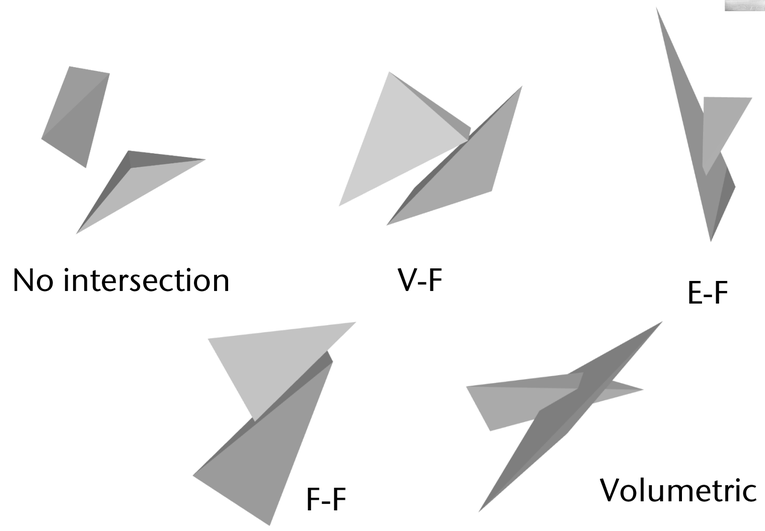
- Reframe tetrahedron intersection as a learned inference problem

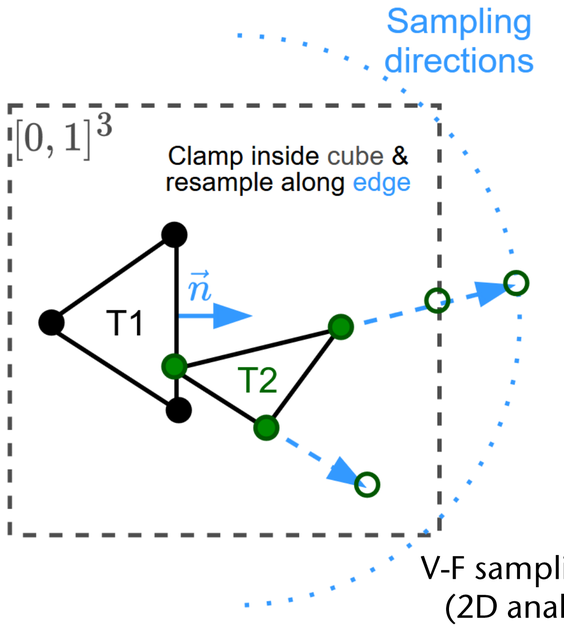


- Permutation-invariant tetrahedron-pair network
  - Jointly predict: Binary intersection and volumetric overlap
- Synthetic data generation for several contact topologies
- 99.3% accuracy and 93x / 16,821x speedups

Training Dataset Generation

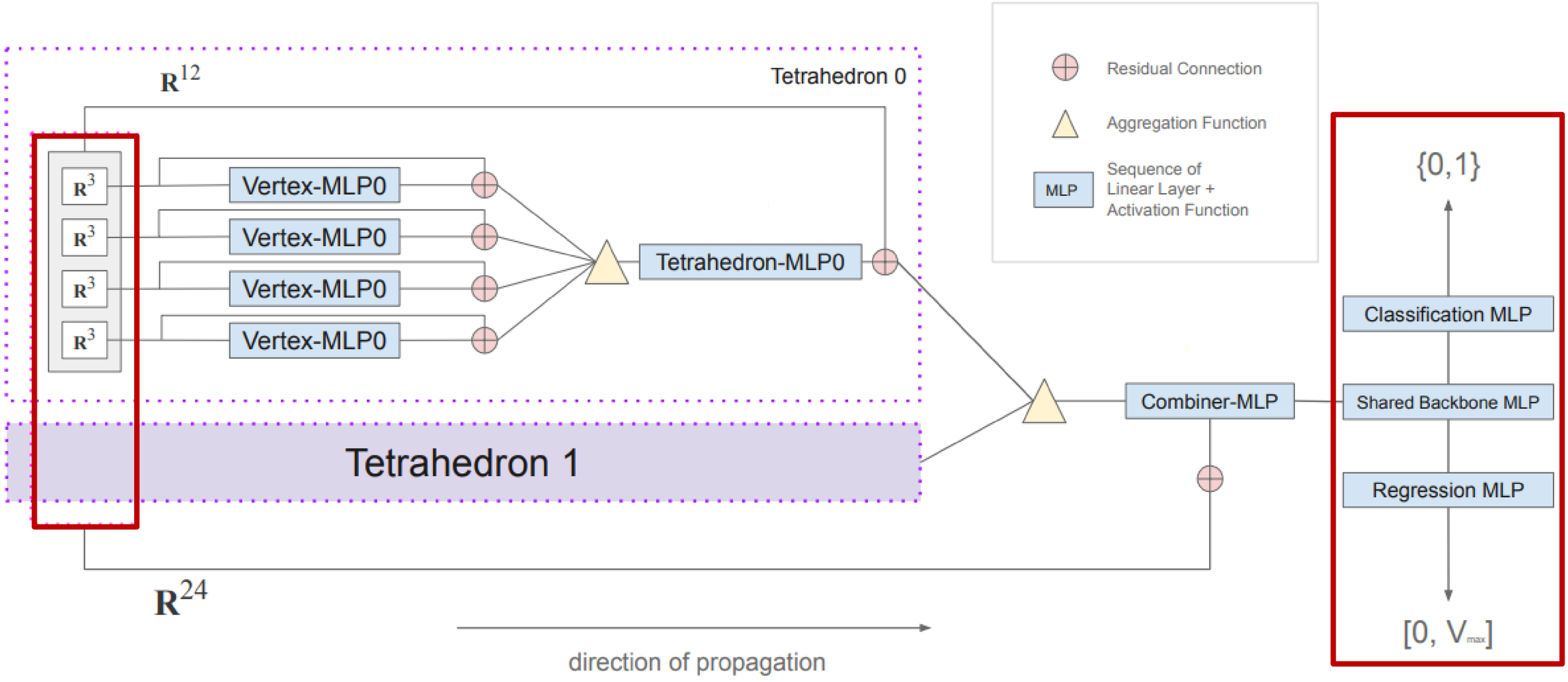
- We use CGAL as exact ground truth
- Dataset generation algorithms based on
  - Rejection sampling for no-intersection and volumetric cases
  - Geometric contraints for other contact cases
- Stratified dataset avoids trivial zero-volume bias
- 12M training samples





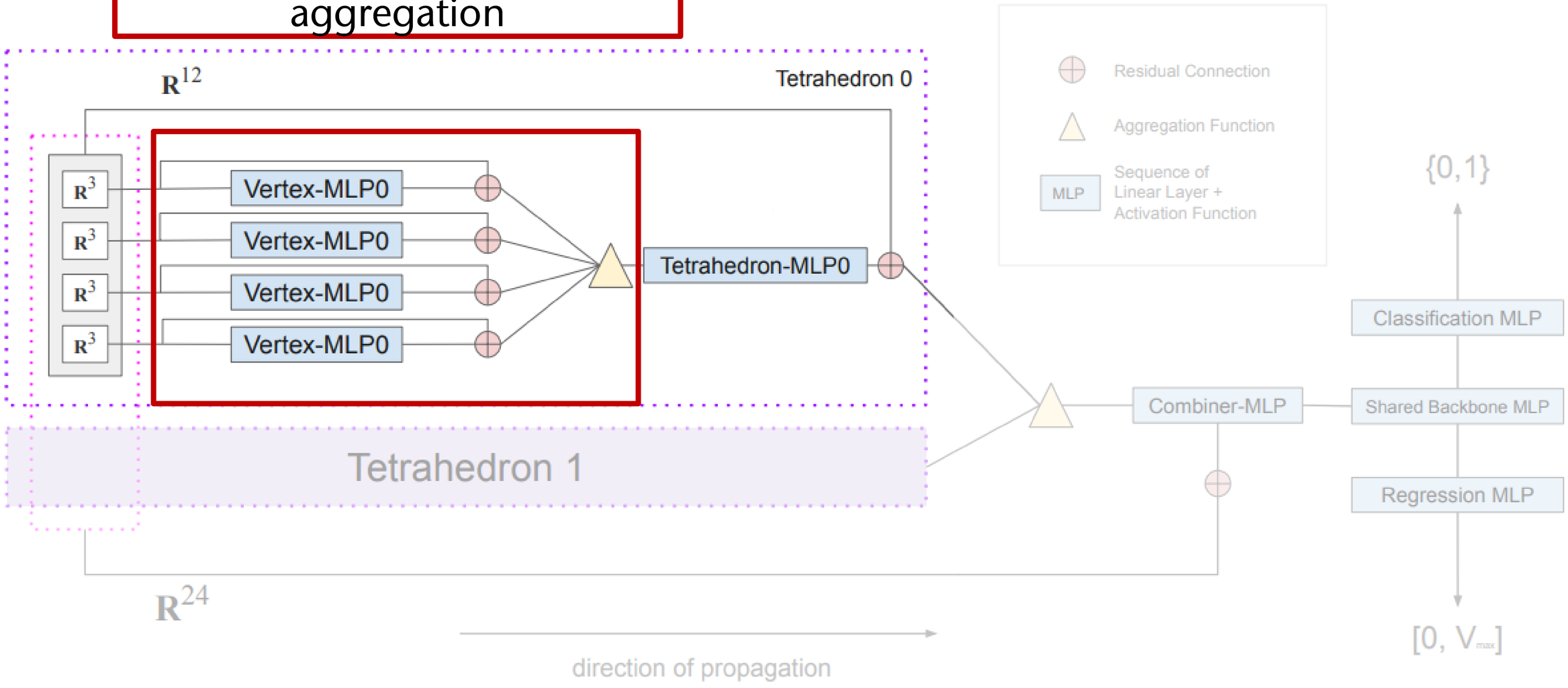
V-F sampling idea (2D analogue)

Architecture of TetrahedronPairNet

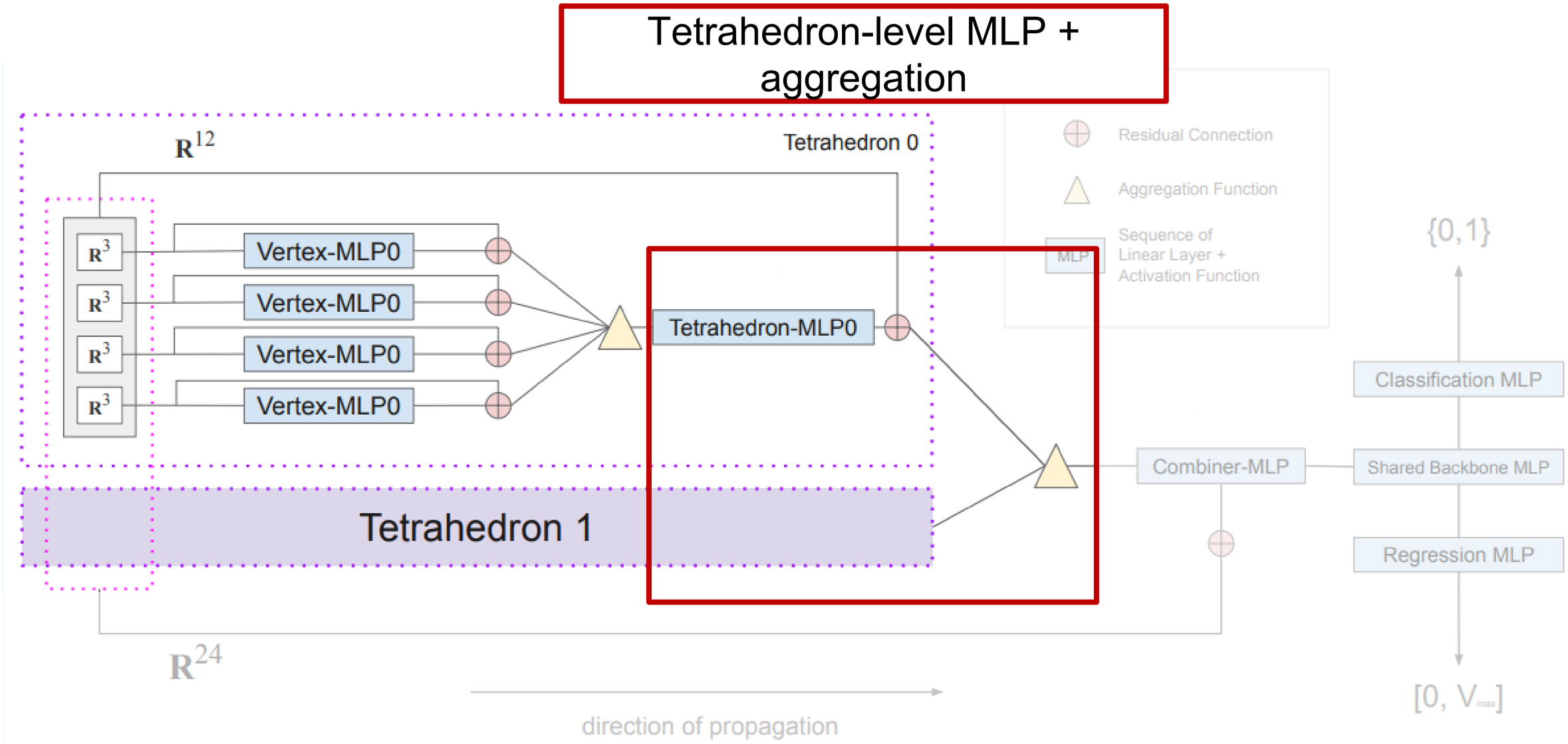


Architecture of TetrahedronPairNet

Vertex-level shared MLP + aggregation

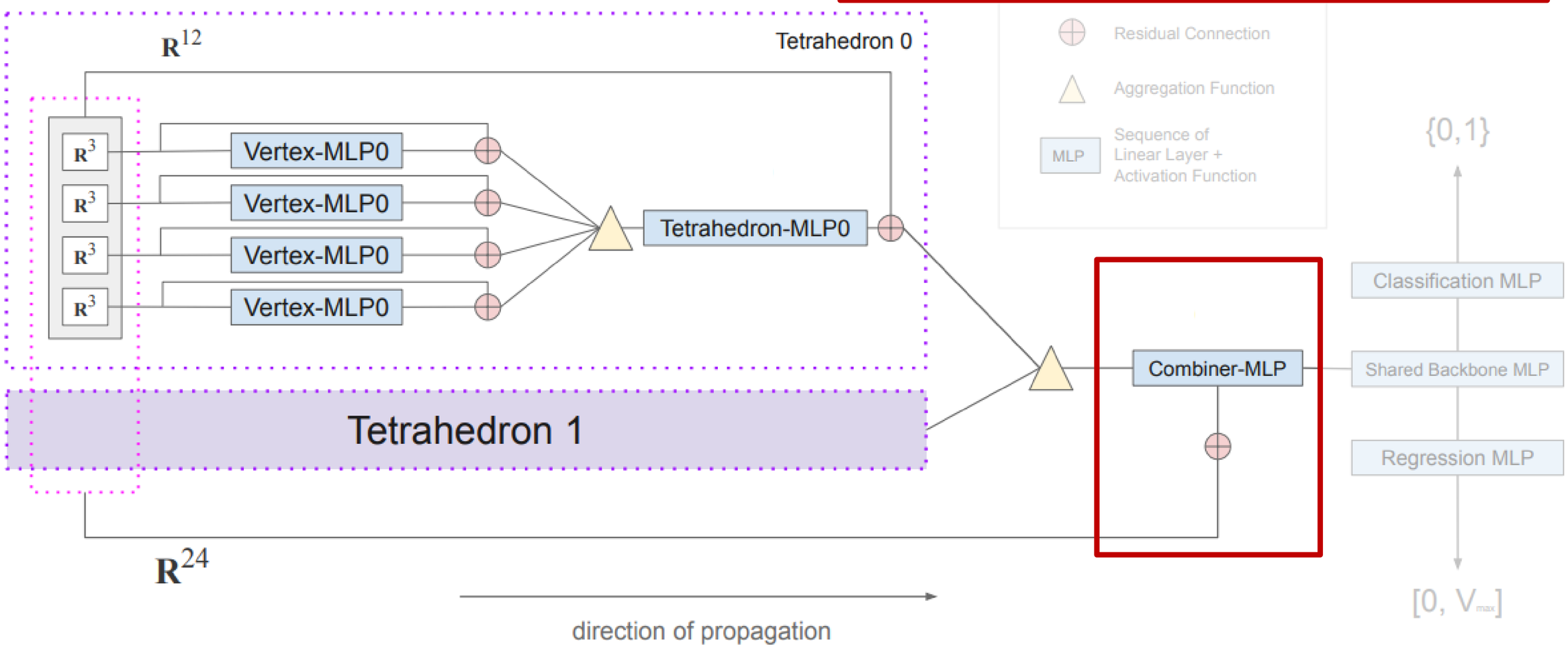


Architecture of TetrahedronPairNet

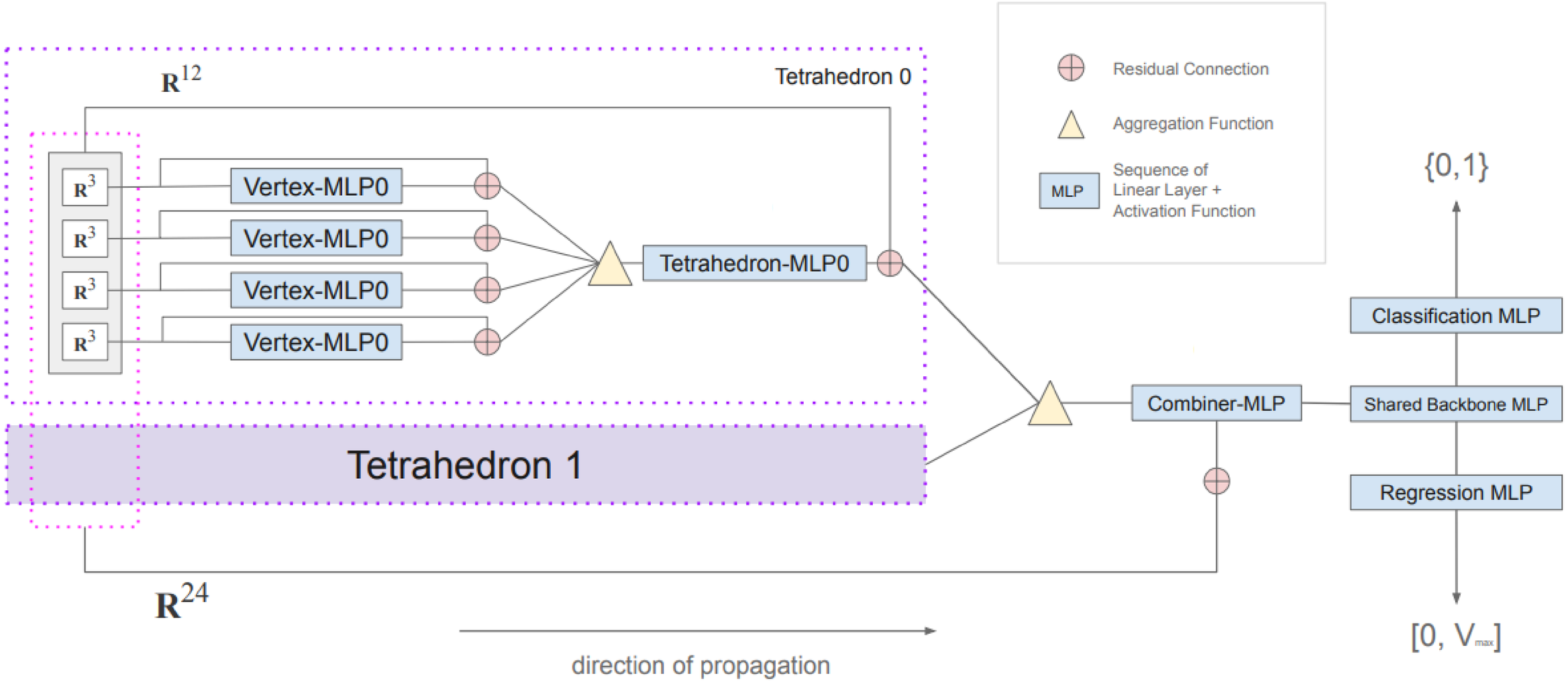


Architecture of TetrahedronPairNet

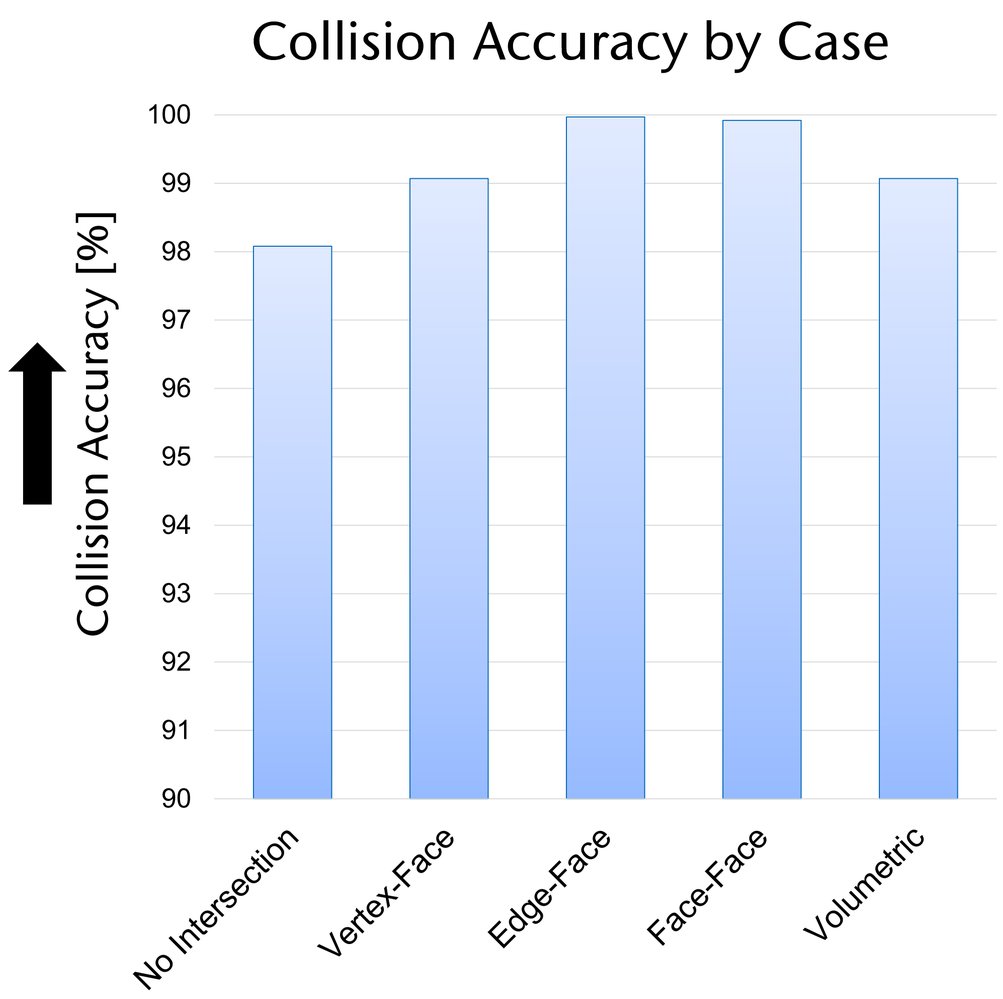
Tetrahedron embeddings combined with 24D input



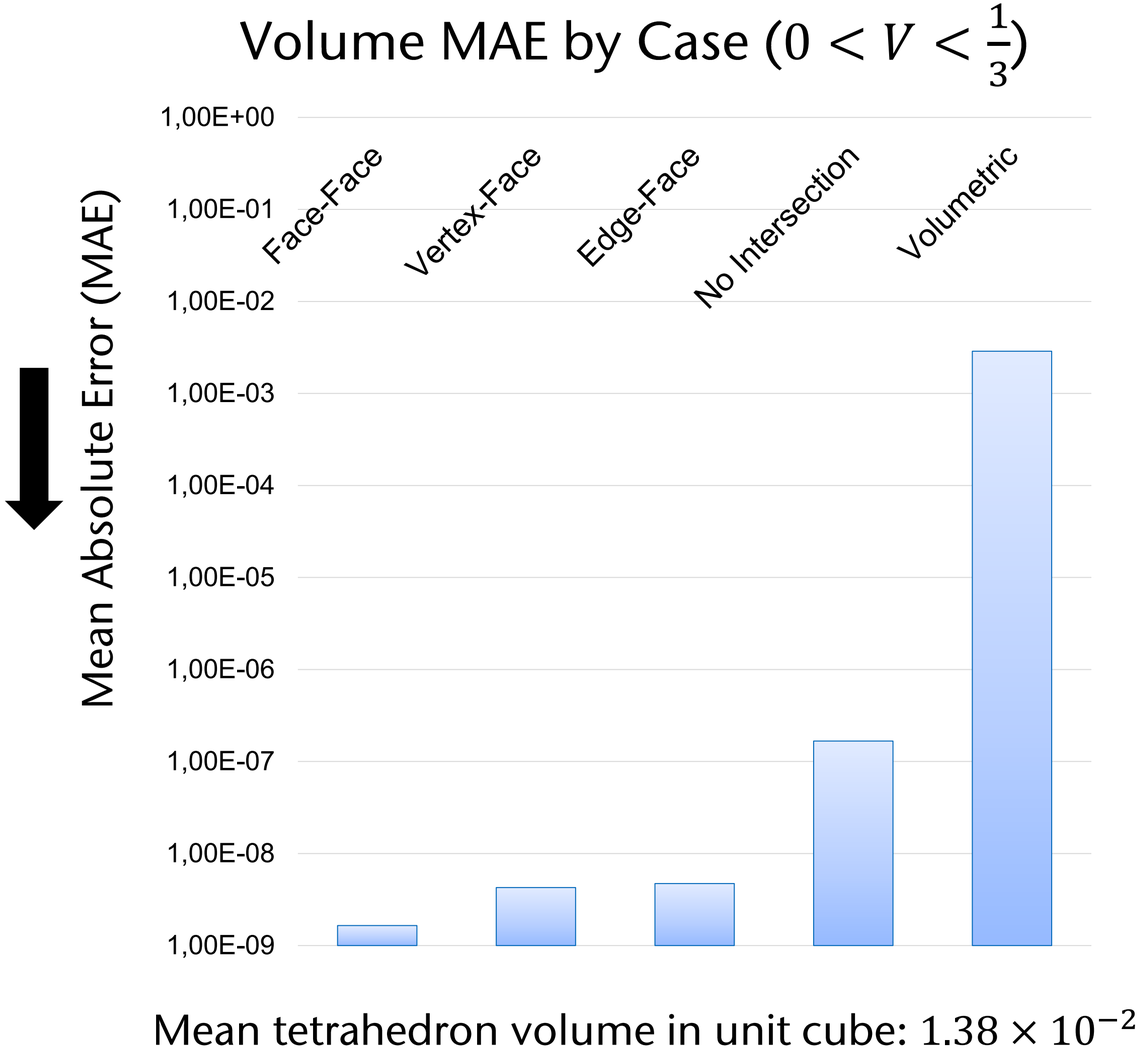
Architecture of TetrahedronPairNet



Accuracy of Collision and Volume Prediction



Mean collision accuracy: 99.37%



Mean tetrahedron volume in unit cube: 1.38×10^{-2}

Volume MAE: 2.88×10^{-3}

Performance Comparison

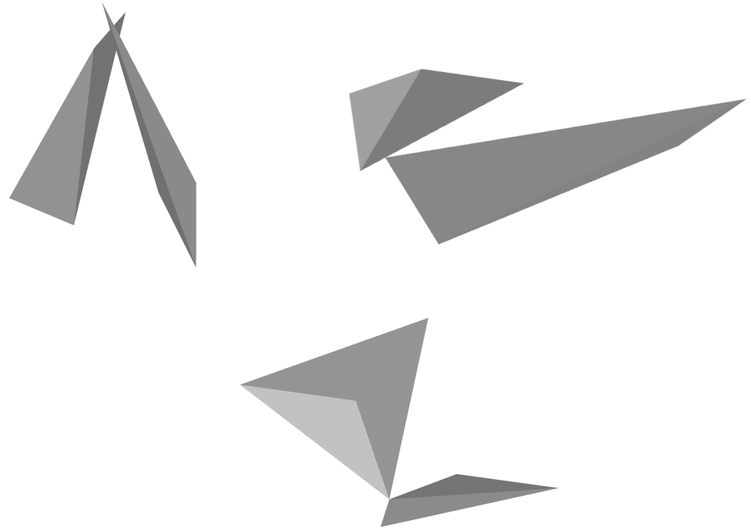
Method	Throughput	Latency (Pre. + Inf.)	Speedup Detection	Speedup Volume
CGAL (Detection, CPU)	6,352 samples/s	157 µs	Baseline CGAL::do_intersect	
CGAL (+ Volume, CPU)	35 samples /s	28.31 ms	Baseline CGAL::Nef_polyhedron_3	
TetrahedronPairNet (CPU, BS=2048)	271,313 samples/s	3.69µs (1.80 + 1.89)	**43×**	**7,751×**
TetrahedronPairNet (GPU, BS=2048)	588,744 samples/s	1.70µs (1.11 + 0.59)	**93×**	**16,821×**

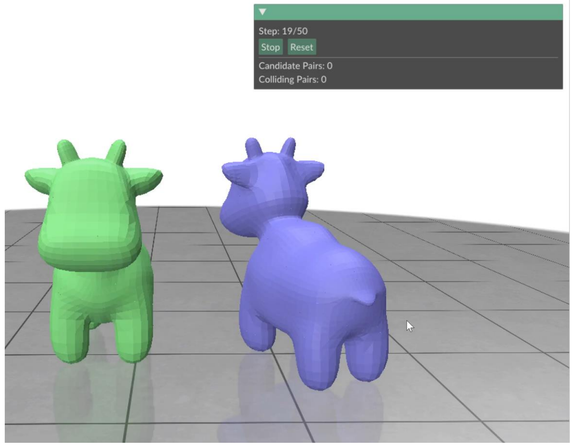
Large speedups on CPU and GPU

Note: CGAL is exact; TetrahedronPairNet is approximate

Limitations and Conclusion

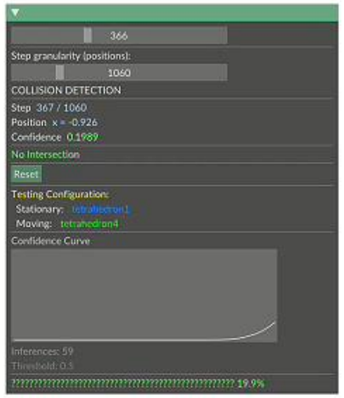
- Current limitations:
  - Missing training cases
  - Volume regression degrades for deeper penetration
- Best deployment model:
  - Use as GPU-resident pre-filter
  - Fall back to exact predicates for hard / low confidence cases
- Take-home message:
  - Tet–tet intersection and approximate volume can be predicted jointly by one massively parallel learned operator

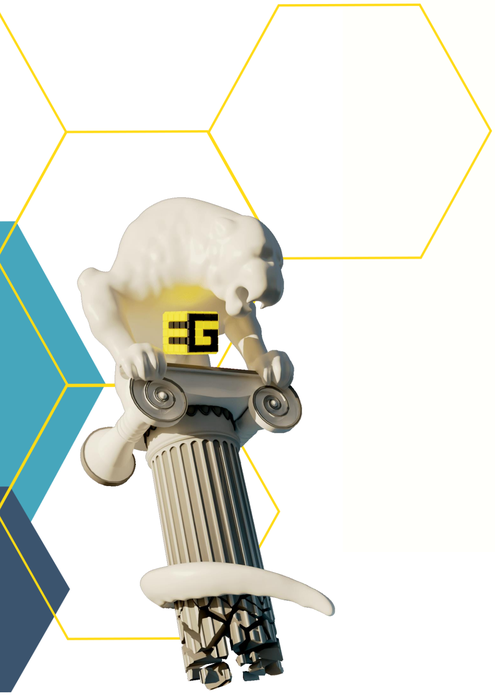


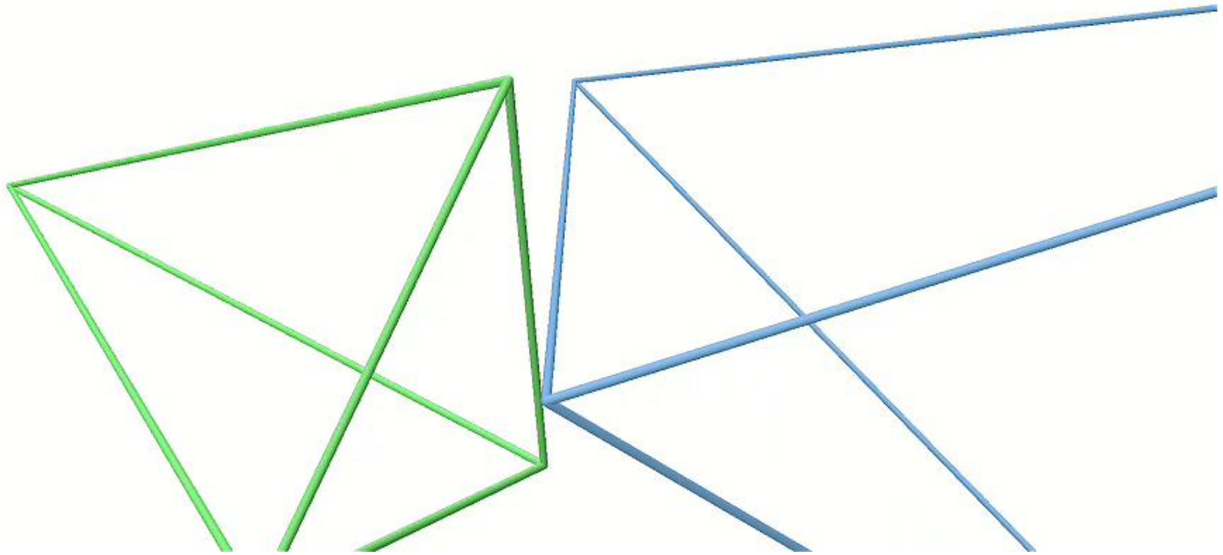


Thank You for Your Attention!

GitHub







erendiropedro@gmail.com
{meissenhelter, zach}@cs.uni-bremen.de

https://cgvr.cs.uni-bremen.de

Training and Architecture Parameters

Training Configuration		Model Architecture	
Training Samples	12,000,000	**Precision**	float32
Optimizer	AdamW	**Total Parameters**	73,370
Learning Rate	10^{-3}	**Model Size**	633 KB
Batch Size	2048	**Activation Function**	ReLU
Epochs	50	**Aggregation Function**	Max-Pooling
Loss Function	0.5 BCE + 0.5 RMSLE	**Vertex MLP**	[24, 24]
Class Balance	0.5 positive/negative	**Tetrahedron MLP**	[48, 48]
Input Normalization	PCA + Unit	**Combiner MLP**	[96, 96]
Data Distribution	[50%, 5%, 7%, 20%, 18%]	**Shared Backbone**	[1024]
Volume Scaling	10^3	**Classification Head**	[128, 64, 32, 1]
		Regression Head	[128, 128, 1]

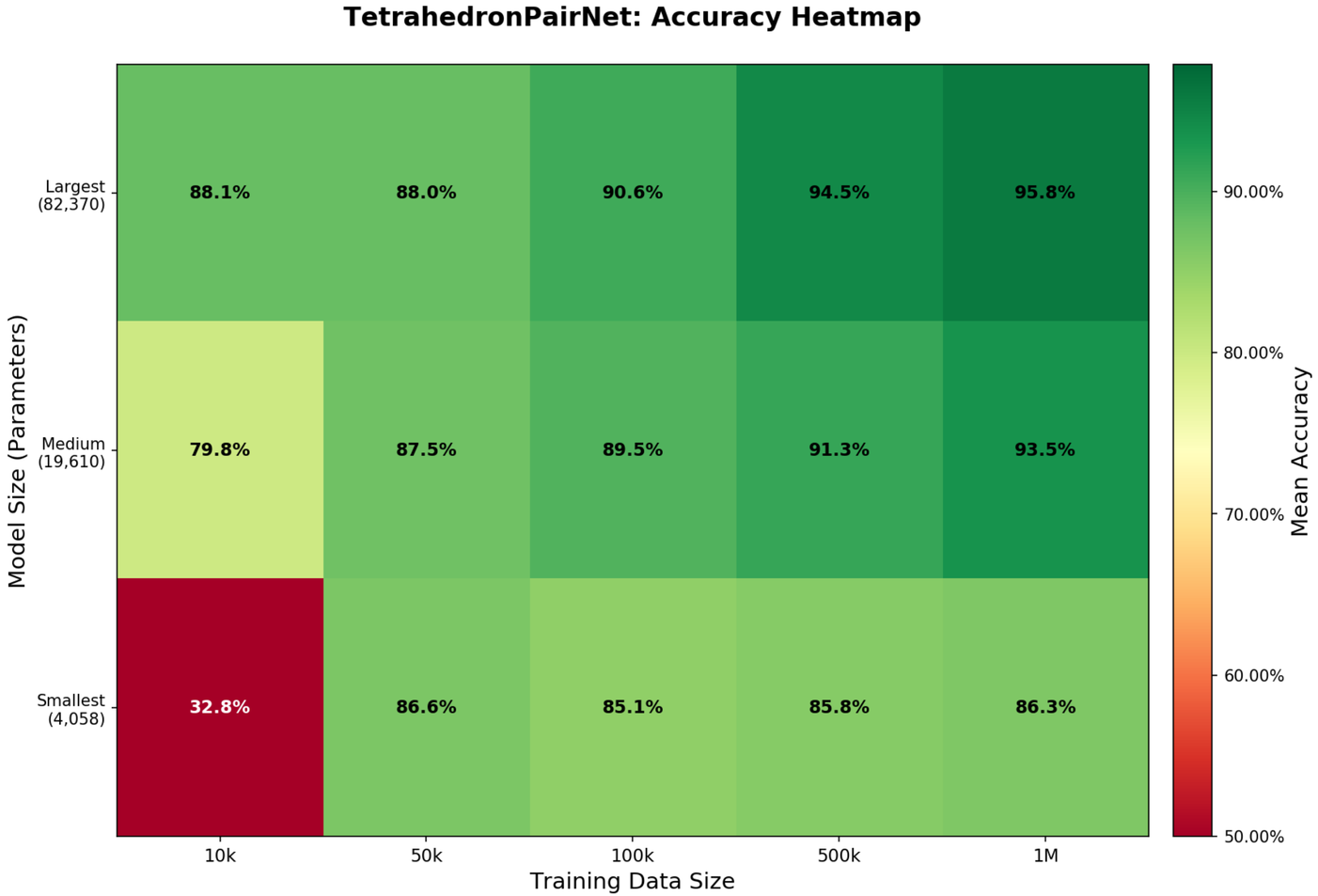
Volume Bin Accuracy

Test Subset / Metric	Detect Acc.	Vol. MAE	Notes
Degenerate & Non-Intersecting Cases			
No Intersection	98.02%	1.67×10^{-7}	Baseline negative class
Vertex–Face	99.07%	4.27×10^{-9}	Correctly predicts ≈ 0 volume
Edge–Face	99.97%	4.72×10^{-9}	Correctly predicts ≈ 0 volume
Face–Face	99.92%	1.65×10^{-9}	Correctly predicts ≈ 0 volume
Volumetric Intersection Analysis			
Volumetric (Overall)	**99.07%**	$\mathbf{2.88 \times 10^{-3}}$	**Volumetric collision case**
Small ($v < 0.0067$)	—	1.47×10^{-3}	88% Bin Acc. (34% of samples)
Medium ($0.0067 \leq v < 0.0133$)	—	2.80×10^{-3}	60% Bin Acc. (33% of samples)
Large ($0.0133 \leq v < 0.0200$)	—	4.44×10^{-3}	41% Bin Acc. (33% of samples)

Normalization

Strategy	Mechanism & Properties
Principal Axis Alignment (PCA)	**Mechanism**: Applies a rigid rotation derived from the PCA of $\mathcal{T}_1$ to both tetrahedra, aligning $\mathcal{T}_1$'s principal axes with the global basis. **Properties**: Standardizes orientation while strictly preserving distances, scale, and volume.
Unitary Tetrahedron Alignment (Unit)	**Mechanism**: Applies an affine transformation that maps $\mathcal{T}_1$ to the unit simplex $\{(0,0,0),(1,0,0),(0,1,0),(0,0,1)\}$ and expresses $\mathcal{T}_2$ in this new frame. **Properties**: Preserves boolean intersection status; intersection volume is scaled linearly by the determinant of the affine map.

Accuracy


TetrahedronPairNet: Accuracy Heatmap
Largest (82,370)
Medium (19,610)
Smallest (4,058)
Model Size (Parameters)
88.1%
88.0%
90.6%
94.5%
95.8%
79.8%
87.5%
89.5%
91.3%
93.5%
32.8%
86.6%
85.1%
85.8%
86.3%
10k
50k
100k
500k
1M
Training Data Size
90.00%
80.00%
70.00%
60.00%
50.00%
Mean Accuracy