Memory-Bounded Continuation of Greedy Sampling for Continual Anomaly Detection

Yoon Gyo Jung*[1], Jaewoo Park*[2], Kuan-Chuan Peng[3],
Seongdeok Bang[2], and Octavia Camps[1]

[1]Northeastern University [2] AIVEX
[3]Mitsubishi Electric Research Laboratories

BMVC 2026

github.com/jungyg/ContCore

Introduction

Q) What does anomaly detection need from memory?
A) A compact yet representative summary of normal data. Greedy sampling builds exactly that coreset.

Q) What breaks when tasks arrive sequentially?
A) Greedy sampling extends naively only with unbounded memory — the coreset grows with the task count T. Prior CAD methods break on complex multi-class tasks (DNE, UCAD) or forget on long schedules (IUF, CDAD).

Q) Can a bounded coreset match an unbounded one?
A) Yes. **Continued greedy sampling** — greedy selection over previously greedy-sampled sets — preserves representativeness under a fixed budget, degrading gracefully instead of catastrophically.

Contributions

- Continued greedy sampling preserves representativeness under strict memory limits, enabling continual AD without unbounded memory growth.
- Theoretical guarantee: the greedy-continued coreset stays within a bounded gap of the oracle coreset.
- ContCore reaches SOTA on 11 task schedules over MVTecAD and VisA, plus cross-dataset transfer.
- Robust in online continual AD, where prior methods degrade significantly.

Method

Greedy sampling on a base set

Greedy sampling collects points maximally distant from those already selected. We extend it with a base set B, so distance is measured against both the sampled set and B:

$$\mathcal{G}_n(\mathcal{Z};B) = \{z_i^* \in \mathcal{Z} : z_i^* = \arg\max_{z\in\mathcal{Z}} d(z, \mathcal{Z}_{i-1}^* \cup B)\}$$

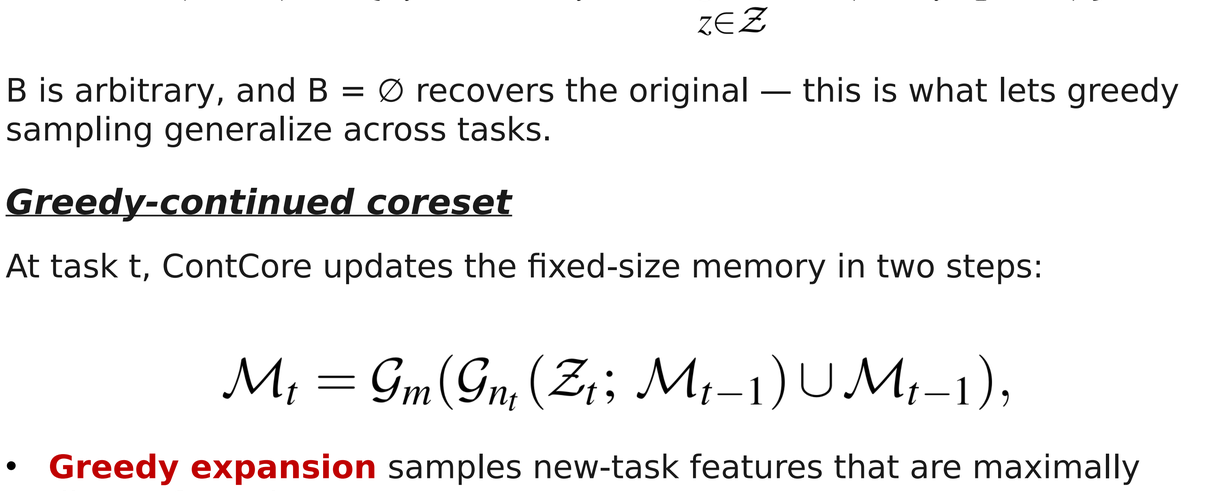
B is arbitrary, and B = ∅ recovers the original — this is what lets greedy sampling generalize across tasks.

Greedy-continued coreset

At task t, ContCore updates the fixed-size memory in two steps:

$$\mathcal{M}_t = \mathcal{G}_m(\mathcal{G}_{n_t}(\mathcal{Z}_t;\, \mathcal{M}_{t-1}) \cup \mathcal{M}_{t-1}),$$

- **Greedy expansion** samples new-task features that are maximally distant from the current coreset.
- **Greedy consolidation** keeps m maximally separated features of the combined set, enforcing the fixed memory budget.
- **In practice**, consolidation is approximated by nearest-neighbour ranking: the fraction q of embeddings with the largest NN distance is kept directly, halving sampling time.
- **At inference**, ContCore follows PatchCore — a patch is scored by its distance to the coreset, so nothing is ever re-trained.



Expansion adds the most distant new-task features (green); consolidation drops the redundant ones (red) to hold the memory budget.

Theoretical Analysis

Theorem 4.1 — bounded gap to the oracle coreset

$$H(\mathcal{O},\, \mathcal{M}_T) \le \varepsilon_o + \max_{1\le k\le T} \left(\varepsilon_k + \sum_{j=k}^{T} \widehat{\varepsilon}_j \right)$$

The greedy-continued coreset stays within a bounded gap of the oracle coreset — greedy sampling run over all task data at once. Error accumulates per task but does not compound catastrophically. Table 7 confirms the bound empirically for 2 ≤ T ≤ 10.

T	$H(\mathcal{O}, \mathcal{M}_T)$	ε_o	$\max_{1\le k\le T}\left(\varepsilon_k+\sum_{j=k}^{T}\widehat{\varepsilon}_j\right)$	Right hand side of Eq. (6)
2	4.87	2.28	4.37	6.65
3	4.72	2.22	5.99	8.21
4	4.59	2.13	7.54	9.67
5	4.50	2.04	9.02	11.06
6	4.44	2.08	10.66	12.74
7	4.35	2.08	12.27	14.35
8	4.35	2.21	13.95	16.16
9	4.31	2.07	15.22	17.29
10	4.36	2.19	16.85	19.04

Empirical tightness of the bound of Eq. (6) on MVTecAD.

Experiments

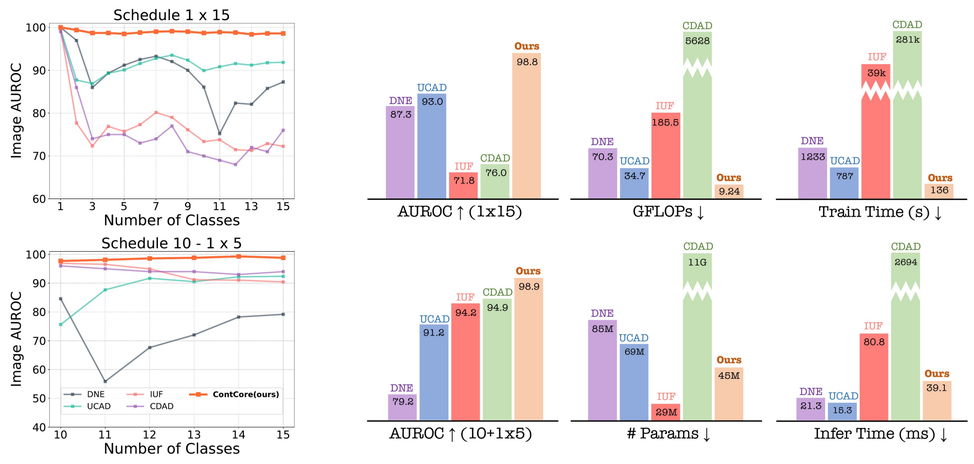
Setup. MVTecAD and VisA over 11 continual task schedules, plus cross-dataset and online CAD. WideResNet50 (layers 2–3), p = 0.01, coreset size m = 20k / 40k, one RTX 3090. We report task-average image / pixel AUROC (↑) and forgetting measure FM (↓).

Method	1 × 15 with 15 Steps		3 × 5 with 5 Steps		10 – 1 × 5 with 6 Steps		10 – 5 with 2 Steps		14 – 1 with 2 Steps	
	AUROC (↑)	FM (↓)	AUROC (↑)	FM (↓)	AUROC (↑)	FM (↓)	AUROC (↑)	FM (↓)	AUROC (↑)	FM (↓)
UniAD NeurIPS'22	77.4 / -	22.9 / -	81.3 / 88.7	7.4 / 10.6	76.6 / 82.3	21.1 / 17.3	86.7 / 91.5	14.9 / 10.6	85.7 / 89.6	18.3 / 13.3
UniAD + EWC PNAS'17	- / -	- / -	79.6 / 89.0	9.5 / 10.1	89.6 / 93.8	5.4 / 3.6	90.5 / 93.6	7.3 / 4.2	92.8 / 95.4	4.1 / 1.9
UniAD + SI PMLR'17	- / -	- / -	81.9 / 88.5	7.0 / 10.8	77.2 / 81.6	20.2 / 18.2	84.1 / 88.3	20.2 / 17.0	85.7 / 89.5	18.4 / 13.4
UniAD + MAS ECCV'18	- / -	- / -	81.5 / 89.0	7.2 / 10.2	77.9 / 82.0	19.5 / 17.7	86.8 / 91.0	14.9 / 11.6	85.8 / 89.6	18.1 / 13.3
UniAD + LVT CVPR'22	- / -	- / -	80.4 / 88.6	8.6 / 10.6	78.2 / 88.3	19.1 / 16.1	87.1 / 90.6	14.1 / 12.3	80.4 / 86.0	29.1 / 20.6
DNE ACMMM'22	87.3 / -	5.3 / -	84.3 / -	1.75 / -	79.2 / -	**0.4** / -	82.8 / -	2.1 / -	79.2 / -	0.1 / -
UCAD AAAI'24	93.0 / 83.3	1.0 / **-0.6**	84.8 / 90.1	10.3 / 9.2	91.2 / 94.0	6.3 / 3.0	88.7 / 93.1	5.2 / 3.7	93.8 / 95.7	1.8 / 0.3
IUF ECCV'24	71.8 / 79.2	5.1 / 6.6	84.2 / 91.1	10.0 / 8.4	94.2 / 95.1	3.2 / 1.0	92.2 / 94.4	9.3 / 4.3	96.0 / 96.3	1.0 / 0.6
CDAD CVPR'25	76.0 / 89.8	10.2 / 5.7	89.1 / 92.2	3.7 / 4.7	94.9 / 95.7	1.0 / 0.7	94.2 / 95.3	2.1 / 2.4	96.4 / 96.6	**-0.6 / -0.04**
ContCore (ours)	**98.8 / 97.4**	**0.1** / 0.2	**98.2 / 96.9**	**0.6 / 0.5**	**98.9 / 96.4**	0.5 / **0.5**	**97.7 / 96.6**	**0.5 / 0.5**	**98.3 / 97.3**	0.0 / 0.1

Best on every MVTecAD schedule, with near-zero forgetting.

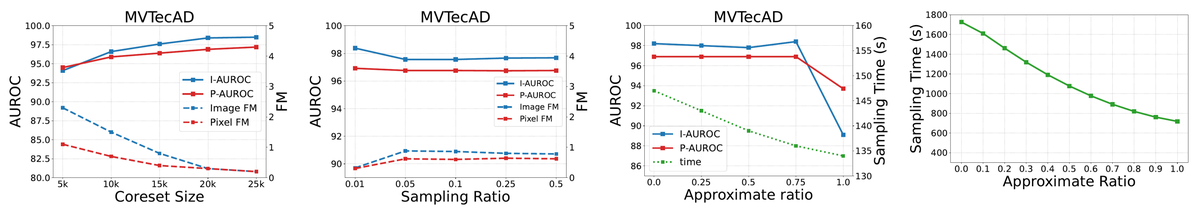
Method	1 × 12 with 12 Steps		8 – 1 × 4 with 5 Stepss		8 – 4 with 2 Steps		11 – 1 with 2 Steps	
	AUROC (↑)	FM (↓)	AUROC (↑)	FM (↓)	AUROC (↑)	FM (↓)	AUROC (↑)	FM (↓)
UniAD NeurIPS'22	63.9 / -	29.7 / -	72.2 / 90.8	16.6 / 9.2	78.1 / 94.0	14.7 / 8.4	75.0 / 792.1	22.4 / 11.4
UniAD + EWC PNAS'17	- / -	- / -	72.3 / 92.3	16.5 / 7.3	80.5 / 95.4	10.0 / 5.3	78.7 / 95.4	14.9 / 4.8
UniAD + SI PMLR'17	- / -	- / -	69.8 / 88.5	19.8 / 12.0	69.8 / 88.5	9.2 / 8.3	78.1 / 92.0	16.9 / 11.5
UniAD + MAS ECCV'18	- / -	- / -	72.1 / 90.6	16.7 / 9.4	72.1 / 90.6	14.1 / 8.4	75.4 / 91.8	21.5 / 11.9
UniAD + LVT CVPR'22	- / -	- / -	70.8 / 94.4	18.3 / 8.4	70.8 / 91.4	13.4 / 8.1	77.5 / 92.3	17.3 / 10.9
DNE (ViT-B/16) ACMMM'22	55.2 / -	11.4 / -	58.6 / -	10.2 / -	64.1 / -	6.1 / -	68.8 / -	1.9 / -
UCAD (ViT+SAM) AAAI'24	78.1 / 76.9	**0.0** / **0.6**	78.8 / 93.5	10.4 / 5.4	79.9 / 94.2	8.9 / 4.8	85.9 / 94.5	2.7 / 0.6
IUF (EfficientNet) ECCV'24	64.7 / 83.6	3.9 / 5.0	79.8 / 95.0	9.8 / 6.8	80.1 / 95.4	9.8 / 6.8	87.3 / **97.6**	2.4 / 1.8
CDAD (SDv1.5) CVPR'25	78.5 / **97.3**	11.1 / 6.0	83.4 / 95.8	3.8 / 1.5	85.3 / **97.1**	1.4 / **0.3**	88.3 / 97.2	**-1.3 / -0.1**
ContCore (ours) (WRN50)	**93.7** / 97.2	0.7 / **0.6**	**96.3 / 97.2**	**0.7 / 0.5**	**93.8** / 96.5	**1.2** / 0.8	**96.1** / 97.7	0.4 / **-0.0**

The same holds on VisA across 4 task schedules.



ContCore stays flat as classes accumulate, and wins on compute and time.

Ablation & Analysis



(a) Coreset size m *(b) Sampling ratio p* *(c) Approx. ratio q* *(d) Sampling time*

Plateaus past m = 20k, insensitive to p, no loss up to q = 0.75.

Sampling Method	Overlapping Samples (↑)	Average Minimum Distance (↓)	Sliced Wasserstein Distance (↓)	Hausdorff Distance (↓)
Random	356	1.17	0.0260	2.55
MGS	3933	1.04	0.0014	1.93

ContCore's coreset lands closer to the PatchCore oracle than random sampling.

Conclusion

- ContCore keeps a greedy-continued coreset under a fixed memory budget, no matter how many tasks arrive.
- SOTA on 11 task schedules, cross-dataset transfer and online CAD — at 9.24 GFLOPs and 754 s of training.