A HYPERPARAMETERS OF DADS

Hyperparamters of DADS are listed in Table A1.

Table A1: Hyperparameters of DADS.

Hyperparameter	Meaning	Value
p	probability of using random sampling	0.7
$[\alpha_1, \alpha_2, \alpha_3]$	values of reward function	[1.0, 5.0, 0.5]
TH_{score}	score threshold	0.1
TH_{conf}	initial confidence threshold	4
$\lvert\mathcal{D}'\rvert$	size of subset used in distance-based sampling	10
n_dup	used in oversampling of labeled anomalies	0.2
s_strength	used in adaptive confidence threshold	1.5
n_steps	number of steps per episode	5000
lr	learning rate	0.001
warmup_steps	steps before learning	20000
batch_size	batch size	64
update_times	number of updates per step	1
τ	soft update parameter	0.2
γ	discount rate	0.99

B ADDITIONAL DETAILS OF ALGORITHM

Oversampling of labeled anomalies. Considering that $\mathcal{D}^a$ usually accounts for a small part of training set $\mathcal{D}$, and anomalies cannot be sampled frequently, which may lead to insufficient supervisory signal. To avoid this problem, oversampling is applied to $\mathcal{D}^a$ in initialization. To be more specific, data in $\mathcal{D}^a$ will be duplicated several times until $\mathcal{D}^a$ accounts for a certain percentage of the whole training set $\mathcal{D}$, we denote this percentage as a hyperparameter n_dup, which is set to 0.2 by default.

Adaptive confidence threshold TH_{conf}. TH_{conf} plays a role in controlling the search intensity, the larger TH_{conf} is, the more steps are needed to search an anomaly from $\mathcal{U}$. The search of anomalies should be accurate and efficient, which requires that the threshold should be set carefully. If threshold is too small, it is likely that normal data will be added to $\mathcal{A}$ and cause contamination. If threshold is too large, no anomalies will be searched, and the whole DADS method is reduced to a method that simply fits the known anomalies.

Here we design an adaptive threshold setting method, which can automatically set threshold TH_{conf} after each episode of training. We set two hyperparameters N_{up} and N_{low}. TH_{conf} will be set to a certain value (e.g. 4) at initialization. After each episode of training, if the number of searched anomalies in the current episode exceeds N_{up}, TH_{conf} will be added by 1 to reduce the risk of $\mathcal{A}$ being polluted. If the number of searched anomalies is below N_{low}, TH_{conf} will be decreased by 1 to increase the probability of the agent finding possible anomalies. For other conditions, TH_{conf} will remain the same. In experiment, $N_{\text{up}} = min(\text{contamination\_ratio}, 0.04) * \text{n\_steps} * \text{s\_search}$, $N_{\text{low}} = N_{\text{up}}/2$, where s_search is a hyperparameter of controlling the strength of search, by default s_search = 1.5.

C BASELINE MODELS IMPLEMENTATION

- DPLAN: we reproduce the method using the same network architecture and hyperparameter setting in the original paper, but considering that the distance calculation step in DPLAN is time-consuming, we decrease the sampling size from 1000 to 100.
- DeepSAD: we use the original implementation in https://github.com/lukasruff/Deep-SAD-PyTorch, HSC loss is used as a further improvement. We set hidden_dims=[32, 16] for cardio, satellite, satimage2, thyroid, and multi_cardio datasets, hidden_dims=[128, 64] for annthyroid, multi_annthyroid, and multi_har datasets.
- DevNet: we use the original implementation in https://github.com/GuansongPang/deviation-network. The network includes one hidden layer with 20 units.
- SSAD: we use the original implementation in https://github.com/nicococo/tilitools.
- Overlap: we use the original implementation in https://github.com/Minqi824/Overlap.
- SRR: We reproduce the method using the same network architecture and hyperparameter setting in the original paper.
- LatentOE: we use the original implementation in https://github.com/boschresearch/LatentOE-AD.git.
- Iforest: we use the IForest model in PYOD (https://github.com/yzhao062/pyod) package, and set max_samples=256 and n_estimators=100.

D THE SAC ALGORITHM

As described in Section 2.1, the objective that SAC aims to maximize can be written as

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\tau_\pi} \gamma^t \Big[r\left(s_t, a_t\right) + \alpha \mathcal{H}\big(\pi(\cdot \mid s_t)\big) \Big].$$

To find the optimal policy, SAC concurrently learns a policy network π_θ and two Q-networks Q_{ϕ_1}, Q_{ϕ_2}. Two target Q-networks $Q_{\phi_{\text{targ},1}}, Q_{\phi_{\text{targ},2}}$ are included to stabilize the training, with parameters initialized equal to Q_{ϕ_1}, Q_{ϕ_2}.

Given a batch of transitions $\mathcal{D} = \left\{(s_i, a_i, r_i, s_i', d_i), i = 1, 2, \cdots, n\right\}$ sampled from the replay buffer, where s_i is state, a_i is action, r_i is reward, s_i' is next state, d_i is termination flag. The loss functions of two Q-networks are:

$$L\left(\phi_i\right) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}} \left[\left(Q_{\phi_i}(s,a) - y\left(r, s', d\right) \right)^2 \right],$$

and the target is given by:

$$y\left(r, s', d\right) = r + \gamma(1-d) \left(\min_{j=1,2} Q_{\phi_{\text{targ},j}}\left(s', \tilde{a}'\right) - \alpha \log \pi_\theta\left(\tilde{a}' \mid s'\right) \right),$$

where $\tilde{a}' \sim \pi_\theta\left(\cdot \mid s'\right)$.

The way of optimizing the policy makes use of the reparameterization trick, in which the policy takes the following form:

$$\tilde{a}_\theta(s, \xi) = \tanh\left(\mu_\theta(s) + \sigma_\theta(s) \odot \xi\right), \quad \xi \sim \mathcal{N}(0, I).$$

Then the policy network is optimized to minimize

$$\max_\theta \underset{\substack{s\sim\mathcal{D} \\ \xi\sim\mathcal{N}}}{} \left[\min_{j=1,2} Q_{\phi_j}\left(s, \tilde{a}_\theta(s,\xi)\right) - \alpha \log \pi_\theta\left(\tilde{a}_\theta(s,\xi) \mid s\right) \right].$$

Two target Q-networks are updated in a soft way, in which

$$\phi_{\text{targ},i} \Longleftarrow \tau * \phi_{\text{targ},i} + (1-\tau) * \phi_i.$$

E DATASETS

Detailed information of datasets are listed in Tables A2 and A3. The last column of Table A3 indicates whether the class is known or not. If the anomaly class is known, it will appear in both anomaly dataset and unlabeled dataset, else it only exists in unlabeled dataset. All datasets are available through links in Table A4.

Table A2: Datasets with single anomaly class.

Dataset			Normal Class	Anomaly Class
Name	Size	Dim		
annthyroid	7200	6	6666(92.6%)	534(7.4%)
cardio	1831	21	1655(90.4%)	176(9.6%)
satellite	6435	36	4399(68.4%)	2036(31.6%)
satimage2	5803	36	5732(98.8%)	71(1.2%)
thyroid	3772	6	3679(97.5%)	93(2.5%)

Table A3: Datasets with multiple anomaly classes.

Dataset			Normal Classes		Anomaly Classes		
Name	Size	Dim	Name	Size	Name	Size	Known
shuttle	58000	9	rad flow	45586(78.6%)	fpv close	50(0.1%)	N
					fpv open	171(0.3%)	N
					high	8903(15.4%)	Y
					bypass	3267(5.6%)	N
					bpv close	10(0.02%)	N
					bpv open	13(0.02%)	N
cardio	2126	21	normal	1655(77.8%)	suspect	295(13.9%)	Y
					pathologic	176(8.3%)	N
har	10299	561	walking, sitting standing, laying	7349(71.3%)	upstairs	1544(15.0%)	Y
					downstairs	1406(13.7%)	N
annthyroid	3772	21	normal	3488(92.4%)	hypothyroid	93(2.5%)	N
					subnormal	191(5.1%)	Y

Table A4: Links of Datasets.

Datsaset	link
annthyroid	http://odds.cs.stonybrook.edu/annthyroid-dataset/
cardio	http://odds.cs.stonybrook.edu/cardiotocogrpahy-dataset/
satellite	http://odds.cs.stonybrook.edu/satellite-dataset/
satimage2	http://odds.cs.stonybrook.edu/satimage-2-dataset/
thyroid	http://odds.cs.stonybrook.edu/thyroid-disease-dataset/
multi_annthyroid	https://www.openml.org/d/40497
multi_cardio	https://archive.ics.uci.edu/ml/datasets/Cardiotocography
multi_har	https://www.openml.org/d/1478
multi_shuttle	https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)

F ADDITIONAL ABLATION STUDIES

Except TH_{conf}, p and α_2, we conduct ablation studies on four more hyperparameters, respectively TH_{score}, α_3, n_dup and s_search. Consistent with the ablation study in the main paper, we set contamination_ratio = 0.04, anomalies_ratio = 0.05.

F.1 Ablation on TH_{score}

We vary TH_{score} from 0.1 to 0.9. Result indicates that a threshold that is either too high or too low can have an negative impact on the performance, as the score range of abnormal or normal data is excessively compressed.

F.2 Ablation on α_3

We vary α_3 from 0.1 to 0.9. Result shows that the best performing α_3 varies across datasets, but overall, a medium value would achieve the best average performance. This result is consistent with the ablation study on α_2, that the intrinsic reward should be set properly. A value that is too small can not efficiently assist the learning, while a too large value will lead to a biased policy.

Table A5: Ablation study of TH_{score} (AUC-PR)

	$TH_{\text{score}} = 0.1$	$TH_{\text{score}} = 0.3$	$TH_{\text{score}} = 0.5$	$TH_{\text{score}} = 0.7$	$TH_{\text{score}} = 0.9$
multi_shuttle	0.989 ±0.007	**0.991 ±0.004**	**0.991 ±0.005**	**0.991 ±0.004**	0.990 ±0.004
multi_cardio	0.858 ±0.049	**0.890 ±0.052**	0.888 ±0.043	0.863 ±0.057	0.827 ±0.064
multi_har	0.908 ±0.067	0.909 ±0.061	0.923 ±0.059	**0.934 ±0.047**	0.926 ±0.048
multi_annthyroid	0.608 ±0.115	0.607 ±0.187	0.650 ±0.147	**0.652 ±0.130**	**0.652 ±0.098**
Average	0.841 ±0.060	0.849 ±0.076	0.863 ±0.064	**0.866 ±0.064**	0.849 ±0.054

Table A6: Ablation study of α_3 (AUC-PR)

	$\alpha_3 = 0.1$	$\alpha_2 = 0.3$	$\alpha_2 = 0.5$	$\alpha_2 = 0.7$	$\alpha_2 = 0.9$
multi_shuttle	0.977 ±0.007	0.985 ±0.008	0.991 ±0.004	0.993 ±0.002	**0.994 ±0.001**
multi_cardio	**0.886 ±0.045**	0.864 ±0.063	0.863 ±0.057	0.858 ±0.064	0.863 ±0.065
multi_har	0.881 ±0.052	0.911 ±0.057	0.934 ±0.047	0.936 ±0.052	**0.946 ±0.039**
multi_annthyroid	0.661 ±0.104	**0.679 ±0.126**	0.676 ±0.148	0.652 ±0.130	0.623 ±0.172
Average	0.851 ±0.052	0.860 ±0.064	**0.866 ±0.064**	0.860 ±0.062	0.856 ±0.069

F.3 Ablation on n_dup

As introduced in Appendix B, n_dup is responsible for the oversampling of labeled anomalies. The larger n_dup is, the more labeled anomalies will be sampled during training. As a result, adjusting n_dup is equivalent to paying more attention to labeled anomalies. From the result we can see that both lack of labels or too many labels will do harm to the learning.

Table A7: Ablation study of n_dup (AUC-PR)

	n_dup=0.05	n_dup=0.1	n_dup=0.2	n_dup=0.3	n_dup=0.4
multi_shuttle	**0.993 ±0.001**	0.992 ±0.003	0.991 ±0.004	0.987 ±0.007	0.986 ±0.007
multi_cardio	0.872 ±0.061	**0.896 ±0.045**	0.863 ±0.057	0.865 ±0.053	0.876 ±0.060
multi_har	0.943 ±0.028	**0.944 ±0.030**	0.934 ±0.047	0.908 ±0.058	0.870 ±0.067
multi_annthyroid	0.537 ±0.220	**0.708 ±0.082**	0.676 ±0.148	0.691 ±0.113	0.652 ±0.127
Average	0.837 ±0.078	**0.885 ±0.040**	0.866 ±0.064	0.863 ±0.058	0.846 ±0.065

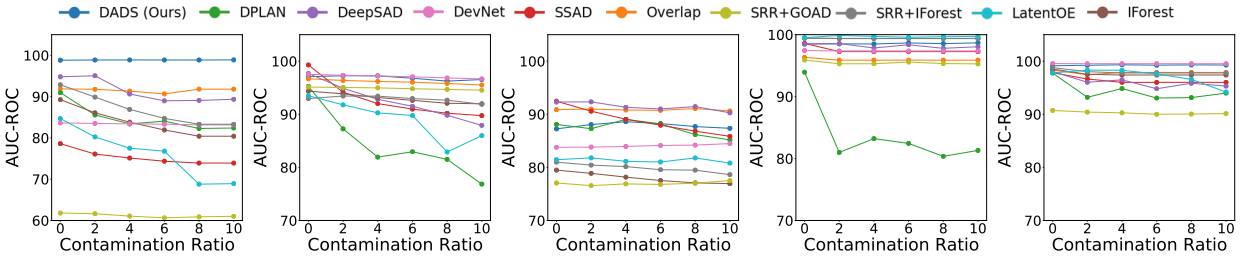
F.4 Ablation on s_search

As stated in Appendix B, s_search is a hyperparameter responsible for adaptively adjusting TH_{conf}. The larger s_search is, the more possible anomalies will be searched during training. We vary s_search within range $[0.5, 2.5]$. On one hand, if s_search is too small (e.g. 0.5), the search mechanism of DADS can not fully exist its effect, which is consitent with the ablation study of TH_{conf} in the main paper. On the other hand, if s_search is too large (e.g. 2.5), the overly aggressive search will pollute the labeled anomalies, thus lowering the performance.

Table A8: Ablation study of s_search (AUC-PR)

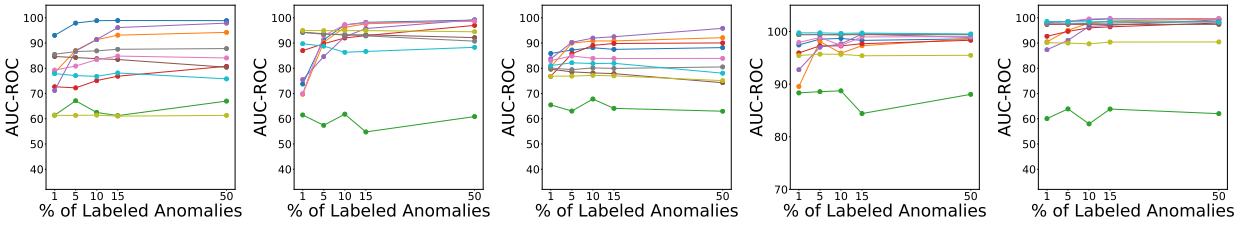
	s_search=0.5	s_search=1.0	s_search=1.5	s_search=2.0	s_search=2.5
multi_shuttle	0.991 ±0.004	0.991 ±0.004	0.991 ±0.004	**0.992 ±0.004**	0.990 ±0.004
multi_cardio	0.866 ±0.052	0.875 ±0.063	0.863 ±0.057	**0.881 ±0.056**	0.857 ±0.048
multi_har	0.909 ±0.056	**0.940 ±0.041**	0.934 ±0.047	0.902 ±0.071	0.912 ±0.062
multi_annthyroid	0.647 ±0.166	0.646 ±0.127	0.676 ±0.148	**0.681 ±0.126**	0.641 ±0.153
Average	0.853 ±0.070	0.863 ±0.059	**0.866 ±0.064**	0.864 ±0.064	0.850 ±0.067

G AUC-ROC PERFORMANCE OF DADS AND BASELINES



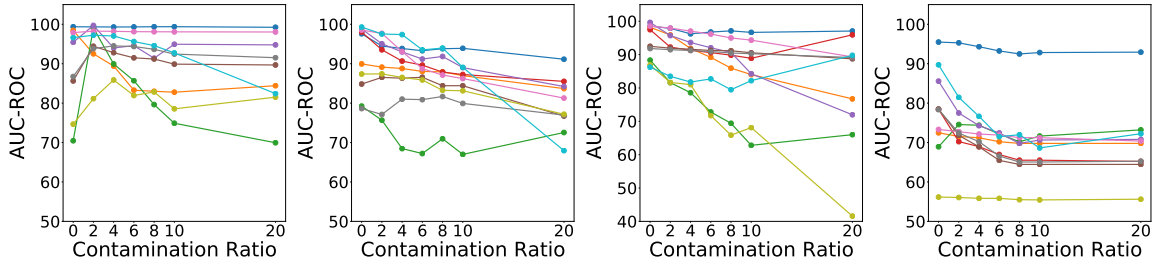
(a) annthyroid (b) cardio (c) satellite (d) satimage2 (e) thyroid

Figure 1: AUC-ROC of DADS and baselines in setting 1.1.



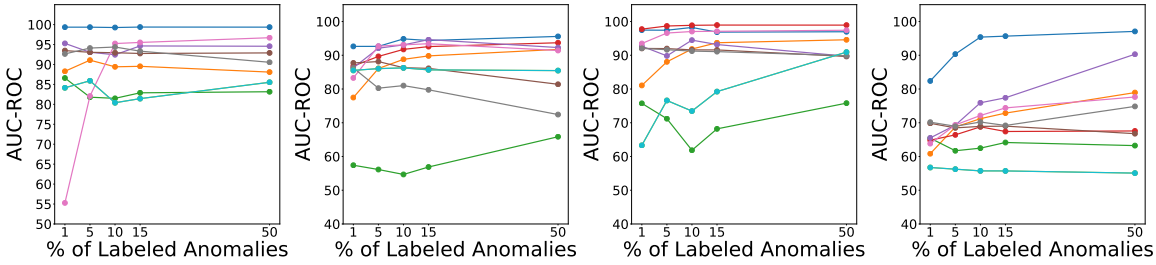
(a) annthyroid (b) cardio (c) satellite (d) satimage2 (e) thyroid

Figure 2: AUC-ROC of DADS and baselines in setting 1.2.



(a) multi_shuttle (b) multi_cardio (c) multi_har (d) multi_annthyroid

Figure 3: AUC-ROC of DADS and baselines in setting 2.1.



(a) multi_shuttle (b) multi_cardio (c) multi_har (d) multi_annthyroid

Figure 4: AUC-ROC of DADS and baselines in setting 2.2.