ESAD: End-to-end Semi-supervised Anomaly Detection — Supplementary Material

Chaoqin Huang[1,2]
huangchaoqin@sjtu.edu.cn
Fei Ye[1]
yf3310@sjtu.edu.cn
Peisen Zhao[1,3]
pszhao@sjtu.edu.cn
Ya Zhang ✉[1,2]
ya_zhang@sjtu.edu.cn
Yanfeng Wang[1,2]
wangyanfeng@sjtu.edu.cn
Qi Tian[3]
tian.qi1@huawei.com

[1] Cooperative Medianet Innovation Center,
Shanghai Jiao Tong University
[2] Shanghai AI Laboratory
[3] Huawei Cloud & AI

1 Supplementary Proofs

Proposition 1 *If* $\mathrm{KL}\left[p_N(X,Z)||p_A(X,Z)\right]$ *is maximized, then it is equivalent that* $\mathrm{KL}\left[p_N(X)||p_A(X)\right]$ *and* $\mathrm{KL}\left[p_N(Z|X)||p_A(Z|X)\right]$ *are maximized.*

Proof 1 *The KL divergence for the joint distributions can be decomposed with the chain rule [5]:*

$$\begin{aligned}
&\mathrm{KL}\left[p_N(X,Z)||p_A(X,Z)\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(X,Z)}{p_A(X,Z)}\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(X)}{p_A(X)}+\log\frac{p_N(Z|X)}{p_A(Z|X)}\right] \\
=&\mathrm{KL}\left[p_N(X)||p_A(X)\right]+\mathbb{E}_{p_N(X)}\left[\mathrm{KL}\left[p_N(Z|X)||p_A(Z|X)\right]\right].
\end{aligned}$$

To maximize the KL divergence for the joint distributions, it is equivalent that we maximize the KL divergence for both marginal and conditional distributions [5].

Proposition 2 *Let* $I(X_N,Z_N)$ *denotes the mutual information between* X_N *and* Z_N*;* $H(Z_N)$ *denotes the entropy of* Z_N*;* $H(p_N(Z|X),p_A(Z|X))$ *denotes the cross-entropy between* $p_N(Z|X)$ *and* $p_A(Z|X)$*;* $KL\left[\mathrm{p}_N(X)\,||\,\mathrm{p}_A(X)\right]$ *denotes the KL divergence between* $p_N(X)$ *and* $p_A(X)$*. Then:*

$$\begin{aligned}
&\mathrm{KL}\left[p_N(X,Z)||p_A(X,Z)\right] \\
&=\mathrm{I}(X_N,Z_N)-\mathrm{H}(Z_N)+\mathbb{E}_{p_N(X)}\left[H(p_N(Z|X),p_A(Z|X))\right]+KL\left[\mathrm{p}_N(X)||\mathrm{p}_A(X)\right].
\end{aligned} \tag{1}$$

Proof 2 *The KL divergence can be reformulated as:*

$$\begin{aligned}
&\mathrm{KL}\left[p_N(X,Z)||p_A(X,Z)\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(X,Z)}{p_A(X,Z)}\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(Z|X)\cdot p_N(X)}{p_A(Z|X)\cdot p_A(X)}\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(Z|X)\cdot p_N(X)\cdot p_N(Z)}{p_A(Z|X)\cdot p_A(X)\cdot p_N(Z)}\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\left(\frac{p_N(Z|X)}{p_N(Z)}\cdot p_N(Z)\cdot\frac{1}{p_A(Z|X)}\cdot\frac{p_N(X)}{p_A(X)}\right)\right].
\end{aligned}$$

The above formula is decomposed into four components. The first term refers to the mutual information between the original data X_N and its latent representation Z_N:

$$\begin{aligned}
&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(Z|X)}{p_N(Z)}\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(Z|X)\cdot p_N(X)}{p_N(X)\cdot p_N(Z)}\right] \\
=&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(X,Z)}{p_N(X)\cdot p_N(Z)}\right] \\
=&I(X_N,Z_N).
\end{aligned}$$

The second term refers to the negative entropy of Z_N:

$$\mathbb{E}_{p_N(X,Z)}\left[\log p_N(Z)\right]=-\mathbb{E}_{p_N(Z)}\left[\log\frac{1}{p_N(Z)}\right]=-\mathrm{H}(Z_N).$$

The third term refers to the expected value of the cross entropy between the conditional distributions $p_A(Z|X)$ and $p_N(Z|X)$:

$$\begin{aligned}
&\mathbb{E}_{p_N(X,Z)}\left[\log\frac{1}{p_A(Z|X)}\right] \\
=&\mathbb{E}_{p_N(X)}\mathbb{E}_{p_N(Z|X)}\left[-\log p_A(Z|X)\right] \\
=&\mathbb{E}_{p_N(X)}\left[H(p_N(Z|X),p_A(Z|X))\right].
\end{aligned}$$

The fourth term is a constant, since $p_N(X)$ and $p_A(X)$ are fixed when the dataset is given:

$$\mathbb{E}_{p_N(X,Z)}\left[\log\frac{p_N(X)}{p_A(X)}\right]=\mathrm{KL}\left[\mathrm{p}_N(X)||\mathrm{p}_A(X)\right]=\mathrm{C}.$$

Thus the KL divergence can be reformulated as:

$$\begin{aligned}
&\mathrm{KL}\left[p_N(X,Z)||p_A(X,Z)\right] \\
=&\mathrm{I}(X_N,Z_N)-\mathrm{H}(Z_N)+\mathbb{E}_{p_N(X)}\left[H(p_N(Z|X),p_A(Z|X))\right]+KL\left[\mathrm{p}_N(X)||\mathrm{p}_A(X)\right].
\end{aligned}$$

Proposition 3 *The third term in the objective Eq. (1), i.e., $\mathbb{E}_{p_N(X)}\left[H(p_N(Z|X),p_A(Z|X)\right]$, is non-negative.*

Proof 3 *We assume that $p_N(Z|X)$ and $p_A(Z|X)$ are separable at the latent space,* i.e., *for $X,Z \sim p_N(X,Z)$, the evaluated density $\log p_A(Z|X) \leqslant 0$. This assumption is indeed consist with the fundamental assumption in [2]: data can be embedded into a certain representation space where normal instances and anomalies appear significantly different. With the above assumption, $\mathbb{E}_{p_N(X)}[H(p_N(Z|X), p_A(Z|X)]$ is shown to be non-negative:*

$$\begin{aligned} &\inf \mathbb{E}_{p_N(X)}[H(p_N(Z|X), p_A(Z|X)] \\ =&\inf \mathbb{E}_{p_N(X,Z)}[-\log p_A(Z|X)] \\ \geqslant&\mathbb{E}_{p_N(X,Z)}[\inf(-\log p_A(Z|X))] \\ \geqslant&0. \end{aligned}$$

Proposition 4 *Assuming Z follows an isotropic Gaussian, with mean μ, covariance Σ and $Z \subseteq \mathbb{R}^d$, the entropy of Z,* i.e., *$\mathrm{H}(Z)$, is proportional to its log-variance for a fixed dimensionality d, without dependence on its mean μ.*

Proof 4 *For Z with covariance Σ and $Z \subseteq \mathbb{R}^d$,*

$$\mathrm{H}(Z) = \mathbb{E}[-\log p(Z)] = -\int_Z p(Z)\log p(Z)\mathrm{d}Z \leq \frac{1}{2}\log((2\pi e)^d \det\Sigma),$$

which holds with equality iff Z is jointly Gaussian [4]. Assuming Z follows an isotropic Gaussian, $Z \sim N(\mu, \sigma^2 I)$ with $\sigma > 0$, we get,

$$\mathrm{H}(Z) = \frac{1}{2}\log((2\pi e)^d \det\sigma^2 I) = \frac{d}{2}(1+\log(2\pi\sigma^2)) \propto \log\sigma^2,$$

which shows that the entropy of Z is proportional to its log-variance for a fixed dimensionality d. The above proof has no dependence on the mean μ.

2 Analysis of Deep SAD

Deep SAD builds upon Infomax principle, which maximizes the mutual information $I(X,Z)$ between data and latent representations with regularization on the representations. The objective function for Deep SAD is formulated as:

$$\max_{\theta}\ I(X,Z) + \beta(H(Z_A) - H(Z_N)), \tag{2}$$

where regularization is enforced through entropy. For $\forall \mathbf{x} \in \mathcal{X}$, Deep SAD adopts an autoencoder consisting of an encoder $Enc(\cdot)$ and a decoder $Dec(\cdot)$: $\mathbf{z} = Enc(\mathbf{x})$, $\hat{\mathbf{x}} = Dec(\mathbf{z})$, where $\hat{\mathbf{x}}$ is the reconstructed sample and $\mathbf{z}$ is the corresponding latent representation, and takes the following two-step process to implement the above objective function.

(i) Autoencoder Pre-training: To *maximize the mutual information* between the data and the latent representations, a reconstruction loss is adopted to pre-train the autoencoder:

$$\mathcal{L}_{rec} = \frac{1}{n+m}\sum_{i=1}^{n+m}\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|^2, \tag{3}$$

where $\mathbf{x}_1, \cdots, \mathbf{x}_{n+m} \in \mathcal{X}$.

(ii) Encoder Fine-turning: To further regularize the entropy of the latent representations, the encoder is fine-turned with an SVDD loss,

$$\mathcal{L}_{SVDD} = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{z}_i^u - \mathbf{c}\|_2 + \frac{\eta}{m}\sum_{j=1}^{m}\|\mathbf{z}_j^l - \mathbf{c}\|_2^{y_j}, \tag{4}$$

where $\mathbf{z}_1^u, \cdots, \mathbf{z}_n^u \in \mathcal{Z}$ are the corresponding latent representations of unlabeled samples $\mathbf{x}_1^u, \cdots, \mathbf{x}_n^u$, $\mathbf{z}_1^l, \cdots, \mathbf{z}_m^l \in \mathcal{Z}$ are the corresponding latent representations of labeled samples $\mathbf{x}_1^l, \cdots, \mathbf{x}_m^l$, and η is set as 1. The hypersphere center $\mathbf{c}$ is set as the mean of the outputs obtained from a forward pass of the encoder for all the data. In fact, Deep SAD does not use the coefficient β in Eq. (2), because the two terms are separately optimized in two stages.

We now argue that the reason for the two-stage implementation of Deep SAD is the contradiction between the optimization of mutual information and entropy. For example, when the latent representations have extremely low entropy, especially zero in the extreme case, the model can be considered as mapping all data into a constant in which the mutual information is restricted to zero, which contradicts with the mutual information maximization. The two-stage implementation for Deep SAD avoids directly facing the above contradiction.

Table 1: Model Structure of ESAD.

Layer	Input	Output
$3\times3\times64$	x $(3\times H\times W)$	x_{0-1} $(64\times H\times W)$
$3\times3\times64$	x_{0-1}	x_{0-2} $(64\times H\times W)$
MaxPool	x_{0-2}	x_{1-1} $(64\times 1/2H\times 1/2W)$
$3\times3\times128$	x_{1-1}	x_{1-2} $(128\times 1/2H\times 1/2W)$
$3\times3\times128$	x_{1-2}	x_{1-3} $(128\times 1/2H\times 1/2W)$
MaxPool	x_{1-3}	x_{2-1} $(128\times 1/4H\times 1/4W)$
$3\times3\times256$	x_{2-1}	x_{2-2} $(256\times 1/4H\times 1/4W)$
$3\times3\times256$	x_{2-2}	x_{2-3} $(256\times 1/4H\times 1/4W)$
MaxPool	x_{2-3}	x_{3-1} $(256\times 1/8H\times 1/8W)$
$3\times3\times512$	x_{3-1}	x_{3-2} $(256\times 1/8H\times 1/8W)$
$3\times3\times512$	x_{3-2}	x_{3-3} $(256\times 1/8H\times 1/8W)$
MaxPool	x_{3-3}	x_{4-1} $(256\times 1/8H\times 1/16W)$
$3\times3\times512$	x_{4-1}	x_{4-2} $(512\times 1/16H\times 1/16W)$
$3\times3\times512$	x_{4-2}	z $(512\times 1/16H\times 1/16W)$
UpSample	z	up_{3-1} $(512\times 1/8H\times 1/8W)$
$3\times3\times256$	up_{3-1}	up_{3-2} $(256\times 1/8H\times 1/8W)$
$3\times3\times256$	up_{3-2}	up_{3-3} $(256\times 1/8H\times 1/8W)$
UpSample	up_{3-3}	up_{2-1} $(256\times 1/4H\times 1/4W)$
$3\times3\times128$	up_{2-1}	up_{2-2} $(128\times 1/4H\times 1/4W)$
$3\times3\times128$	up_{2-2}	up_{2-3} $(128\times 1/4H\times 1/4W)$
UpSample	up_{2-3}	up_{1-1} $(128\times 1/2H\times 1/2W)$
$3\times3\times64$	up_{1-1}	up_{1-2} $(64\times 1/2H\times 1/2W)$
$3\times3\times64$	up_{1-2}	up_{1-3} $(64\times 1/2H\times 1/2W)$
UpSample	up_{1-3}	up_{0-1} $(64\times H\times W)$
$3\times3\times64$	up_{0-1}	up_{0-2} $(64\times H\times W)$
$3\times3\times64$	up_{0-2}	up_{0-3} $(64\times H\times W)$
$3\times3\times3$	up_{0-3}	$\hat{x}$ $(3\times H\times W)$
$3\times3\times64$	$\hat{x}$	x_{5-1} $(64\times H\times W)$
$3\times3\times64$	x_{5-1}	x_{5-2} $(64\times H\times W)$
MaxPool	x_{5-2}	x_{6-1} $(64\times 1/2H\times 1/2W)$
$3\times3\times128$	x_{6-1}	x_{6-2} $(128\times 1/2H\times 1/2W)$
$3\times3\times128$	x_{6-2}	x_{6-3} $(128\times 1/2H\times 1/2W)$
MaxPool	x_{6-3}	x_{7-1} $(128\times 1/4H\times 1/4W)$
$3\times3\times256$	x_{7-1}	x_{7-2} $(256\times 1/4H\times 1/4W)$
$3\times3\times256$	x_{7-2}	x_{7-3} $(256\times 1/4H\times 1/4W)$
MaxPool	x_{7-3}	x_{8-1} $(256\times 1/8H\times 1/8W)$
$3\times3\times512$	x_{8-1}	x_{8-2} $(256\times 1/8H\times 1/8W)$
$3\times3\times512$	x_{8-2}	x_{8-3} $(256\times 1/8H\times 1/8W)$
MaxPool	x_{8-3}	x_{9-1} $(256\times 1/8H\times 1/16W)$
$3\times3\times512$	x_{9-1}	x_{9-2} $(512\times 1/16H\times 1/16W)$
$3\times3\times512$	x_{9-2}	$\hat{z}$ $(512\times 1/16H\times 1/16W)$

3 Model Architecture and Training Details

The model architecture for ESAD is shown in Table 1. For the training, we use stochastic gradient descent (SGD) [3] optimizer with default hyperparameters in Pytorch. ESAD is trained using a batch size of 32 for 200 epochs with NVIDIA GTX 2080Ti. The learning rate is initially set as 0.1, and is divided by 2 every 50 epoch.

Table 2: Classic anomaly detection benchmarks [18].

Dataset	Numbers	Dimensions	#outliers (%)
arrhythmia	452	274	66 (14.6%)
cardio	1,831	21	176 (9.6%)
satellite	6,435	36	2,036 (31.6%)
satimage-2	5,803	36	71 (1.2%)
shuttle	49,097	9	3,511 (7.2%)
thyroid	3,772	6	93 (2.5%)

4 Datasets

Natural Image Datasets. MNIST [14], a dataset consists of 70,000 28×28 handwritten grayscale digit images; Fashion-MNIST [29], a relatively new dataset comprising 28×28 grayscale images of 70,000 fashion products from 10 categories, with 7,000 images per category; CIFAR-10 [13], a dataset consists of 60,000 32×32 RGB images of 10 classes, with 6,000 images for per class.

Medical Image Datasets. Following [24, 26], we examine the detection of metastases in H&E stained images of lymph nodes in Camelyon16 [2] and the recognition of fourteen diseases on the chest X-rays in the NIH dataset [28].

For the NIH dataset, images without any disease marker were considered normal. Pulmonary and cardiac abnormalities in this dataset include atelectasis, effusion, infiltration, mass, nodule, pneumonia, pneumothorax, consolidation, edema, emphysema, fibrosis, pleural thickening, hernia and cardiomegaly, which are all considered anomalous. Following [24, 26], we split the dataset into two sub-datasets having only posteroanterior (PA) or anteroposterior (AP) projections. Note that in the training set, the ratios of labeled anomalous samples are 3.9% for AP and 3.3% for PA. We also experiment on a subset containing clearer normal/anomalous cases [24]. This subset consists of 5110 normal and 857 anomalous images for training, and 677 normal and 677 anomalous images for testing.

For the Camelyon16 dataset, we sample the Vahadane-normalized [27] 64×64 tiles from the fully normal slides with magnification of $10\times$, and treat these as normal. Tiles with metastases are treated as anomalous. It contains 7612 normal and 200 anomalous training images, and 4000 (normal) + 817 (anomalous) images for the test.

Classic anomaly detection benchmark datasets. We use six non-image classic anomaly detection benchmark datasets [18]. Following [20], for the evaluation, we consider random train-to-test set splits of 60:40 while maintaining the original proportion of anomalies in each set. The supplementary details of the classic anomaly detection benchmarks [18] are shown in Table 2.

5 Competing Methods

We consider several shallow unsupervised methods, deep unsupervised anomaly detection competitors and semi-supervised anomaly detection approaches as baselines. Complete details are shown as follows:

(1) OC-SVM/SVDD [23, 25]: The OC-SVM and SVDD are equivalent for the Gaussian/RBF kernel. OC-SVM/SVDD here have unfair advantages by selecting their hyperparameters to maximize AUC on a subset (10%) of the test set to establish a strong baseline. The RBF scale parameters $\gamma \in \{2^{-7}, 2^{-6}, \ldots 2^{2}\}$ are considered and the best performing one is selected.

Table 3: Average area under the ROC curve (AUC) in % on natural image datasets, comparing with unsupervised anomaly detection methods. "†" denotes the highest test AUC among multiple running for the strong baselines. "∗" denotes the highest test AUC among all training epochs for the stronger baselines. We report the results of unsupervised ESAD, where we ignore the labeled data in the training set. Emphasizing that ESAD focuses on the semi-supervised setting but not the unsupervised setting.

Method	MNIST	F-MNIST	CIFAR-10
CAE [16]	92.9 ± 5.7	90.2 ± 5.8	56.2 ± 13.2
IF Hybrid [15]	90.5 ± 5.3	82.5 ± 8.1	59.9 ± 6.7
Deep SVDD [19]	92.8 ± 4.9	89.2 ± 6.2	60.9 ± 9.4
AnoGAN† [23]	93.7	-	61.2
ALOCC* [21]	93.3	-	62.2
ADGAN* [1]	94.7	88.4	62.4
OC-SVM Hybrid [23]	96.3 ± 2.5	91.2 ± 4.7	63.8 ± 9.0
OCGAN† [24]	97.5	-	65.6
GANomaly* [4]	92.8	80.9	69.5
P-KDGAN† [30]	97.8	-	73.8
DGEO† [10]	98.0	93.5	86.0
ESAD (unsupervised)	98.5 ± 1.3	94.0 ± 4.5	78.8 ± 6.5
ESAD	99.6 ± 0.3	95.9 ± 4.0	88.5 ± 6.9

Then the best final results are reported over ν-parameter, where $\nu \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$.
(2) Isolation Forest [15]: The number of trees is set as $t = 100$ and the sub-sampling size is set as $\psi = 256$ as recommended in the original work.
(3) SSAD [11]: SSAD also have the unfair advantages the same as OC-SVM/SVDD. The scale parameters γ of the RBF kernel are selected from $\gamma \in \{2^{-7}, 2^{-6}, \ldots 2^{2}\}$ and then report the best performing one. Otherwise we set the hyperparameters as recommend by the original authors to $\kappa = 1, \kappa = 1, \eta_u = 1$, and $\eta_l = 1$ [11].
(4) Convolutional Autoencoder (CAE) [16]: The autoencoders are trained on the MSE reconstruction loss that also serves as the anomaly score.
(5) Deep SVDD [19]: Both variants, Soft-Boundary Deep SVDD and One-Class Deep SVDD are considered as unsupervised baselines and always report the better performance as the unsupervised result. For Soft-Boundary Deep SVDD, The radius R on every mini-batch is optimally solved. For Deep SVDD, all the bias terms from a network are removed to prevent a hypersphere collapse as recommended by the authors in the original work [19].
(6) SS-DGM [17]: We consider both the $M2$ and $M1 + M2$ model and always report the better performing result. Other settings are following the original work [17].
(7) Deep SAD [20]: The results are borrow from [20]. We set $\lambda = 10^{-6}$ and equally weight the unlabeled and labeled examples by setting $\eta = 1$ if not reported otherwise.

To establish hybrid methods, we apply the OC-SVM, IF, and SSAD to the resulting bottleneck representations given by the respective converged autoencoders. To complete the full learning spectrum, we also include a fully supervised deep classifier trained on the binary cross-entropy loss.

6 Supplementary Experimental Results

Besides the experiments in the main paper, we examine three scenarios [20] in which we vary the following three experimental parameters: (i) γ_l, the ratio of labeled samples in the training data; (ii) γ_p, the ratio of pollution, i.e., unknown anomalies, in the unlabeled training data, and (iii) the number of anomaly classes k_l included in the labeled training data.

Table 4: Complete results of **experimental scenario (ii)**, where we pollute the unlabeled part of the training set with (unknown) anomalies. We report the avg. AUC in % with st. dev. computed over 90 experiments at various ratios γ_p.

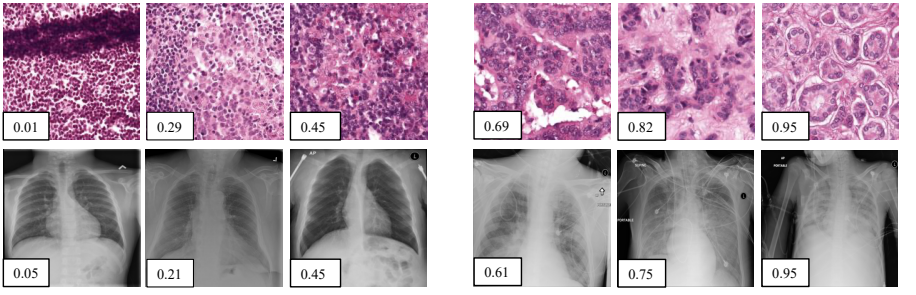
Data	γ_p	OC-SVM Hybrid [8]	IF Hybrid [8]	CAE [8]	Deep SVDD [8]	SSAD Hybrid [8]	SS-DGM [8]	Deep SAD [8]	TLSAD [1]	ESAD (ours)	Supervised Classifier
MNIST	.00	96.3 ± 2.5	90.5 ± 5.3	92.9 ± 5.7	92.8 ± 4.9	97.4 ± 2.0	92.2 ± 5.6	96.7 ± 2.4	96.9	**99.4 ± 0.3**	94.5 ± 4.6
	.01	95.6 ± 2.5	90.6 ± 5.0	91.3 ± 6.1	92.1 ± 5.1	95.2 ± 2.3	92.0 ± 6.0	95.5 ± 3.3	94.5	**99.2 ± 0.6**	91.5 ± 5.9
	.05	93.8 ± 3.9	89.7 ± 6.0	87.2 ± 7.1	89.4 ± 5.8	89.5 ± 3.9	91.0 ± 6.9	93.5 ± 4.1	94.0	**98.5 ± 1.0**	86.7 ± 7.4
	.10	91.4 ± 5.1	88.2 ± 6.5	83.7 ± 8.4	86.5 ± 6.8	86.0 ± 4.6	89.7 ± 7.5	91.2 ± 4.9	93.5	**97.8 ± 1.3**	83.6 ± 8.2
	.20	85.9 ± 7.6	85.3 ± 7.9	78.6 ± 10.3	81.5 ± 8.4	82.1 ± 5.4	87.4 ± 8.6	86.6 ± 6.6	88.6	**96.7 ± 2.0**	79.7 ± 9.4
F-MNIST	.00	91.2 ± 4.7	82.5 ± 8.1	90.2 ± 5.8	89.2 ± 6.2	90.5 ± 5.9	71.4 ± 12.7	90.5 ± 6.5	91.4	**95.6 ± 4.1**	76.8 ± 13.2
	.01	91.5 ± 4.6	84.9 ± 7.2	87.1 ± 7.3	86.3 ± 6.3	87.8 ± 6.1	71.2 ± 14.3	87.2 ± 7.1	92.3	**95.5 ± 4.1**	67.3 ± 8.1
	.05	90.7 ± 4.9	85.5 ± 7.2	81.6 ± 9.6	80.6 ± 7.1	82.7 ± 7.8	71.9 ± 14.3	81.5 ± 8.5	89.8	**94.5 ± 4.5**	59.8 ± 4.6
	.10	89.3 ± 6.2	85.5 ± 7.7	77.4 ± 11.1	76.2 ± 7.3	79.8 ± 9.0	72.5 ± 15.5	78.2 ± 9.1	90.2	**93.6 ± 4.7**	56.7 ± 4.1
	.20	88.1 ± 6.9	86.3 ± 7.4	72.5 ± 12.6	69.3 ± 6.3	74.3 ± 10.6	70.8 ± 16.0	74.8 ± 9.4	88.4	**92.5 ± 4.9**	53.9 ± 2.9
CIFAR-10	.00	63.8 ± 9.0	59.9 ± 6.7	56.2 ± 13.2	60.9 ± 9.4	73.3 ± 8.4	50.8 ± 4.7	77.9 ± 7.2	80.0	**86.9 ± 6.8**	63.5 ± 8.0
	.01	63.8 ± 9.3	59.9 ± 6.7	56.2 ± 13.1	60.5 ± 9.4	72.8 ± 8.1	51.1 ± 4.7	76.5 ± 7.2	76.4	**86.5 ± 6.9**	62.9 ± 7.3
	.05	62.6 ± 9.2	59.6 ± 6.4	55.7 ± 13.3	59.6 ± 9.8	71.0 ± 8.4	50.1 ± 2.9	74.0 ± 6.9	75.9	**84.3 ± 7.4**	62.2 ± 8.2
	.10	62.9 ± 8.2	59.1 ± 6.6	55.4 ± 13.3	58.6 ± 10.0	69.3 ± 8.5	50.5 ± 3.6	71.8 ± 7.0	72.6	**81.9 ± 7.7**	60.6 ± 8.3
	.20	61.9 ± 8.1	58.3 ± 6.2	54.6 ± 13.3	57.0 ± 10.6	67.9 ± 8.1	50.1 ± 1.7	68.5 ± 7.1	71.4	**79.8 ± 8.8**	58.5 ± 6.7

Table 5: Complete results of **experimental scenario (iii)**, where we increase the number of anomaly classes k_l included in the labeled training data. We report the avg. AUC in % with st. dev. computed over 100 experiments at various numbers k_l. Note that unsupervised methods [15, 16, 19, 23] cannot be applied to the semi-supervised setting when $k_l \neq 0$, while SS-DGM [12] and the supervised classifier are not compatible for the unsupervised setting when $k_l = 0$.

Data	k_l	OC-SVM Hybrid [8]	IF Hybrid [8]	CAE [8]	Deep SVDD [8]	SSAD Hybrid [8]	SS-DGM [8]	Deep SAD [8]	ESAD (ours)	Supervised Classifier
MNIST	0	91.4 ± 5.1	88.2 ± 6.5	83.7 ± 8.4	86.5 ± 6.8	91.4 ± 5.1		86.5 ± 6.8	**92.7 ± 3.8**	
	1					86.0 ± 4.6	89.7 ± 7.5	91.2 ± 4.9	**97.8 ± 1.3**	83.6 ± 8.2
	2					87.7 ± 3.8	92.8 ± 5.3	92.0 ± 3.6	**98.2 ± 0.9**	90.3 ± 4.6
	3					89.8 ± 3.3	94.9 ± 4.2	94.7 ± 2.8	**99.1 ± 0.6**	93.9 ± 2.8
	5					91.9 ± 3.0	96.7 ± 2.3	97.3 ± 1.8	**99.3 ± 0.5**	96.9 ± 1.7
F-MNIST	0	89.3 ± 6.2	85.5 ± 7.7	77.4 ± 11.1	76.2 ± 7.3	89.3 ± 6.2		76.2 ± 7.3	**91.2 ± 5.4**	
	1					79.8 ± 9.0	72.5 ± 15.5	78.2 ± 9.1	**93.6 ± 4.7**	56.7 ± 4.1
	2					80.1 ± 10.5	74.3 ± 15.4	80.5 ± 8.2	**94.7 ± 4.6**	62.3 ± 2.9
	3					83.8 ± 9.4	77.5 ± 14.7	83.9 ± 7.4	**95.8 ± 4.8**	67.3 ± 3.0
	5					86.8 ± 7.7	79.9 ± 13.8	87.3 ± 6.4	**96.7 ± 4.3**	75.3 ± 2.7
CIFAR-10	0	62.9 ± 8.2	59.1 ± 6.6	55.4 ± 13.3	86.6 ± 10.0	62.9 ± 8.2		58.6 ± 10.0	**73.5 ± 6.8**	
	1					69.3 ± 8.5	50.5 ± 3.6	71.8 ± 7.0	**81.9 ± 7.7**	60.6 ± 8.3
	2					72.3 ± 7.5	50.3 ± 2.4	75.2 ± 6.4	**83.8 ± 6.0**	61.0 ± 6.6
	3					73.3 ± 7.0	50.0 ± 0.7	77.5 ± 5.9	**84.9 ± 8.1**	62.7 ± 6.8
	5					74.2 ± 6.5	50.0 ± 1.0	80.4 ± 4.6	**86.7 ± 7.0**	60.9 ± 4.6

Besides the baselines considering in the main paper, we further consider several shallow unsupervised methods and deep unsupervised anomaly detection competitors as baselines. For the shallow unsupervised methods, OC-SVM [23] and Isolation Forest [15] are considered. For the deep unsupervised anomaly detection competitors, we consider CAE [16], Deep SVDD [19] AnoGAN [22], ALOCC [21], ADGAN [7], OCGAN [17], GANomaly [1], P-KDGAN [30] and DGEO [10]. OC-SVM here have unfair advantages by selecting their hyperparameters to maximize AUC on a subset (10%) of the test set to establish strong baselines.

Experimental Scenario (i). For the experimental scenario (i), where the effectiveness of adding labeled anomalies during training is investigated, i.e., increasing γ_l, has been shown in the main paper. In this part, we further report the results comparing with several unsupervised methods under the unsupervised setting in Table 3. We emphasize that our ESAD is not designed for the unsupervised setting. In these experiments, the semi-supervised terms are not working and make ESAD incomplete, since it remains only the unsupervised terms. Thus, these results are somewhat unfair for ESAD. Note that this paper still focus on the semi-supervised setting but not the unsupervised setting.



(a) Normal Samples (b) Anomalous Samples

Figure 1: Examples of normal (left) and anomalous (right) samples of H&E-stained lymph node of Camelyon16 challenge [2] (top) and chest X-rays of NIH dataset [28] (bottom). We show the predicted anomaly score by the proposed method. The higher the score, the more likely to be an anomaly. Best viewed in color.

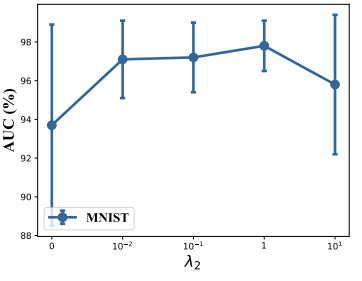


Figure 2: ESAD sensitivity analysis w.r.t. λ_1 on MNIST. We report avg. AUC with st. dev. over 90 experiments for various values of hyperparameter λ_2. Best viewed in color.

Experimental Scenario (ii). For the experimental scenario (ii), where the robustness is investigated in this scenario through adding polluted data. With an increasing pollution ratio γ_p, we pollute the unlabeled training set with anomalies drawn from all nine anomaly classes. We fix $\gamma_l = 0.05$ in this scenario. We report the average results over 90 experiments per pollution ratio γ_p. The corresponding results are shown in Table 4. Results show that ESAD is least affected by the pollution data and show the best robustness in all the polluted levels.

Experimental Scenario (iii). For the experimental scenario (iii), we increase the number of anomaly classes k_l included in the labeled part of the training set to increase the diversity of labeled anomalous data. As shown in Table 5, ESAD shows better performance in this scenario. For example, the AUC of ESAD on CIFAR-10 increases from 81.9% to 86.7% ($\gamma_l = 0.05$, $\gamma_p = 0.1$) when we change k_l from 1 to 5.

Examples Visualization. We illustrate the predictions of our model in Figure 1. Samples are randomly chosen from H&E-stained lymph node of Camelyon16 challenge [2] (top) and chest X-rays of NIH dataset [28] (bottom). These samples and their corresponding scores show that the higher the score, the more likely to be an anomaly.

Sensitivity Analysis on λ_2. We analyze the sensitivity of ESAD over the hyperparameters λ_2. Figure 2 shows the performance with different λ_2 using ESAD on MNIST. We set $\gamma_l = 0.05$, $\gamma_p = 0.1$, $k_l = 1$ in this experiment. Results show that without the assistant loss, i.e.,

$\lambda_2 = 0$, ESAD shows relatively low and unstable AUCs. ESAD shows best performance when $\lambda_2 = 1$. When λ_2 is too large, ESAD also shows unstable performance. This is because both two encoders will converge into the same constant function if the impact of the assistant loss is much greater than the other two mutual information and entropy based loss functions.

References

[1] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *International Joint Conference on Neural Networks (IJCNN)*, 2019.

[2] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 2017.

[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. 2010.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 2009.

[5] Thomas M Cover and Joy A Thomas. Elements of information theory 2nd edition (Wiley series in Telecommunications and Signal Processing), 2006.

[6] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[7] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Anomaly detection with generative adversarial networks. *arXiv preprint arXiv:1809.04758*, 2018.

[8] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *ICLR*, 2017.

[9] Zhe Feng, Jie Tang, Yishun Dou, and Gangshan Wu. Learning discriminative features for semi-supervised anomaly detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[10] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *NeurIPS*, 2018.

[11] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 2013.

[12] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NeurIPS*, 2014.

[13] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[14] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[15] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *ICDM*, 2008.

[16] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks (ICANN)*, 2011.

[17] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *CVPR*, 2019.

[18] Shebuti Rayana. Odds library. http://odds.cs.stonybrook.edu, 2016.

[19] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, 2018.

[20] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *ICLR*, 2020.

[21] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *CVPR*, 2018.

[22] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, 2017.

[23] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 2001.

[24] Yu-Xing Tang, You-Bao Tang, Mei Han, Jing Xiao, and Ronald M Summers. Deep adversarial one-class learning for normal and abnormal chest radiograph classification. In *Medical Imaging 2019: Computer-Aided Diagnosis*, 2019.

[25] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 2004.

[26] Nina Tuluptceva, Bart Bakker, Irina Fedulova, Heinrich Schulz, and Dmitry V Dylov. Anomaly detection with deep perceptual autoencoders. *arXiv preprint arXiv:2006.13265*, 2020.

[27] Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Melissa Schlitter, Irene Esposito, and Nassir Navab. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE transactions on medical imaging*, 2016.

[28] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, 2017.

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[30] Zhiwei Zhang, Shifeng Chen, and Lei Sun. P-kdgan: Progressive knowledge distillation with gans for one-class novelty detection. In *IJCAI*, 2020.