



Session T2: Visual Perception From 3D Point Clouds

MAMVI: 3D Test-Time Adaptation via Masked Multi-View Point Clouds

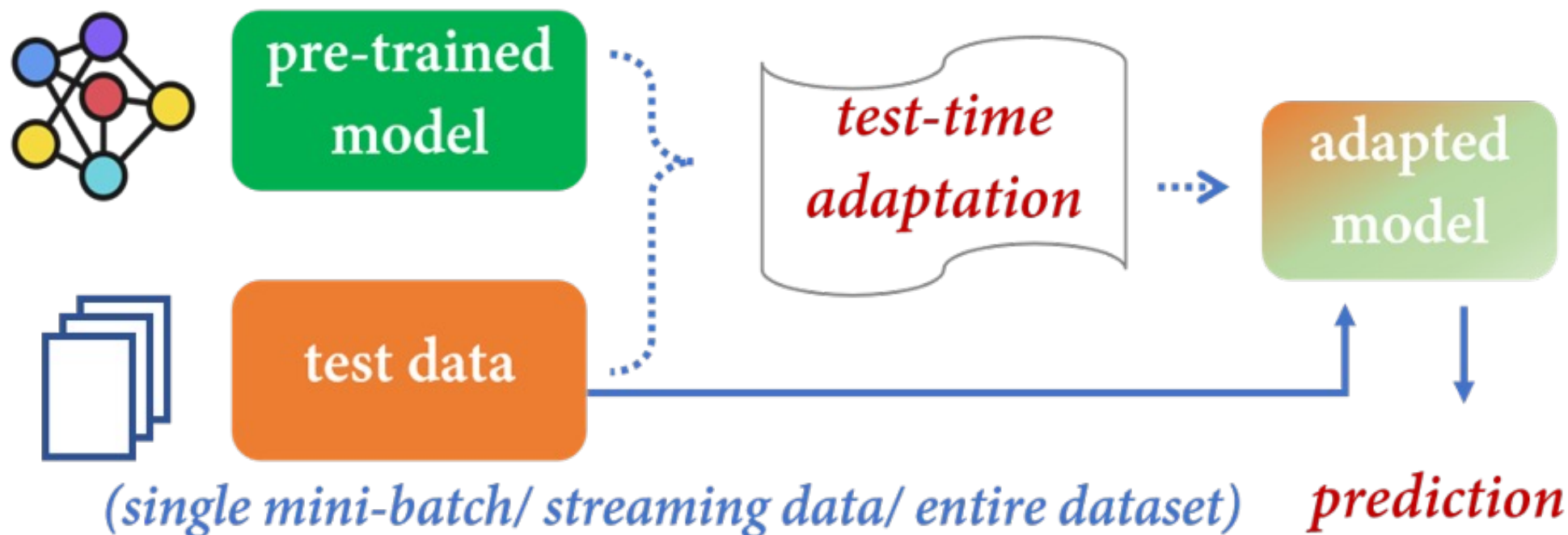
Inseok Kong¹, Geunyoung Jung², and Jiyoung Jung^{2,*}

¹ Department of Geo Informatics, University of Seoul

² Department of Artificial Intelligence, University of Seoul

Presenter: Inseok Kong

• What is Test-Time Adaptation?



Liang et al., IJCV 2025

- Adapt pre-trained models during inference using only unlabeled test data — no source data, no labels, no retraining
- A practical paradigm for handling distribution shifts in real-world deployment

• Why TTA Matters in Practice

- Distribution shifts are unavoidable in real-world deployments
- Autonomous driving: weather and sensor changes affect LiDAR point clouds
- Robotics: new environments introduce unseen object geometries
- Medical imaging: different scanners produce different noise patterns
- Collecting labeled data for every possible scenario is impractical
- Models must adapt on-the-fly without human intervention

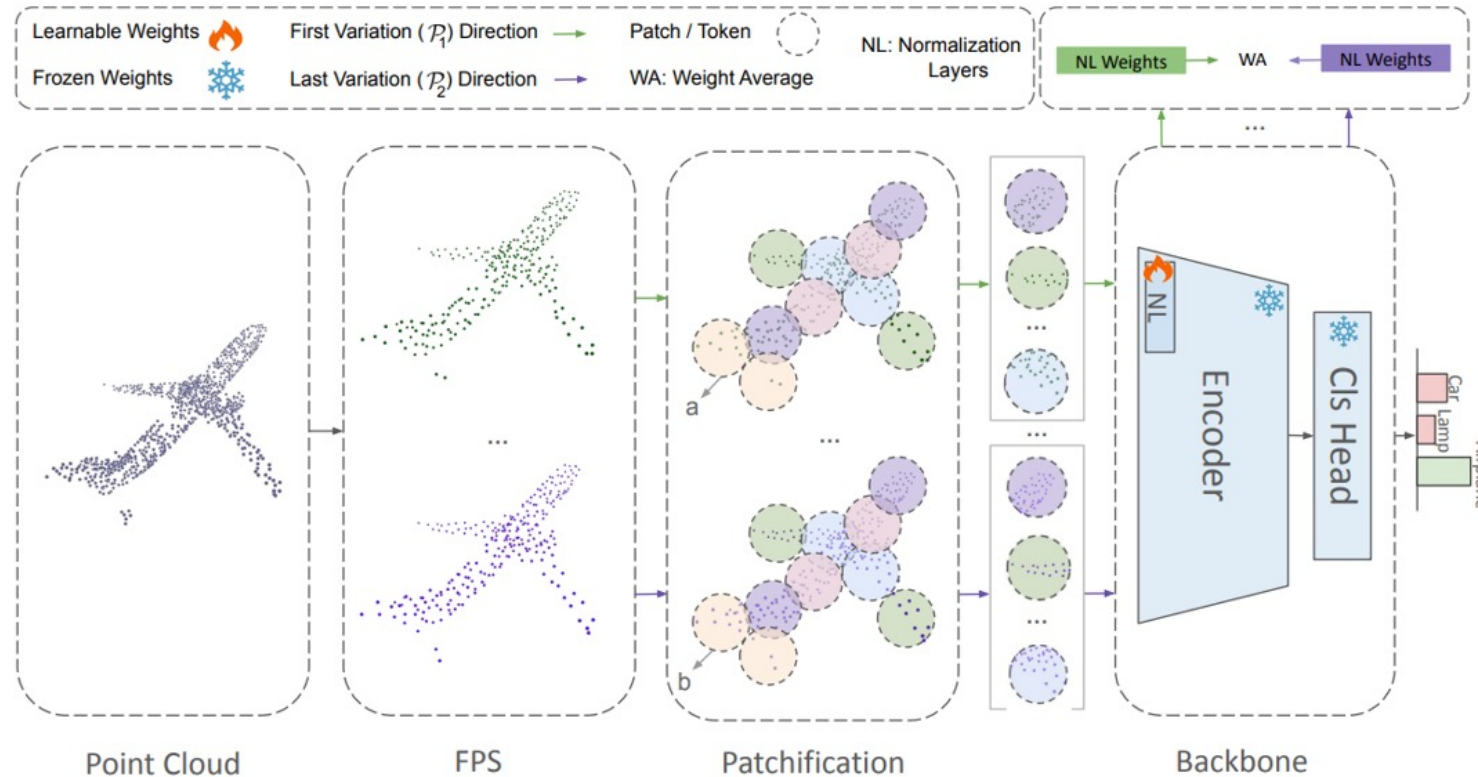
• Why is 3D Point Cloud TTA Harder?

- 2D images have regular grid structure with dense pixel information
- 3D point clouds are sparse, unordered, with no grid structure
- Existing 2D TTA methods (TENT, SHOT, SAR) assume grid-structured inputs
- Cannot directly apply these methods to irregular point cloud data
- 3D TTA requires methods specifically designed for geometric data
- This remains an emerging and under-explored research area

• Challenges in Existing 3D TTA Methods

- Source data dependency: some methods need source data or prototypes at test time (CloudFixer, BFTT3D)
- Pre-training modification: some require architecture changes and auxiliary tasks (MATE, SMART-PC)
- Inference latency: multi-view methods are accurate but sequential optimization is slow
- No existing method is simultaneously source-free, modification-free, and real-time
- These limitations motivate a new approach to 3D TTA

• Related Work: SVWA

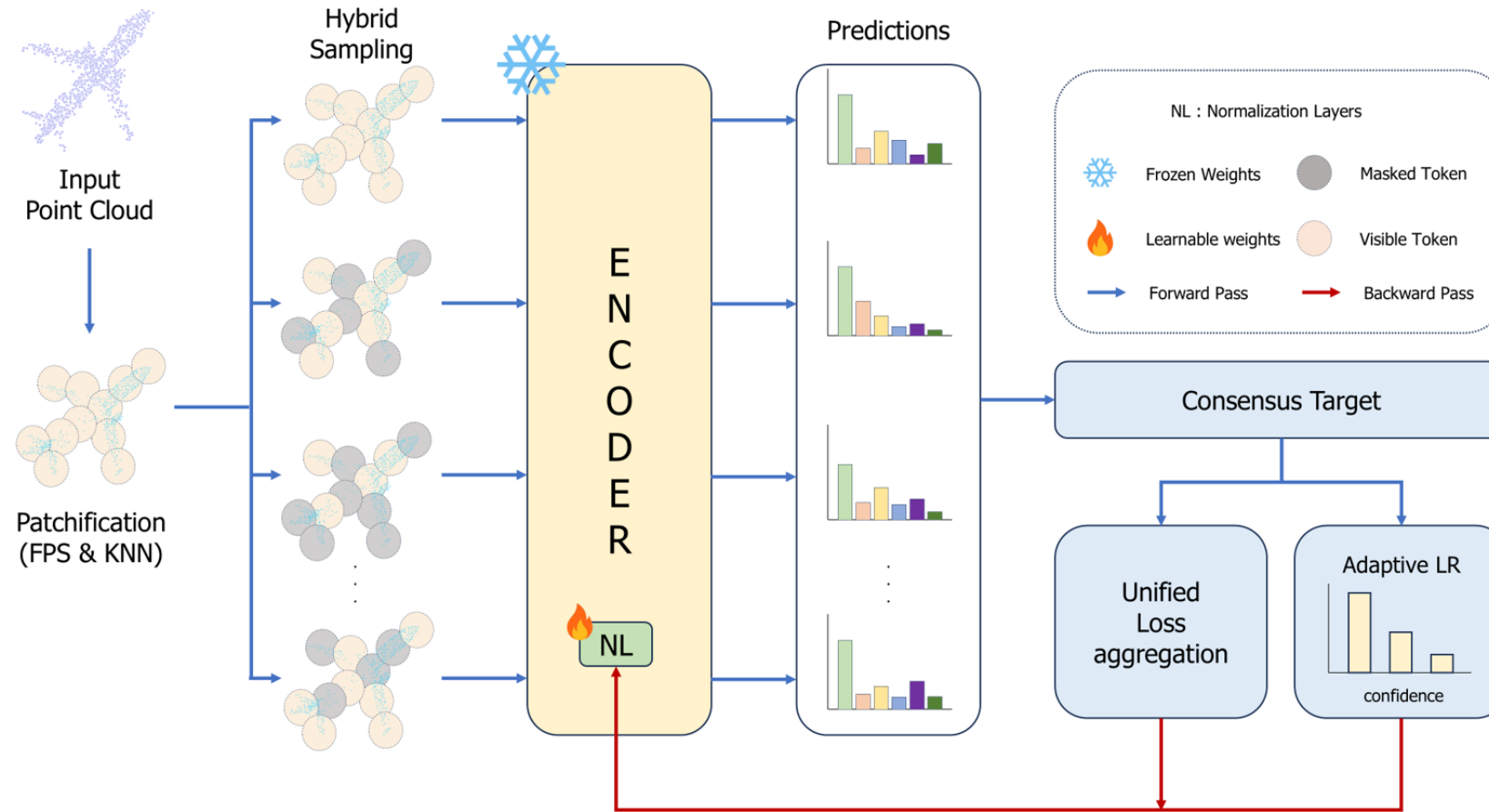


- Source-free multi-view TTA with strong accuracy, but M views require M sequential backward passes
- Only ~ 9 FPS — too slow for real-time applications

• Motivation

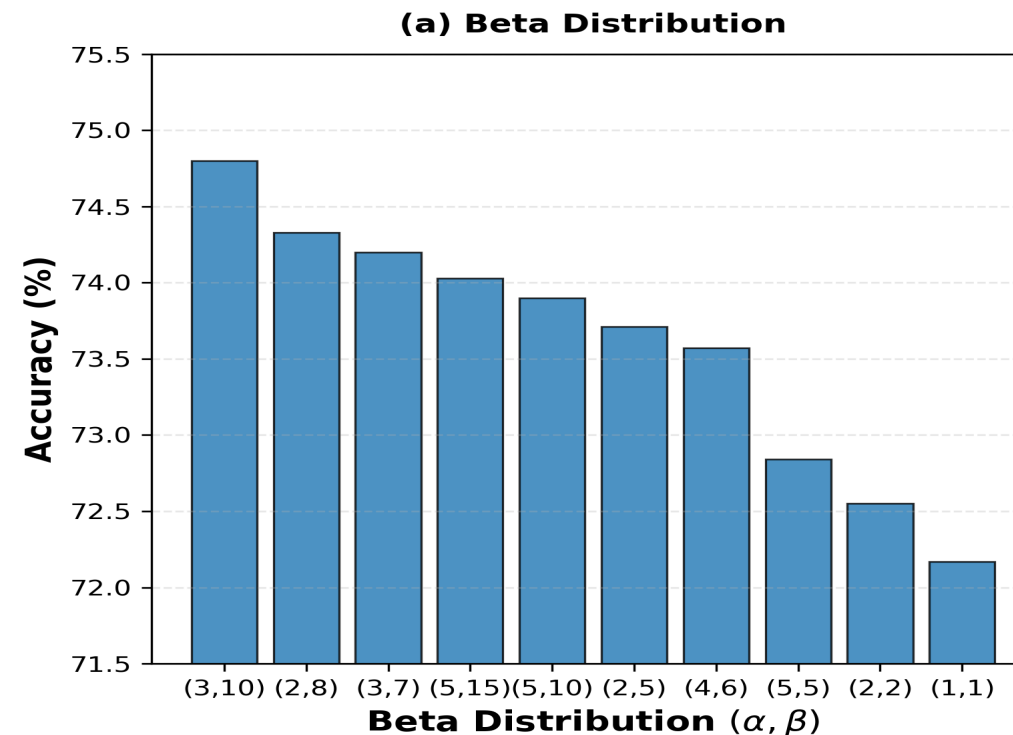
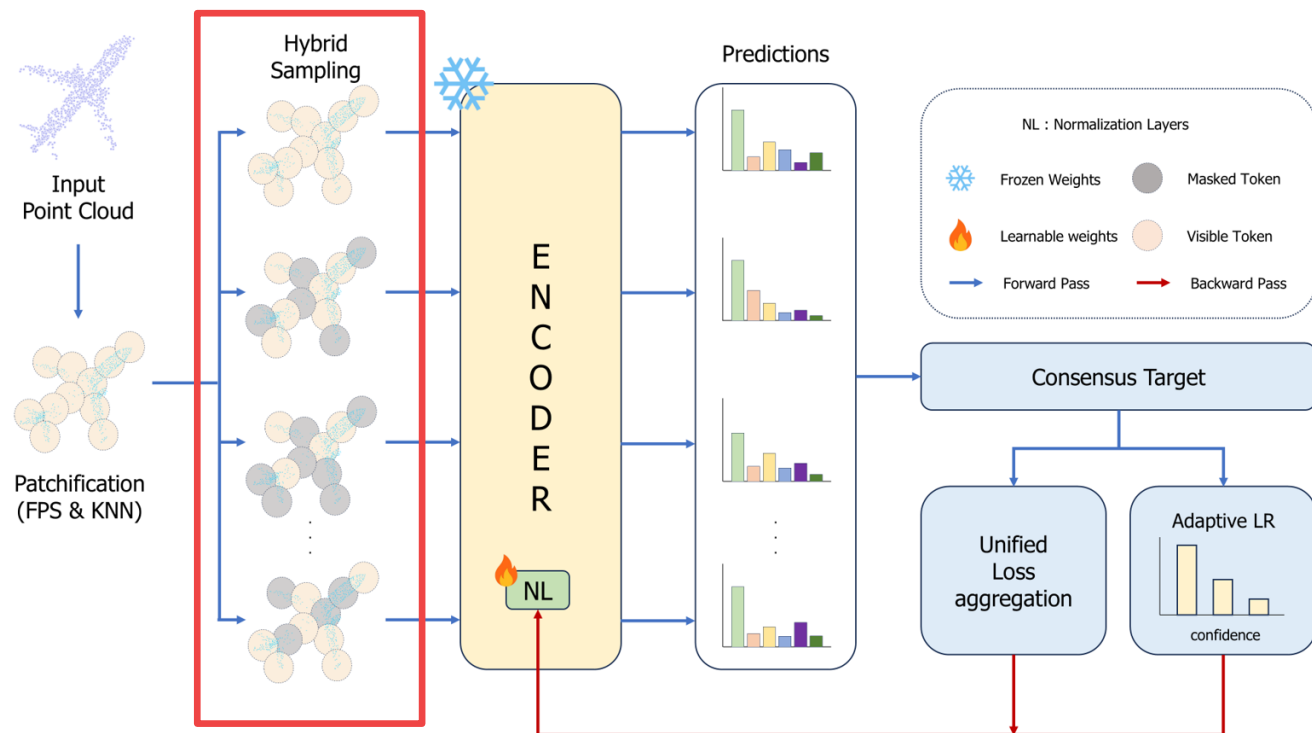
- SVWA shows multi-view augmentation is effective for 3D TTA
- But sequential per-view optimization is the key speed bottleneck
- Our idea: aggregate all view losses into a single backward pass
- Generate views via masking instead of repeating expensive FPS
- Dynamically adjust learning rate based on prediction confidence
- Goal: match or exceed accuracy while dramatically improving speed

• MAMVI: Framework Overview



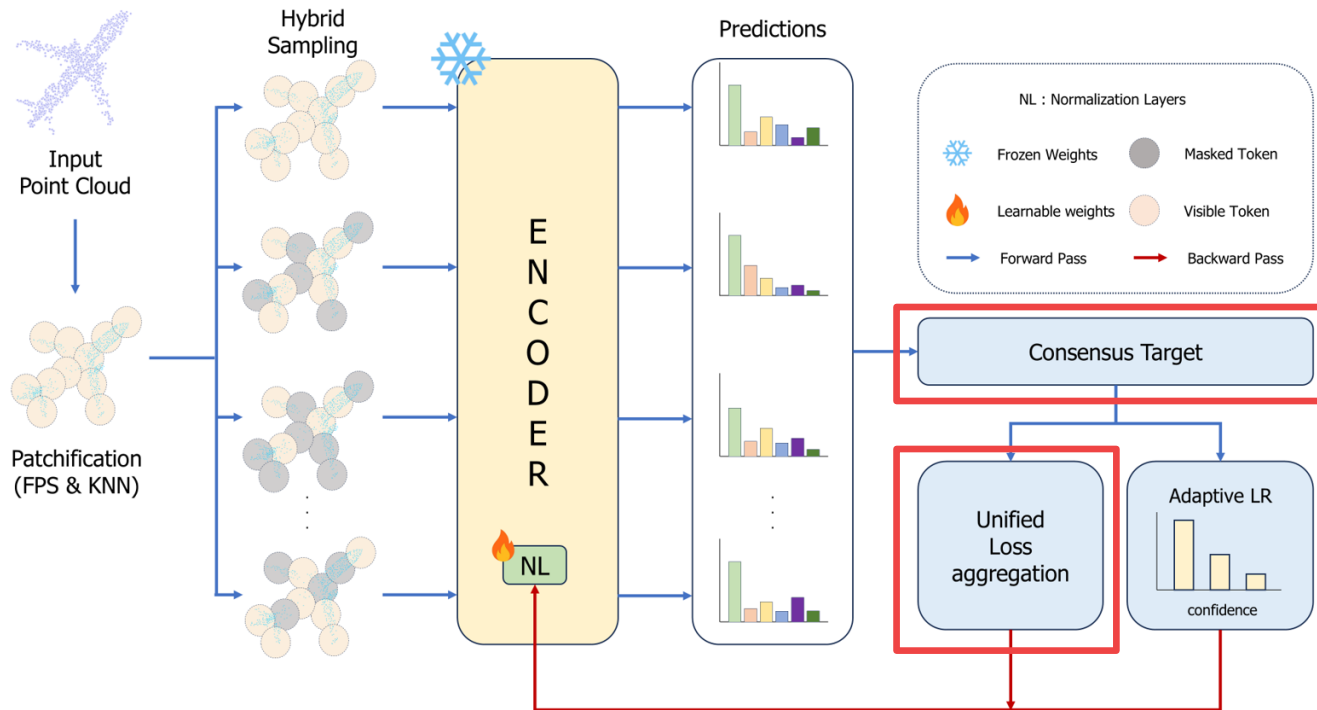
- ① Hybrid masking → ② Unified loss aggregation → ③ Adaptive learning rate
- Single backward pass updates only normalization layers, regardless of view count

• Step ① Hybrid Masking Strategy



- $R = \{ r_{\text{fixed}} \} \cup \{ r \sim \text{Beta}(\alpha, \beta) \}$ — fixed ratios for stability, Beta for diversity
- $M = 12$ views generated efficiently without repeating FPS

• Step ② Consensus Target & Unified Loss



$$\bar{y} = (1/M) \sum p_m$$

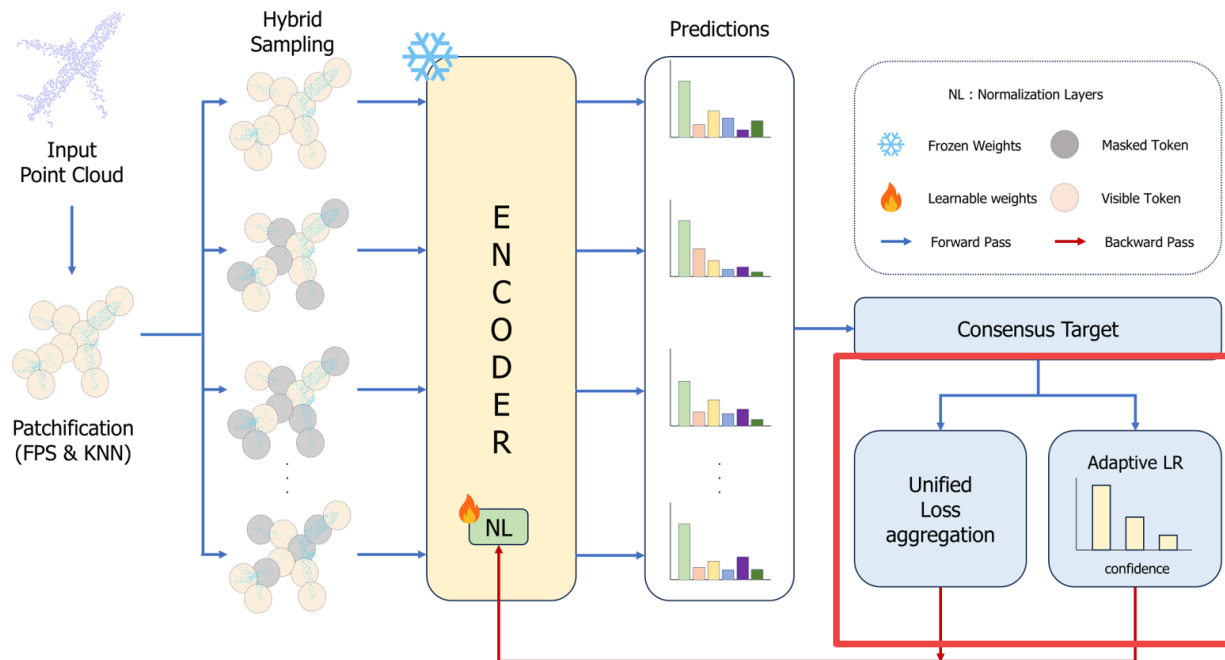
Consensus target: average predictions from M views

$$L = (1/M) \sum [\lambda_{\text{ent}} \cdot H(p_m) + \lambda_{\text{cons}} \cdot D_{\text{KL}}(p_m \parallel \bar{y})]$$

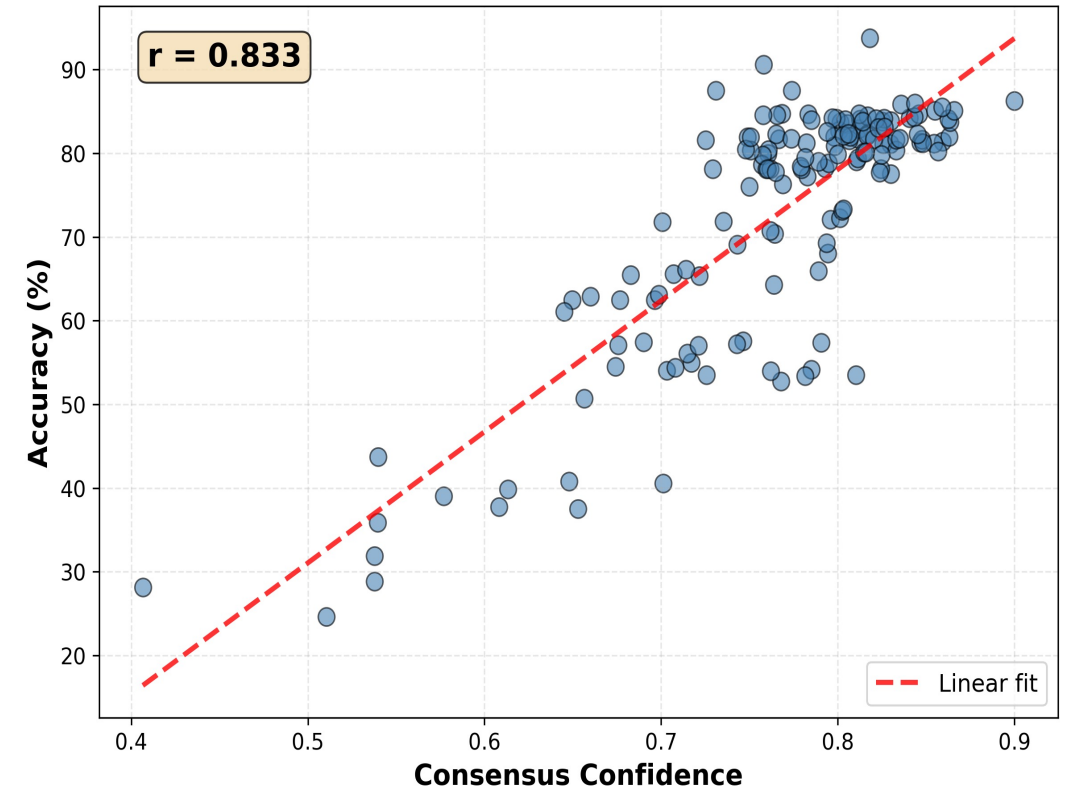
Entropy → confident predictions
KL divergence → align with consensus

- Consensus formed by averaging M masked view predictions as stable target
- All M views optimized in a single backward pass — normalization layers only

• Step ③ Adaptive Learning Rate



Correlation Analysis



- $\eta = \eta_{\text{base}} \times g(\phi)$ — low confidence $\rightarrow$ larger update; high confidence $\rightarrow$ smaller update
- Pearson $r = 0.833$ validates confidence as a reliable proxy for correctness

• Results & Conclusion

Method	ModelNet-40C	ShapeNet-C	ScanObjectNN-C
Source Only	54.13	61.35	36.96
TENT	63.65	61.01	45.97
SMART-PC	63.10	64.40	39.60
BFTT3D	57.16	60.73	33.00
PG-SP	68.93	64.17	49.13
PG-SF	68.68	64.40	47.54
SVWA	75.04	65.74	47.45
MAMVI (Ours)	74.80	66.77	49.76

Inference Speed (FPS)

	ModelNet	ShapeNet	ScanObj
SVWA	8.99	8.63	9.03
MAMVI	77.10	76.64	44.50
Speedup	8.6×	8.9×	4.9×

SOTA

ShapeNet-C
66.77%

SOTA

ScanObjectNN-C
49.76%

8.9×

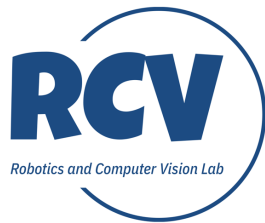
Faster
inference

- SOTA on ShapeNet-C and ScanObjectNN-C; competitive on ModelNet-40C
- 4.9–8.9× faster than SVWA (44–77 FPS vs. ~9 FPS)

- ① Hybrid masking for diverse views
- ② Single backward pass → 4.9–8.9× speedup
- ③ SOTA on ShapeNet-C & ScanObjectNN-C

• Future Work

- Extend beyond classification to 3D segmentation and detection tasks
- Explore continual adaptation across sequential test batches
- Investigate applicability to non-Transformer backbone architectures
- Combine with input-level adaptation methods for complementary gains
- Scale to large-scale outdoor scenes for autonomous driving and robotics



Thank you!

[For More Information]



Linkedin



Github

Appendix

Table 1. Top-1 classification accuracy (%) for all corruption types in the ModelNet-40C dataset.

Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Source Only	66.53	59.12	7.21	31.69	74.22	67.71	69.77	62.28	75.08	74.35	38.13	58.63	70.02	38.53	18.62	54.13
<i>TTT Methods</i>																
TTT-Rot [22]	61.30	58.30	34.50	48.90	66.70	63.60	63.90	59.80	68.60	55.20	27.30	54.60	64.00	40.00	29.10	53.00
MATE [15]	69.80	61.80	18.90	63.90	72.50	64.00	66.00	74.00	80.80	71.00	36.70	69.20	66.30	38.40	29.90	58.90
SMART-PC [1]	82.40	80.10	12.00	67.10	84.50	76.00	78.60	67.30	72.90	73.30	43.90	72.60	73.50	37.40	24.80	63.10
<i>Source-Dependent Methods</i>																
BFTT3D [25]	66.53	59.20	7.01	31.85	65.60	67.79	69.81	77.96	85.25	74.35	38.17	74.80	70.02	38.61	30.51	57.16
PG-SP [27]	77.92	73.83	28.12	69.00	80.88	72.93	76.09	81.40	83.59	77.43	56.94	79.58	74.60	48.99	52.67	68.93
<i>Source-Free Methods</i>																
PL [11]	74.11	69.61	19.25	61.71	68.44	70.10	70.58	77.55	81.48	73.50	53.12	74.68	70.66	43.35	39.47	63.17
TENT [24]	74.96	71.07	20.26	61.75	69.57	70.14	72.08	77.84	81.89	73.95	53.12	75.08	70.34	44.69	38.01	63.65
SHOT [12]	62.52	55.83	19.53	54.74	58.51	57.58	62.07	58.91	69.37	58.75	45.18	63.78	58.91	39.38	37.88	53.52
T3A [10]	65.67	61.40	21.14	51.66	67.17	63.27	65.87	66.36	69.77	66.36	50.61	63.43	62.91	44.89	43.59	60.24
LAME [3]	72.16	59.76	4.34	33.51	67.26	67.95	69.94	78.04	85.82	74.51	40.32	74.47	70.91	37.88	28.32	57.67
DUA [14]	65.00	58.50	14.70	48.50	68.80	62.80	63.20	62.10	66.20	68.80	46.20	53.80	64.70	41.20	36.50	54.73
MEMO [28]	80.79	75.28	23.99	64.79	73.42	74.88	76.18	83.14	86.10	78.36	58.06	80.79	74.92	48.30	43.72	68.18
SAR [16]	75.20	70.99	24.55	61.22	69.65	70.06	71.47	78.16	81.65	74.47	54.58	76.54	69.89	44.17	39.30	64.12
PG-SF [27]	78.12	74.49	40.88	66.69	81.73	73.58	75.81	79.09	82.37	77.23	58.19	75.49	73.58	47.85	45.14	68.68
SVWA [2]	83.64	83.97	33.08	77.39	86.89	80.40	82.71	84.78	87.95	82.30	67.33	83.73	80.36	55.80	55.11	75.04
MAMVI (Ours)	84.01	82.35	41.48	73.50	85.84	79.42	81.98	84.01	86.40	82.47	66.27	83.12	79.99	53.61	57.55	74.80

Appendix

Table 2. Top-1 classification accuracy (%) for all corruption types in the ShapeNet-C dataset.

Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Source Only	77.34	71.78	8.64	54.51	77.90	75.50	76.06	85.25	76.42	80.48	57.08	85.12	76.05	10.97	7.13	61.35
<i>TTT Methods</i>																
TTT-Rot [22]	74.60	72.40	23.10	59.90	74.90	73.80	75.00	81.40	82.00	69.20	49.10	79.90	72.70	14.00	12.00	60.90
MATE [15]	77.80	74.70	4.30	66.20	78.60	76.30	75.30	86.10	86.60	79.20	56.10	84.10	76.10	12.30	13.10	63.10
SMART-PC [1]	80.80	78.90	8.90	60.40	81.80	81.10	81.70	84.80	78.40	80.80	63.70	84.90	79.80	11.50	8.80	64.40
<i>Source-Dependent Methods</i>																
BFTT3D [25]	72.58	65.91	10.57	60.86	67.78	73.45	74.38	84.55	83.52	78.58	57.96	83.71	75.17	11.95	10.00	60.73
PG-SP [27]	78.63	76.52	15.78	72.70	78.78	76.49	77.18	83.28	81.18	79.05	63.36	83.07	76.33	9.85	10.34	64.17
<i>Source-Free Methods</i>																
PL [11]	70.30	64.20	7.86	58.00	64.22	72.25	71.61	79.92	79.16	74.90	59.25	78.30	72.19	10.30	10.50	59.41
TENT [24]	71.90	67.48	9.29	63.53	66.87	72.55	73.48	82.09	81.35	76.56	59.82	81.03	72.65	8.52	8.93	61.01
SHOT [12]	27.78	21.97	3.26	22.81	23.24	24.27	31.22	33.86	31.00	27.86	22.57	34.19	28.73	3.48	2.77	22.84
T3A [10]	66.69	64.18	14.00	59.40	67.14	64.32	64.90	71.07	66.69	66.34	54.76	69.42	64.15	9.89	10.46	54.23
LAME [3]	72.88	66.90	4.90	58.43	64.48	73.54	74.63	84.40	83.58	78.45	53.44	83.36	75.25	10.39	8.48	60.85
DUA [14]	76.10	70.10	14.30	60.90	76.20	71.60	72.90	80.00	83.80	77.10	57.50	75.00	72.10	11.90	12.10	60.77
MEMO [28]	75.27	71.58	12.47	68.03	70.86	75.88	76.76	84.40	83.70	79.98	65.63	83.53	76.10	10.63	11.49	64.30
SAR [16]	72.49	67.68	9.65	64.26	67.12	73.14	73.84	82.44	81.62	77.54	59.56	81.60	72.55	9.58	10.52	61.46
PG-SF [27]	79.21	75.43	22.46	68.38	78.95	76.78	77.12	83.52	80.73	79.24	64.91	83.55	76.08	10.15	9.55	64.40
SVWA [2]	82.04	80.31	24.93	70.98	81.39	78.84	79.19	82.84	81.40	78.11	67.86	80.07	75.06	11.35	11.79	65.74
MAMVI (Ours)	81.70	79.11	21.54	67.25	81.42	79.72	80.26	84.96	83.98	81.27	69.17	84.59	78.75	12.94	14.94	66.77

Appendix

Table 3. Top-1 classification accuracy (%) for all corruption types in the ScanObjectNN-C dataset.

Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Source Only	21.69	18.76	16.70	18.42	22.20	45.96	46.99	71.95	69.36	48.88	35.63	72.98	49.40	6.71	8.78	36.96
<i>TTT Methods</i>																
TTT-Rot [22]	30.20	34.10	16.20	22.60	22.60	32.40	32.10	45.50	45.00	34.50	29.30	47.80	36.20	7.10	8.10	29.58
MATE [15]	27.50	29.40	14.30	22.20	25.60	40.80	42.00	73.70	63.20	45.10	35.30	73.30	45.30	7.10	9.30	36.90
SMART-PC [1]	27.50	39.10	19.30	21.50	29.80	44.20	48.90	68.30	60.20	49.40	45.40	70.10	49.10	8.40	12.20	39.60
<i>Source-Dependent Methods</i>																
BFTT3D [25]	16.35	14.97	14.29	17.04	13.43	38.90	42.00	69.71	70.74	41.14	28.92	69.19	43.20	6.54	8.61	33.00
PG-SP [27]	41.48	55.59	22.89	38.90	42.17	56.80	62.31	76.08	76.42	59.04	49.74	76.76	62.48	7.57	8.78	49.13
<i>Source-Free Methods</i>																
PL [11]	39.76	36.83	19.10	43.55	37.87	57.14	62.31	66.27	66.27	60.59	49.74	67.30	61.79	8.26	9.12	46.51
TENT [24]	39.24	34.42	17.04	44.75	34.77	58.00	59.04	66.95	66.95	60.24	50.26	67.64	61.27	7.06	9.12	45.97
SHOT [12]	25.82	34.25	13.08	54.04	32.53	57.49	58.86	66.44	66.61	54.73	46.99	65.92	63.34	6.02	8.26	44.57
T3A [10]	37.50	46.01	18.92	37.33	37.67	50.69	52.26	63.72	65.97	50.35	46.88	64.41	54.17	9.72	8.68	42.95
LAME [3]	14.46	12.56	10.33	15.83	12.91	48.88	55.59	58.35	61.10	52.32	35.80	58.35	53.36	9.12	9.47	34.91
DUA [14]	30.50	40.10	10.20	23.60	29.90	43.70	46.10	68.30	66.30	48.50	38.90	68.70	48.40	8.60	8.10	38.66
MEMO [28]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SAR [16]	41.65	38.73	18.24	46.47	39.24	57.14	62.65	66.61	65.58	60.41	52.32	68.16	62.48	8.95	9.47	47.26
PG-SF [27]	40.96	54.76	21.86	36.49	39.59	55.25	58.35	75.56	74.04	56.80	48.71	75.56	58.69	7.06	9.47	47.54
SVWA [2]	44.62	60.94	25.17	38.72	46.70	52.60	55.56	72.05	72.40	52.60	44.44	71.18	52.00	9.28	11.46	47.45
MAMVI (Ours)	43.06	60.07	22.92	41.15	45.49	56.77	61.81	74.83	74.83	56.77	48.79	77.26	63.72	8.33	10.59	49.76

Table 4. Inference throughput (Frames/Second) comparison between MAMVI and SVWA across datasets.

Method	ModelNet-40C	ShapeNet-C	ScanObjectNN-C
SVWA (Frames/Second)	8.99	8.63	9.03
MAMVI (Frames/Second)	77.10	76.64	44.50
Speedup	8.6×	8.9×	4.9×

Appendix

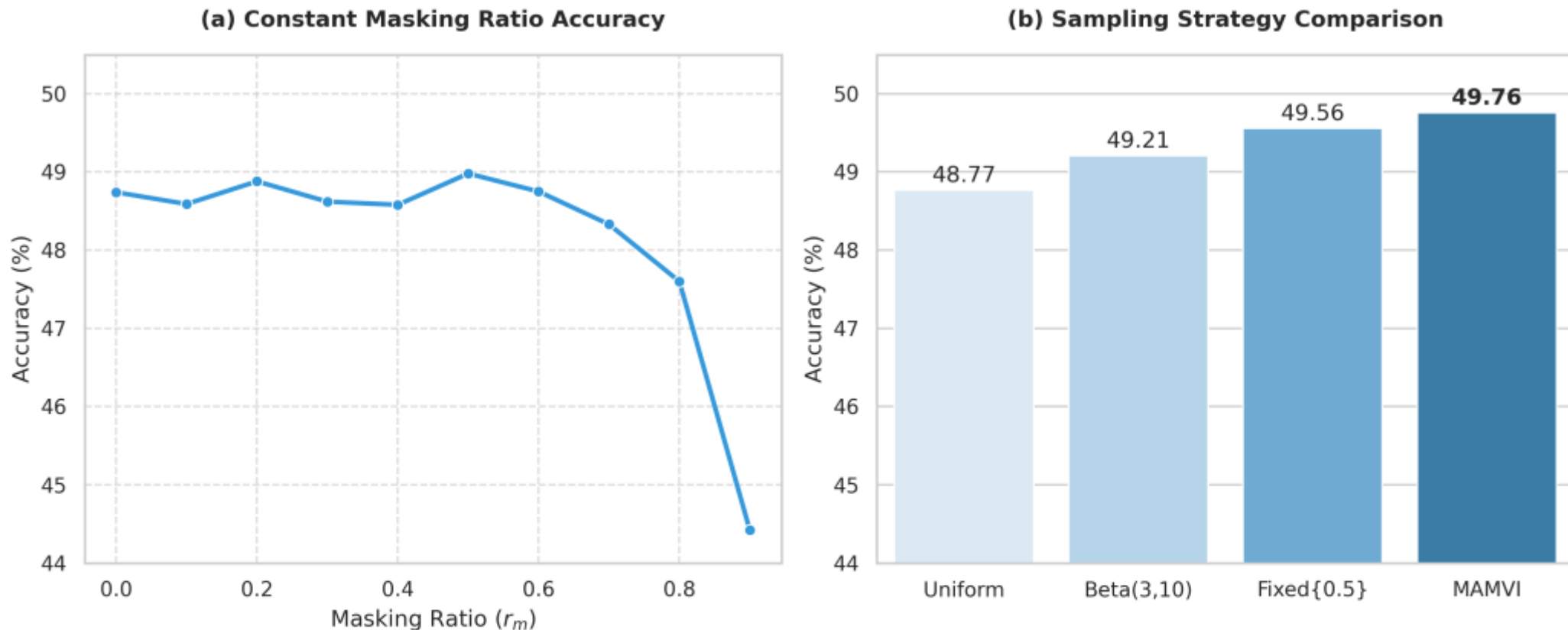


Fig. 2. Analysis of masking strategies on ScanObjectNN-C. (a) Classification accuracy across varying constant masking ratios. (b) Comparison of different sampling strategies, where MAMVI yields the best performance.

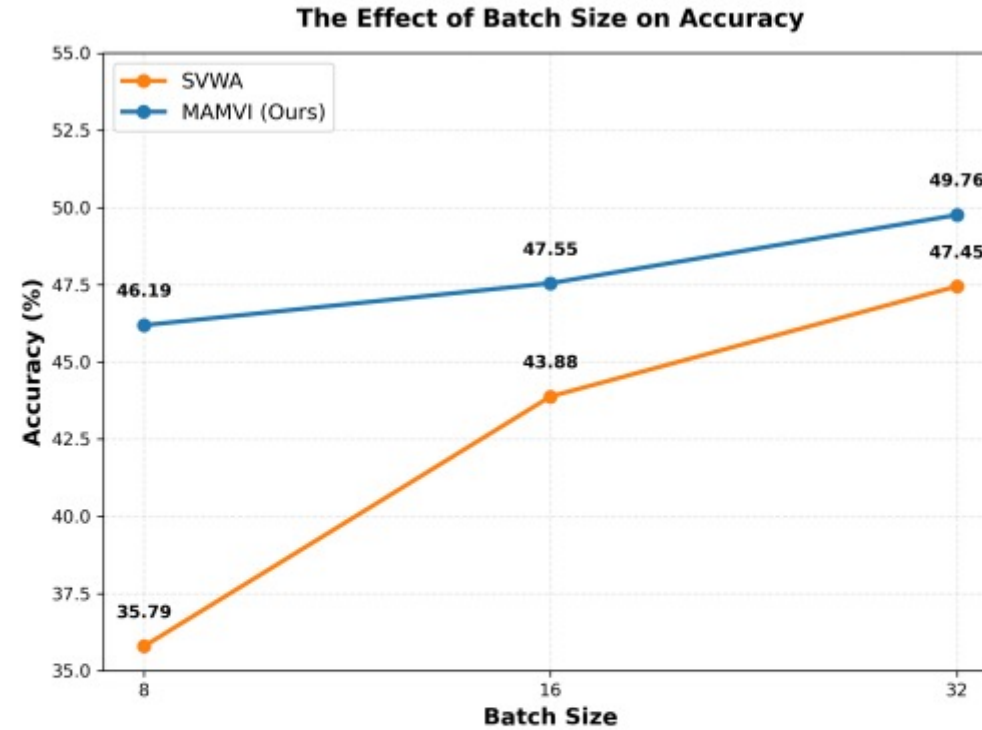


Fig. 3. Impact of batch size on accuracy for two methods. The experiments were conducted on ScanObjectNN-C.

Table 5. Impact of the adaptive learning rate (ALR) mechanism across benchmarks.

Method	ModelNet-40C	ShapeNet-C	ScanObjectNN-C
No ALR	73.86	65.63	49.33
MAMVI (ALR)	74.80	66.77	49.76
Gain (Δ)	+0.94	+1.14	+0.43

Appendix

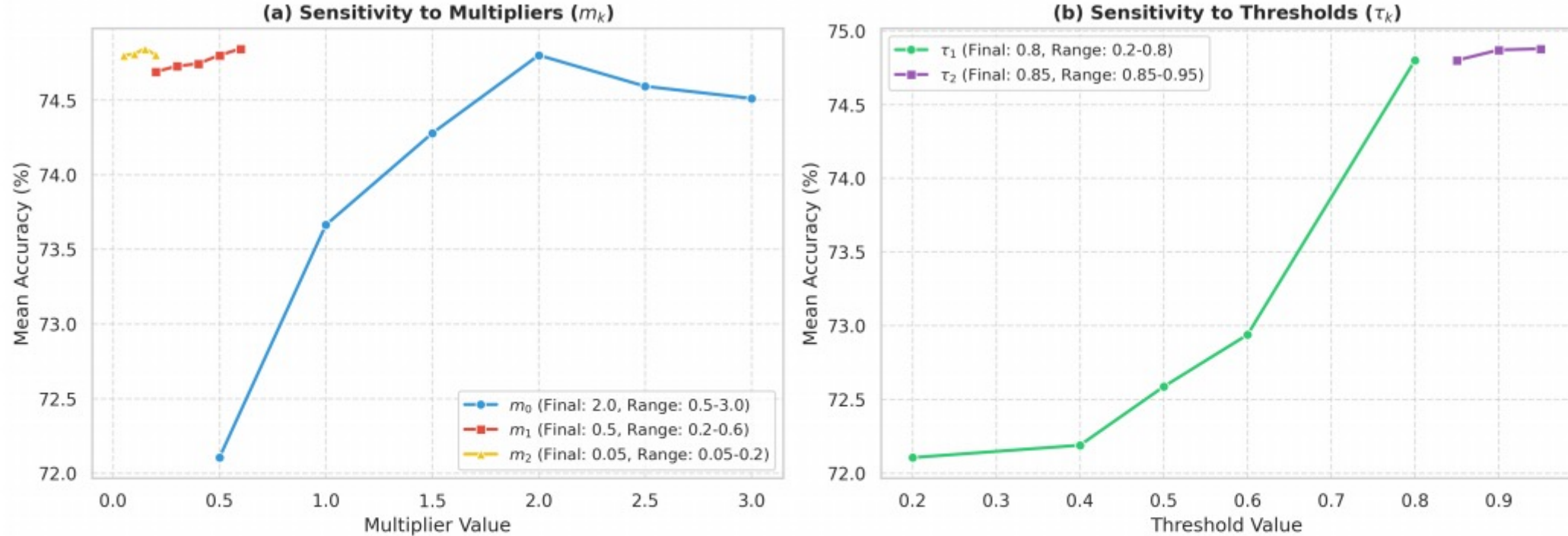


Fig. 5. Sensitivity analysis of individual Adaptive Learning Rate (ALR) parameters on ModelNet-40C. We evaluate (a) Multipliers (m_0, m_1, m_2) and (b) Thresholds (τ_1, τ_2). Values in legend denote the sweep range for each parameter.

Appendix

