Few-Shot Fast-Adaptive Anomaly Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

The ability to detect anomaly has long been recognized as an inherent human ability, yet to date, practical AI solutions to mimic such capability have been lacking. This lack of progress can be attributed to several factors. To begin with, the distribution of "abnormalities" is intractable. Anything outside of a given normal population is by definition an anomaly. This explains why a large volume of work in this area has been dedicated to modeling the normal distribution of a given task followed by detecting deviations from it. This direction is however unsatisfying as it would require modeling the normal distribution of every task that comes along, which includes tedious data collection. In this paper, we report our work aiming to handle these issues. To deal with the intractability of abnormal distribution, we leverage Energy Based Model (EBM). EBMs learn to associate low energies to correct values and higher energies to incorrect values. At its core, the EBM employs Langevin Dynamics (LD) in generating these incorrect samples based on an iterative optimization procedure, alleviating the intractable problem of modeling the world of anomalies. Then, in order to avoid training an anomaly detector for every task, we utilize an adaptive sparse coding layer. Our intention is to design a plug and play feature that can be used to quickly update what is normal during inference time. Lastly, to avoid tedious data collection, this mentioned update of the sparse coding layer needs to be achievable with just a few shots. Here, we employ a meta learning scheme that simulates such a few shot setting during training. We support our findings with strong empirical evidence.

1 Introduction

Anomaly detection is an important area of study in the field of artificial intelligence. It has found utility in computer vision applications such as industrial inspection [4] and video surveillance [22, 50, 32], in the context of abuse prevention such as misinformation, fraud and network intrusion detection [49, 6, 29], and others such as system health monitoring and fault detection [3, 33]. In this paper, we propose an approach for detecting anomaly in images, where we have carefully designed steps to handle some of the bigger issues that have prevented the deployment of image anomaly detection in the real-world.

Image anomaly detection can generally be defined as the identification of abnormalities in a given image. An exact definition of abnormality in this case is elusive because abnormality can be derived from any unknown distribution outside of a normal population. Many studies have hence focused on modeling the normal population instead of learning irregularities, where the goal is to capture the shared concept among all of the normal data as one or several reference models. This process usually requires investing significant efforts in curating a large set of normal samples for *each task*, after which anomaly is detected as deviations from the reference model(s) [1, 47]. Recent work from [40] provides algorithms that utilize only a few normal samples to train models from scratch. However, the models still have to be provisioned for each new task, which requires considerable human efforts and expertise, and thus lack the fast deployment criterion that is often time critical for real-world

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

applications. In view of these challenges, our goals for this work are threefold. We are interested in designing an anomaly detection system that is capable of: (G1) modeling the normal population while at the same time has a principled approach towards modeling the abnormalities; (G2) quickly adapting to a new task at inference time; and (G3) requiring only a few normal shots to update itself to the new task at hand.

For (G1), we introduce the class of Energy Based Model (EBM), which is an important family of generative models [51, 12, 46]. EBMs have been shown to demonstrate superior capability on modeling data density and localizing anomaly [15]. For our purpose, the EBM we adopted learns to assign low energy to normal samples but high energy to abnormal samples. More importantly, the abnormal samples are generated with a procedure known as Langevin Dynamics (LD) [44], which, in its original form, starts with a noise image (see App. Fig F) and gradually samples from the distribution along the direction of lower energy. This lends itself gracefully to utilizing the generated intermediate samples as negative/abnormal. The LD procedure is then coupled with maximum likelihood loss [19] that aims to maximize the energy differences between the normal and abnormal samples.

To achieve (G2), we propose an adaptive sparse coding layer that is attached to the deep feature extractor in the EBM as Figure 1 shows. The extracted deep feature is forwarded to the sparse coding layer, where the dictionary is constructed with the features of a few normal samples of the given task. In essence, the input representation has been decomposed into a linear combination of normal features with the sparsity constraint imposed. The final energy score is measured by the distance between the original and the reconstructed features (after the sparse coding layer). Under this scheme, the dictionary for a particular task is not obtained by learning, but instead is constructed by the feature representations of a few normal samples during inference. As a result, this simple "plug-and-play" trick allows the model to be adapted to novel tasks promptly without re-training. Further, we expect that the dictionary, which is formed by normal features, will not be able to explain the abnormal samples well, causing relatively high reconstruction error that lends itself for subsequent detection. As a bonus, a backward pass of energy score minimization can be used for localizing abnormal regions. We show that using gradient to localize anomalies yields superior robustness.

Towards (G3), we utilize meta learning [42, 13] to simulate the scenario of being given a new task with a few normal shots to update the dictionary, followed by training the EBM. This is accomplished by episodic training, where in each episode the model is adapted to a held back task that is given a few normal samples. To accelerate the EBM training, we introduce "learning from inpainting", a simple yet effective strategy for synthesizing hard abnormal samples quicker by starting the LD procedure with a synthesized image that is simply a normal sample with a noise patch injected as opposed to a noise image that is traditionally what is used.

We show the proposed few-shot fast-adaptive anomaly detection and localization framework is able to efficiently adapt to a novel task (e.g., a new object category or scenes from a new camera) with a few normal samples without training on both industrial inspection and video surveillance. Compared with previous methods that adapts to new task through either from scratch training in few shots [40, 45] or few-shot with few steps of gradient descent [26], the proposed framework is the first that performs task adaptation with a single forward pass and without any gradient descent. Despite the fast adaptation, we provide both qualitative and quantitative results to demonstrate that our method outperforms other adaptive frameworks and is comparable to methods that rely on large amount of normal samples.

2 Backgrounds

We briefly introduce two key ingredients of the proposed method: EBMs and sparse coding.

Energy-based Model In EBMs, the goal is to learn an energy function $E_\theta(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ which parametrizes the data density $p_\theta(\mathbf{x})$ as:

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{\int_{\mathbf{x}} \exp(-E_\theta(\mathbf{x}))}, \tag{1}$$

where θ is the parameter of the energy function and $Z_\theta = \int_{\mathbf{x}} \exp(-E_\theta(\mathbf{x}))$ is the partition function. Approximating the true data distribution $p_{\text{data}}(\mathbf{x})$ is equivalent to minimizing the expected negative log-likelihood function over the data distribution, defined by the loss function:

$$\mathcal{L}_{\text{ML}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[-\log p_\theta(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[E_\theta(\mathbf{x}) + \log Z_\theta]. \tag{2}$$

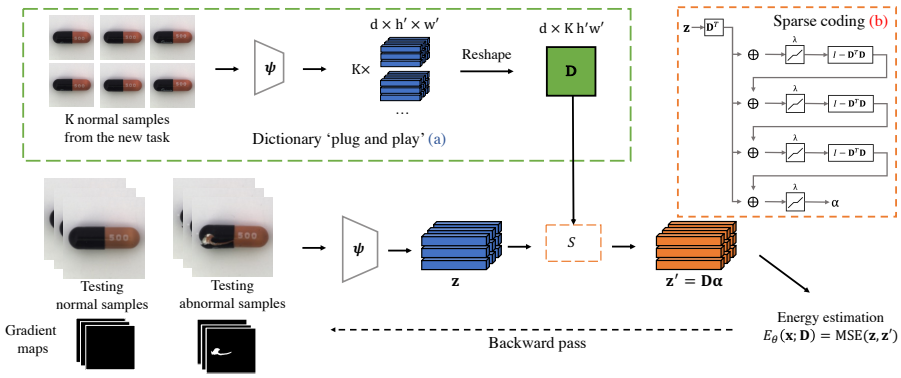

Figure 1: Overview of the inference stage on a new task. (a) Adapting the task-specific dictionary with K normal samples. (b) Sparse coding with three iterations as Eqn.6 shows. We also show a backward pass from the reconstruction error to localize the abnormal regions

As the computation of $\mathcal{L}_{\text{ML}}$ involves an intractable term Z_θ, the common practice is to represent the gradient of $\mathcal{L}_{\text{ML}}$ as,

$$\nabla_\theta \mathcal{L}_{\text{ML}} = \mathbb{E}_{\mathbf{x}^+\sim p_{\text{data}}(\mathbf{x})[\nabla_\theta E_\theta(\mathbf{x}^+)]} - \mathbb{E}_{\mathbf{x}^-\sim p_\theta(\mathbf{x})[\nabla_\theta E_\theta(\mathbf{x}^-)]}. \tag{3}$$

This objective decreases the energy of positive data samples $\mathbf{x}^+$ from the true distribution (normal samples in our use case) and increases the energy of negative samples $\mathbf{x}^-$ from the model p_θ (synthesized abnormal samples). In practice, the synthesized negative samples are achieved through Langevin dynamics [44], which a J-steps sampling along the direction of energy minimization is given by:

$$\tilde{\mathbf{x}}^j = \tilde{\mathbf{x}}^{j-1} - \frac{\beta}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{j-1}) + \omega^k, \quad \omega^k \sim \mathcal{N}(0, \beta\mathbf{I}), \quad j = \{1, \ldots, J\} \tag{4}$$

where β is the step size, and the initialization $\mathbf{x}^0$ is sampled from a predefined prior distribution. The synthesizing ability of EBMs enables generating abnormal samples to help in learning a more accurate data density, and is often touted as the one of the advantages of using an EBM.

Sparse coding. Approximating a signal $\mathbf{z} \in \mathbb{R}^d$ with the sparse linear combination over a dictionary $\mathbf{D} \in \mathbb{R}^{d\times k}$ can be expressed as:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}||\mathbf{z} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1, \tag{5}$$

where $\boldsymbol{\alpha}$ is the sparse coefficients, with its sparsity (l_1 norm) and λ is the weight of the sparsity constraint. $\mathbf{D}\boldsymbol{\alpha}$ is a sparse approximation to the original signal $\mathbf{z}$. In practice, finding the dictionary atoms and the sparse coefficients is usually formulated as an optimization problem.

In this paper, we adopt Iterative Soft Thresholding Continuation (ISTC) [20] to convert this optimization problem into linear operations with a non-linear shrinkage function, which allows sparse coding to be seamlessly integrated into the deep neural networks. To compute a sparse coefficient $\boldsymbol{\alpha}$, ISTC performs iterations of gradient steps on reconstruction $||\mathbf{z} - \mathbf{D}\boldsymbol{\alpha}||^2$ and a proximal projection step to increase coefficient sparsity.

Formally, initializing the coefficients at the first step $\boldsymbol{\alpha}_0$ with all zeros, each step of ISTC refines the sparse code with descending values of λ from λ_{max} to $\lambda_\star$: each step of ISTC is expressed as:

$$\boldsymbol{\alpha}_{n+1} = \sigma(\boldsymbol{\alpha}_n + \mathbf{D}^\top(\mathbf{z} - \mathbf{D}\boldsymbol{\alpha}_n), \lambda_n), \quad \text{with} \quad \lambda_n = \lambda_{\text{max}}\frac{\lambda_{\text{max}}}{\lambda_\star}^{-n/N}, \tag{6}$$

where $\sigma(\cdot,\cdot)$ here is a shrinkage function that truncates small values (lower than λ) of the coefficients to 0 to enforce sparsity, and can be easily implemented by a customized ReLU activation function:

$$\sigma(\mathbf{z}, \lambda) = \text{sgn}(\mathbf{z})(\max(|\mathbf{z}| - \lambda, 0)) = \text{sgn}(\mathbf{z})\text{ReLU}(|\mathbf{z}| - \lambda). \tag{7}$$

3 Proposed Method

In this section, we describe the proposed fast adaptive anomaly detection framework in details. In Section 3.1, we introduce the adaptive EBM which consists of a deep feature extractor followed

by an adaptive sparse coding layer. From there, we further show that utilizing larger receptive field in the sparse coding could improve training robustness (Section 3.1.1), and applying smoothed shrinkage functions could help speed up convergence (Section 3.1.2). In Section 3.2, we describe the episodic training regime on various anomaly detection tasks that mimics few-shot adaptation in the meta-testing stage while learning common knowledge across tasks. Instead of synthesizing negative samples (anomaly) directly from noise, we introduce a simple but effective "learning from inpainting" operation to accelerate the training in Section 3.3. Finally, we summarize training steps of the proposed method in Algorithm 1.

3.1 Adaptive Energy-based Model

An EBM is a form of generative model and it is widely used for modeling data density and sampling. While there has been recent work [15] applying EBM to anomaly detection, it still requires re-training for each new task. To efficiently adapt the EBM to novel tasks, we introduce an adaptive sparse coding layer which is conditioned on the dictionary constructed by the features of normal samples. Specifically, as illustrated in Fig 1, given an input image, $\mathbf{x} \in \mathbb{R}^{3\times h\times w}$, we first obtain the corresponding feature $\mathbf{z} \in \mathbb{R}^{d\times h'\times w'}$ from the deep feature extractor $\mathbf{\Psi}$ with parameters θ, so that $\mathbf{z} = \mathbf{\Psi}(\mathbf{x};\theta)$. All feature vectors along spatial axes of $\mathbf{z}$ are then sparsely decomposed through the sparse coding layer over a task-specific dictionary $\mathbf{D} \in \mathbb{R}^{d\times Kh'w'}$, which contains the features of K normal samples of the current task as shown in the Fig 1(a). Each feature vector of the normal sample features is then directly used as an atom in the task dictionary. The decomposed coefficients are $\alpha = \boldsymbol{\mathcal{S}}(\mathbf{z};\mathbf{D})$, where $\alpha \in \mathbb{R}^{Kh'w'\times h'\times w'}$ and $\boldsymbol{\mathcal{S}}$ denotes the iterative sparse decomposition process of (6). By multiplying the coefficient α with the dictionary $\mathbf{D}$, we obtain the reconstructed features $\mathbf{z}' = \mathbf{D}\alpha$. The sparsity regularization to α is important, as it encourages input features to be reconstructed by simple combinations of dictionary atoms (normal features), so that it would be difficult for features of abnormal samples to be well-approximated, therefore producing higher reconstruction errors that make it conducive for detecting anomalies. From here, the final energy score is formulated as the mean squared error (MSE) between the original and the reconstructed features:

$$E_\theta(\mathbf{x};\mathbf{D}) = \text{MSE}(\mathbf{z},\mathbf{z}') = ||\mathbf{\Psi}(\mathbf{x};\theta) - \mathbf{D}\boldsymbol{\mathcal{S}}\big(\mathbf{\Psi}(\mathbf{x};\theta);\mathbf{D}\big)||^2. \tag{8}$$

In effect, Eqn. 8 depicts a conditional EBM, which is conditioned on the task-specific $\mathbf{D}$ formed by normal features. With the energy score, we can obtain pixel-wise **anomaly localization maps** through $\nabla_{\mathbf{x}} - E_\theta(\mathbf{x};\mathbf{D})$, i.e., the gradients of pixels along the direction of minimization. High gradient magnitudes indicate regions that cannot be well explained by the dictionary $\mathbf{D}$. Modifications to these regions can potentially remove the anomaly and reduce the energy as in Eqn. 4. In Section 4.1 and Appendix Section B.5, we show that using the gradient (as a natural ingredient of EBMs with LD) is more robust compared with auto-encoder and reconstruction based methods to generalize well to unseen tasks (Appendix Figure C). In the following sections, we will discuss how to make the training of this adaptive structure more robust.

3.1.1 Sparse Coding with Receptive Field.

As discussed in Section 3.1, the input feature $\mathbf{z}$ is represented as $h' \times w'$ of d-dim feature vectors and they are treated independently while passing through the sparse coding layer. The region of the input image that affects one feature vector is determined by the receptive field of the feature extractor. The trade-off is that a small receptive field may not capture enough contextual information, while applying a large receptive field would make feature maps spatially coarse and make it hard to spot small anomaly regions. To solve this dilemma, instead of carefully tuning the receptive field of each layer of the feature extractor, we introduce a simple yet effective technique of applying the receptive field on the sparse coding layer. Specifically, as illustrated in Appendix Fig A, rather than performing sparse coding to each individual d-dim feature vectors, we apply it on $d \times l \times l$ volumes centered around each feature vector, where l is the receptive field. This is equivalent to applying a $l \times l$, sliding window on spatial axes of the feature map and can be easily implemented by *image to column* (*Im2Col*) operation. Then we flatten the feature volumes into dl^2-dim vectors and adjust the shape of the dictionary accordingly. In this way, we are able to capture contextual information without needing to carefully tune the architecture of the feature extractor and we show in the later experiments that this technique improves the robustness of the network on different types of objects.

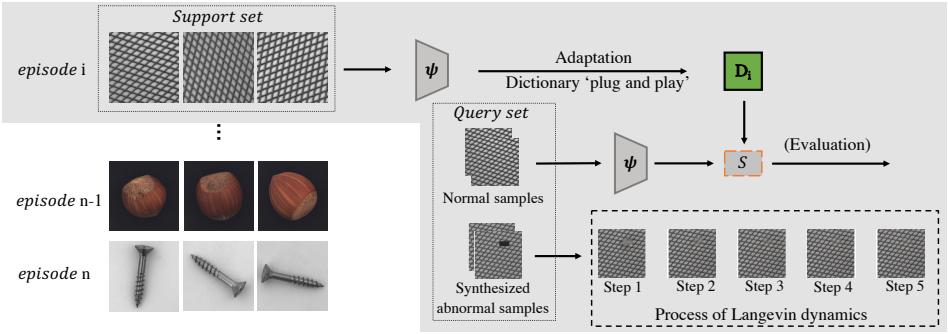

Figure 2: Illustration of episodic training and (a) "learning by inpainting".

3.1.2 Shrinkage Function

The effectiveness of training the EBM for localizing anomaly regions heavily depends on the gradient propagation from later to earlier layers. It is shown in [11] that smooth activation functions like Swish [35] could be beneficial here. Notably, the gradients of the dictionary $\mathbf{D}$ are determined by the sparse coding coefficients α as shown in Eqn. 6. However, the sparsity constraint of α would turn off the gradient computation of many elements in $\mathbf{D}$ and this could be detrimental during the early stage of the training. To alleviate the sparse gradient issue, we replace the RELU-like shrinkage function in Eqn. 7 with its smoothed counterparts by introducing the Sigmoid based shrinkage functions (SigShrink). The SigShrink function is originally proposed for non-parametric signal estimation in [2], and can be defined as:

$$\sigma_\tau(\mathbf{z}, \lambda) = \frac{\mathbf{z}}{1 + \exp(-\tau(|\mathbf{z}| - \lambda))}, \tag{9}$$

where τ is the hyperparameter of smoothness. We present visualizations of the hard shrinkage function Eqn. 7 and SigShrink with different values of τ in Fig B. Comparing to the hard shrinkage function which truncates small values into zeros, the SigShrink with a large τ can sharply force small values to near-zeros. Therefore, the SigShrink will guarantee non-zero gradients everywhere.

3.2 Episodic Training

To train the proposed adaptive EBM, we perform episodic training that is widely adopted by meta-learning few-shot learners [13, 41]. Following the terminology of few-shot learning, in each training episode, the model is adapted and tested with a task sampled from the underlying task distribution. Specifically, the model is adapted to a support set of the given task, then a query set with ground truth labels is applied to evaluate the adaptation, which is used to update the model parameters. As shown in Fig 2, the support set of the i-th episode task contains a small number of K normal samples $\{\mathbf{s}_k^i\}_{k=1}^K$. The features $\mathbf{z}_k^i = \mathbf{\Psi}(\mathbf{x}_k^i; \theta)$ of these normal samples are plugged into the dictionary $\mathbf{D}^i \in \mathbb{R}^{d \times Kh'w'}$ corresponding to the i-th task to adapt the dictionary. After that, the adapted model is measured by a query set consisting of M normal samples $\{\mathbf{q}_m^i\}_{m=1}^M$ and M abnormal samples $\{\hat{\mathbf{q}}_m^i\}_{m=1}^M$. Note that there is no actual abnormal samples given during training, instead, they are iteratively sampled from the EBM and the sampling will be discussed in details in Section 3.3. Recall that the training of EBM with contrastive divergence as in Eqn. 3 requires the estimation of energy scores of both positive samples from the true data distribution and negative samples from the modeled distribution. The positive energy can be estimated empirically with normal query set samples. The negative energy can be estimated by performing the MCMC-based sampling technique [31, 44], typically Langevin Dynamics as described in Eqn. 4. Denoting the output of Langevin dynamics (sampled abnormal samples) initialized with $\hat{\mathbf{q}}_m^i$ as $\mathbf{LD}(\hat{\mathbf{q}}_m^i)$, we have the empirical estimation of the contrastive divergence of the i-th episode as:

$$\mathcal{L}_{\text{cd}} = \frac{1}{m} \sum_{m=1}^{M} \left[E_\theta(\mathbf{q}_m^i; \mathbf{D}^i) - E_\theta(\mathbf{LD}(\hat{\mathbf{q}}_m^i); \mathbf{D}^i) \right]. \tag{10}$$

With the energy score equivalent to the feature reconstruction error in Eqn. 8, minimizing $\mathcal{L}_{\text{cd}}$ encourages normal features to be well-reconstructed by a sparse linear combination of dictionary atoms while the features from abnormal samples tend to produce relatively higher reconstruction errors so that they can be easily spotted.

Algorithm 1 Training steps of few-shot adaptive anomaly detection.

Given: A feature extractor $\mathbf{\Psi}$ with parameter θ; a training dataset of multiple tasks with positive (normal) samples only.
Given: Number of shots K; number of query samples Q; step size β of Langevin dynamics; total training episodes I; and learning rate ϵ.
Initialize the feature extractor $\mathbf{\Psi}$.
for Episode $i = 1 : I$ **do**
  Sample the i-th task from the dataset, and randomly pick $K + M$ samples to form the support set $\{\mathbf{s}_k^i\}_{k=1}^K$ and the query set $\{\mathbf{q}_m^i\}_{m=1}^M$.
  Generate corrupted query samples $\{\hat{\mathbf{q}}_m^i\}_{m=1}^M$ by placing random patches to $\{\mathbf{q}_m^i\}_{m=1}^M$.
  Extract the support and query sample features with $\mathbf{\Psi}$ and update the adaptive sparse coding layer with i-th task dictionary $\mathbf{D}^i$, which is constructed by support sample features. The energy function of the i-th task is now parametrized by $E_\theta(\cdot, \mathbf{D}^i)$.
  Obtain synthesized negative samples $\{\mathbf{LD}(\hat{\mathbf{q}}_m^i)\}_{m=1}^M$ with the updated energy function using Langevin dynamic in (4).
  Obtain the final loss $\mathcal{L}$ with $\mathcal{L}_{\text{cd}}$ (10) and $\mathcal{L}_{\text{rec}}$ (11).
  Update parameters $\theta \leftarrow \theta - \epsilon \nabla_\theta \mathcal{L}$.
end for
Return $\mathbf{\Psi}$ with parameter θ.

3.3 Synthesizing Negative Samples

Typical EBM training with contrastive divergence conducts negative sampling from the modeled density using techniques such as Langevin Dynamics, which applies gradient descent to a noise initialization (App. Fig F) with small step size and large number of steps [12]. Such negative sampling steps can be costly and we argue that it is unnecessary in our case. Instead, we introduce a new strategy of "learning by inpainting". Starting from a positive query sample $\mathbf{q}_m^i$, we synthesize the corresponding negative sample $\hat{\mathbf{q}}_m^i$ by randomly placing a small uniform noise patch on the image. The Langevin Dynamics procedure is then initialized with the resulting image instead of a noise image. As the Langevin Dynamics proceeds, synthesized abnormal samples $\mathbf{LD}(\hat{\mathbf{q}}_m^i)$ are inpainted along the direction of "normal", $\mathbf{q}_m^i$, and we introduce the following reconstruction loss:

$$\mathcal{L}_{\text{rec}} = \frac{1}{m} \sum_{m=1}^{M} \text{MSE}(\mathbf{LD}(\hat{\mathbf{q}}_m^i), \mathbf{q}_m^i). \tag{11}$$

We show in Fig 2(a) that, starting from a synthesized abnormal sample, only 5 steps of Langevin dynamic would be sufficient to make it visually close to the corresponding normal sample during training, serving as "hard negatives" that further facilitates the learning. The final loss of the episodic training is simply:

$$\mathcal{L} = \eta_0 \mathcal{L}_{\text{rec}} + \eta_1 \mathcal{L}_{\text{cd}}, \tag{12}$$

where η_0 and η_1 are balance two loss terms.

We summarize the overall training of the proposed few-shot adaptive anomaly detection method in Alg. 1, and testing steps at deployment in Appendix Alg. 2.

4 Experiments

In this section, we conduct evaluation on the industrial inspection task with the MVTec-AD dataset [4] (Section 4.1). Even though our proposed framework is image-based, we further demonstrate it's efficacy on the video anomaly detection task in Section 4.2. In Section 4.3, we show ablations and insights relating to the adaptive sparse coding components. We show additional ablations including the superiority of using gradient of EBMs over pixel-wise reconstruction to localize anomalies in App. B and we provide implementation details in App. A.

4.1 Industrial Inspection

The goal of this anomaly detection task is to predict whether a manufactured component contains any defects. The MVTec-AD dataset includes 15 categories of object. To demonstrate the fast adaptation capability of the proposed method, we adopt a *leave-one-out* training strategy. Specifically, samples

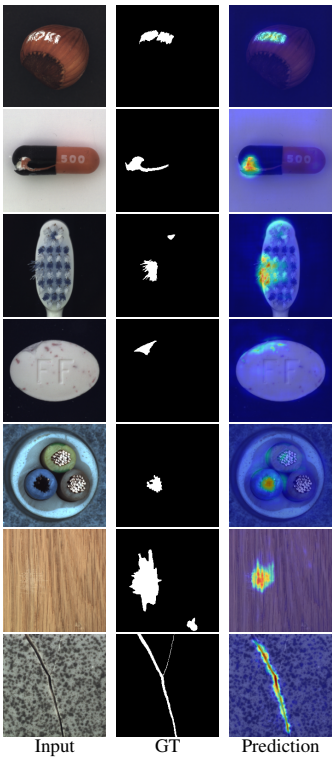

Figure 3: Visualizations of localized anomaly by our method.

Category	AE (SSIM)	AE (MSE)	AnoGAN	VE-VAE	MAML-AE	HTDGM	Ours
Carpet	0.69 0.87	0.38 0.59	0.34 0.54	0.1 0.78	0.20 0.68	0.21 0.78	0.28 0.83
Grid	0.88 0.94	0.83 0.90	0.04 0.58	0.02 0.73	0.01 0.53	0.07 0.74	0.12 0.81
Leather	0.71 0.78	0.67 0.75	0.34 0.64	0.74 0.87	0.12 0.77	0.39 0.90	0.42 0.98
Tile	0.04 0.59	0.23 0.51	0.08 0.50	0.14 0.93	0.14 0.52	0.12 0.66	0.28 0.81
Wood	0.36 0.73	0.29 0.73	0.14 0.62	0.47 0.91	0.11 0.68	0.22 0.79	0.23 0.78
Bottle	0.15 0.93	0.22 0.86	0.05 0.86	0.07 0.78	0.02 0.56	0.25 0.88	0.23 0.82
Cable	0.01 0.82	0.05 0.86	0.01 0.78	0.18 0.90	0.04 0.74	0.16 0.86	0.24 0.87
Capsule	0.09 0.94	0.11 0.88	0.04 0.84	0.11 0.74	0.03 0.68	0.04 0.88	0.12 0.90
Hazelnut	0.00 0.97	0.41 0.95	0.02 0.87	0.44 0.98	0.11 0.72	0.38 0.95	0.40 0.94
Metal nut	0.01 0.89	0.26 0.86	0.00 0.76	0.49 0.94	0.10 0.78	0.40 0.84	0.39 0.87
Pill	0.07 0.91	0.25 0.85	0.17 0.87	0.18 0.83	0.10 0.62	0.14 0.71	0.22 0.88
Screw	0.03 0.96	0.34 0.96	0.01 0.80	0.17 0.97	0.02 0.55	0.17 0.81	0.17 0.83
Toothbrush	0.08 0.92	0.51 0.93	0.07 0.90	0.14 0.94	0.06 0.80	0.14 0.84	0.23 0.82
Transistor	0.01 0.90	0.22 0.86	0.08 0.80	0.30 0.93	0.02 0.76	0.10 0.82	0.26 0.85
zipper	0.10 0.88	0.13 0.77	0.01 0.78	0.06 0.78	0.04 0.68	0.16 0.89	0.12 0.82

Table 1: Numerical evaluation of anomaly localization on MVTec-AD. We report both mIoU (top rows) and AUC-ROC (bottom rows) values. Col 2-5 are upper-bound methods trained with massive normal samples.

of each target category are reserved for testing only, and the episodic training is performed on the remaining categories. During the training stage, the model will not see any samples from the target category. During testing, we first adapt the model to the target category with *10 randomly selected normal samples*, then measure the performance with the entire testing set. We run the test 5 times, each time the model is adapted to random sets of 10 normal samples from the target category. The final result is the average of the 5 runs.

In Table 1, we first show performance of "upper-bound" methods, which train each category from scratch with massive normal samples. Specifically, [5, 4] train auto-encoders (AE) with normal samples and measure the reconstruction errors during the inference; AnoGAN [38] adopts a generative adversarial network (GAN) to learn the manifold of normal; VE-VAE [23] presents a visually explainable variational auto-encoder through gradient-based attention. For apple-to-apple comparison, we create a strong baseline by applying model-agnostic meta-learning [13] on an AE (denoted as MAML-AE, more details in App. Sec. A.3). Hierarchical transformation-discriminating generative models (HTDGM) [40] trains a GAN-based anomaly detector in few shots. Although HTDGM allows adapting to a new task with few normal samples, the model still needs to be trained from scratch, which takes 4000 iterations in the 10 shot setting based on their official implementation. To the best of our knowledge, the proposed method is the first to allow adapting to new tasks with **a single forward pass without any training**, meanwhile demonstrate strong performance on challenging anomaly detection and localization tasks. Therefore, the learned parameters are shared across tasks which greatly accelerate the model deployment. All results from our methods are obtained *without any data augmentation*. The proposed method outperforms MAML-AE by a large margin, and performs better than HTDGM [40] in majority of tasks. Our results are even competitive with the "upper-bounds" in some categories. We show the localized anomaly regions from our method in Fig 3. Additional visualizations are in the App. Fig D.

4.2 Video Surveillance

In video anomaly detection, a common goal is to detect abnormal events captured by surveillance cameras (e.g., a motorcycle on the sidewalk). A model trained on videos from one camera might not generalize well on other cameras due to different locations / mounting heights / lightning conditions,

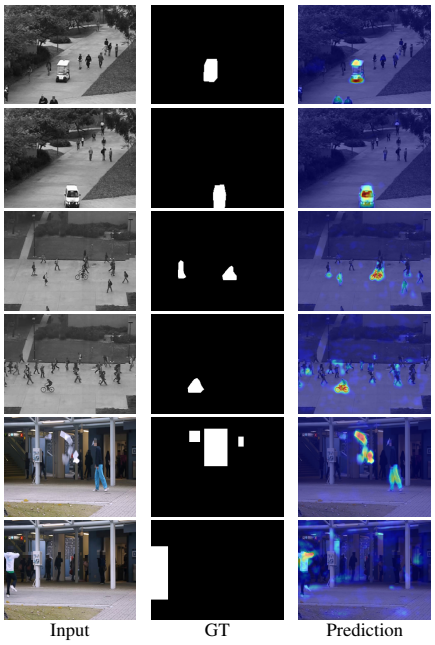

Figure 4: Visualizations of anomaly localization with video anomaly detection.

Target datasets	Methods	1-shot	5-shot	10-shot
UCSD Ped 1	r-GAN Pre-train	73.10	73.10	73.10
	r-GAN Fine-tune	76.99	77.85	78.23
	r-GAN MAML	80.60	81.42	82.38
	MAML-AE	64.12	66.88	67.34
	Ours	77.42	78.12	78.65
UCSD Ped 2	r-GAN Pre-train	81.95	81.95	81.95
	r-GAN Fine-tune	85.64	89.66	91.11
	r-GAN MAML	91.19	91.80	92.80
	MAML-AE	78.24	82.04	83.30
	Ours	91.22	92.00	92.45
CUHK Avenue	r-GAN Pre-train	71.43	71.43	71.43
	r-GAN Fine-tune	75.43	76.52	77.77
	r-GAN MAML	76.58	77.10	78.79
	MAML-AE	68.72	69.67	70.01
	Ours	80.68	83.41	84.46
Sh-Tech	r-GAN Pre-train	70.11	70.11	70.11
	r-GAN Fine-tune	71.61	70.47	71.59
	r-GAN MAML	74.51	75.28	77.36
	MAML-AE	66.62	67.12	68.04
	Ours	75.32	79.64	81.28

Table 2: Frame-level AUC-ROC for the video anomaly detection tasks.

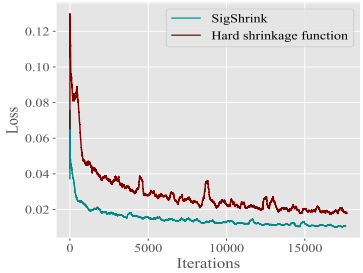

Figure 5: Loss curves with smooth (SigShrink) and non-smooth (hard-shrink RELU-like) shrinkage functions.

Category	Leather		Grid		Hazelnut		Cable		Pill	
$l = 1$	0.41	0.98	0.11	0.80	0.36	0.91	0.21	0.85	0.10	0.85
$l = 3$	0.42	0.98	0.12	0.81	0.40	0.94	0.24	0.87	0.22	0.88

Table 3: Comparison of different sparse coding receptive fields. We report both mIoU (left) and AUC-ROC (right) values.

Category	Leather			Hazelnut			Cable		
Ours	0.42	0.98	1.6e-4	0.40	0.94	2.4e-4	0.24	0.87	2.0e-4
No sparsity	0.32	0.90	0.9e-4	0.24	0.80	1.7e-4	0.12	0.68	1.5e-4

Table 4: Performance w/ and w/o sparsity constraint. From left to right: mIoU; AUC-ROC; the difference of averaged reconstruction errors between abnormal/normal samples.

and it is not feasible to train one model for every new camera in practice. The ability to quickly adapt to new scenes is a significant contribution to the task of video surveillance. We are only aware of the work in [26] (r-GAN) that has such adaptation capability. Specifically, the model adapts to a new scene using gradient descent with several beginning frames of a query video, after which a GAN is applied to generate future frames. Anomaly is then detected via the discrepancy between predicted future frames and the original frames. Note that the MAML-AE baseline we conducted in Section 4.1 can be seen as an ablation of r-GAN on the single-frame without temporal information.

We follow the same evaluation regime as r-GAN by training with normal samples in all 13 scenes from SH-Tech [22] and testing on UCSD Pedestrian 1, UCSD Pedestrian 2 [28], and CUHK Avenue [24]. Note that since our method is image-based, it predicts the video frames independently without leveraging any temporal information as in r-GAN. In each episode, we adapt our model with a support set containing a few normal frames randomly sampled from the target scenes. In Table 2, we compare our method against r-GAN pre-trained on SH-Tech only (r-GAN Pre-train), fine-tuned on target datasets (r-GAN Fine-tune), and with one step gradient descent with meta-learning (r-GAN MAML). We also show the performance of MAML-AE as a baseline for image-based meta-learning method. In the last section of Table 2, we present intra-dataset results as well by training with 6 scenes of SH-Tech and testing on remaining 7. We follow common evaluation protocol and measure the frame-level AUC-ROC. Without leveraging temporal information and re-training (gradient descent), our method achieves comparable results to r-GAN MAML and outperforms image-based meta-learning method

by a large margin. In App B.4, we show that incorporating simple temporal information can further improve the performance.

4.3 Ablation Studies

Sparse coding receptive fields. To evaluate the effectiveness of using large receptive fields in the sparse coding layer, we conduct additional experiments on the MVTec-AD dataset, and select 5 representative categories with different levels of difficulties to present the comparisons with $l = 1$ and $l = 3$ (Sec. 3.1.1) in Table 3. Sparse coding with large receptive field clearly benefits more complex structural objects (hazelnut, cable, and capsule), while the improvements are limited for the texture objects (leather and grid), where contextual regularization is intuitively less important.

Shrinkage functions. To show the benefits of smooth shrinkage function, we plot the loss curves of models trained with smooth SigShrink (Eqn. 9) and non-smooth RELU-like shrinkage (Eqn. 7) functions in Fig 5. The model with smooth shrinkage function converges notably faster in the early training stage and achieves lower loss.

Sparsity constraint. As discussed in Section 3.1, we impose sparsity constraint to the feature decomposition in the adaptive sparse coding layer, in order to prevent abnormal features from being well-approximated by the linear combinations of normal features, so that the reconstruction errors are effective for detecting anomaly. To validate this, we conduct experiments by removing the shrinkage function σ in the sparse coding stage (Eqn. 6). We show comparison in Table 4 with mIoU, AUC-ROC, and the difference of averaged reconstruction errors between abnormal and normal samples. Without sparsity, the performance drops dramatically, and reconstruction errors of normal and abnormal samples become closer.

5 Related Work

Anomaly detection with sparse coding. Early efforts on adopting sparse coding in anomaly detection are based on optimization (with L1 penalty) [24, 50]. Recent advances on iterative sparse thresholding algorithms [9, 20] allow seamless integration of online sparse coding with deep neural networks, and [27] formulates the sparse coding as stack RNNs for video anomaly detection.

Anomaly detection with generative models. Generative models are widely utilized in anomaly detection due to the capability in modeling the density of desired data distribution. Early efforts on variational autoencoders (VAE) based methods [1, 47] are arguably having hard time calibrating uncertainties in novel samples [30], accurately localizing abnormal regions through reconstruction errors [10]. Recent efforts have explored variant generative architectures like energy-based models (EBM) [15], GANs [40], and combining VAE with EBM [10]. Various methods also exploit intra-image structures [8, 5], cross-frame consistency [25], and motion-appearance consistency in videos [32] while detecting anomaly.

Few-shot learning. Few-shot learning is extensively explored in classification tasks. Proposed methods are based on optimization [13, 37, 14, 48, 36], learning metric [41, 43] and parameter prediction [17, 34, 16]. These technologies are further applied in other tasks like image generation [7, 21] and out-of-distribution detection [39].

6 Conclusion

In this paper, we introduced few-shot fast-adaptive anomaly detection. We formulated our model as an energy-based model with an adaptive sparse coding layer, of which the dictionary is directly formed by normal features of a target task. We adopted episodic meta-learning to learn common knowledge across tasks, which enables few shots adaptation. We further introduced smooth shrinkage functions, sparse coding with large receptive fields, and learning by inpainting to improve and accelerate the training. Notably, when evaluating our method's performance on industrial inspection and video anomaly detection, our method is comparable and even boasts better performance than methods trained with a large amount of normal samples. Through this work, we hope to have made a significant contribution to the important problem of anomaly detection by shedding light on our findings that anomaly detection can indeed be generalized to new tasks with a few normal samples only.

Social Impact and Ethics. As a general framework for few-shot anomaly detection, the proposed method does not suffer from particular ethical concerns or negative social impacts. All datasets used are public, and we have blurred all human faces in the qualitative visualizations.

References

[1] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[2] Abdourrahmane M Atto, Dominique Pastor, and Gregoire Mercier. Smooth sigmoid wavelet shrinkage for non-parametric estimation. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3265–3268. IEEE, 2008.

[3] Yuequan Bao, Zhiyi Tang, Hui Li, and Yufeng Zhang. Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, 18(2):401–421, 2019.

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019.

[5] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

[6] Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical science*, 17(3):235–255, 2002.

[7] Louis Clouâtre and Marc Demers. Figr: Few-shot image generation with reptile. *arXiv preprint arXiv:1901.02199*, 2019.

[8] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020.

[9] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

[10] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative energy-based projection on a normal data manifold for anomaly localization. *ICLR*, 2020.

[11] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *International Conference on Machine Learning*, 2021.

[12] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *NeurIPS*, 2019.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[14] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

[15] Ergin Utku Genc, Nilesh Ahuja, Ibrahima J Ndiour, and Omesh Tickoo. Energy-based anomaly detection and localization. *arXiv preprint arXiv:2105.03270*, 2021.

[16] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2019.

[17] Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E Turner. Meta-learning probabilistic inference for prediction. *arXiv preprint arXiv:1805.09921*, 2018.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[20] Yuling Jiao, Bangti Jin, and Xiliang Lu. Iterative soft/hard thresholding with homotopy continuation for sparse recovery. *IEEE Signal Processing Letters*, 24(6):784–788, 2017.

[21] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10551–10560, 2019.

[22] W. Liu, D. Lian W. Luo, and S. Gao. Future frame prediction for anomaly detection – a new baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[23] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J Radke, and Octavia Camps. Towards visually explaining variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8642–8651, 2020.

[24] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013.

[25] Yiwei Lu, K Mahesh Kumar, Seyed shahabeddin Nabavi, and Yang Wang. Future frame prediction using convolutional vrnn for anomaly detection. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE, 2019.

[26] Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*, pages 125–141. Springer, 2020.

[27] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 341–349, 2017.

[28] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981. IEEE, 2010.

[29] Biswanath Mukherjee, L Todd Heberlein, and Karl N Levitt. Network intrusion detection. *IEEE network*, 8(3):26–41, 1994.

[30] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[31] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[32] Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1273–1283, 2019.

[33] Afrooz Purarjomandlangrudi, Amir Hossein Ghapanchi, and Mohammad Esmalifalak. A data mining approach for fault diagnosis: An application of anomaly detection algorithm. *Measurement*, 55:343–352, 2014.

[34] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018.

[35] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[36] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations*, 2016.

[37] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *International Conference on Learning Representations*, 2019.

[38] 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*, pages 146–157. Springer, 2017.

[39] Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. *arXiv preprint arXiv:2103.12051*, 2021.

[40] Shelly Sheynin, Sagie Benaim, and Lior Wolf. A hierarchical transformation-discriminating generative model for few shot anomaly detection. *arXiv preprint arXiv:2104.14535*, 2021.

[41] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 2017.

[42] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

[43] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 2016.

[44] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[45] Jhih-Ciang Wu, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Learning unsupervised metaformer for anomaly detection. In *International Conference on Computer Vision*, pages 4369–4378, 2021.

[46] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644. PMLR, 2016.

[47] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196, 2018.

[48] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7343–7353, 2018.

[49] Qiang Zhang, Aldo Lipani, Shangsong Liang, and Emine Yilmaz. Reply-aided detection of misinformation via bayesian deep learning. In *The world wide web conference*, pages 2333–2343, 2019.

[50] Bin Zhao, Li Fei-Fei, and Eric P Xing. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR 2011*, pages 3313–3320. IEEE, 2011.

[51] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[52] Yang Zhao, Jianwen Xie, and Ping Li. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *International Conference on Learning Representations*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Code will be released upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Implementation Details

We adopt a residual network (ResNet) [18] based feature extractor, with ELU as the activation function. Following [11], we adopt Group Normalization for better stability of the network. The detailed network construction is shown in Table A.

ResNet-10	
Output size	**Layers**
$224\times224\times3$	Input images
$56\times56\times32$	*Conv*(7×7, stride=2), *GroupNorm*, *ELU*, *AveragePool*(3×3, stride=2)
$56\times56\times32$	[*Conv* (3×3), *GroupNorm*, *ELU*] $\times$ 2, *Skip connect Conv* (1×1, stride=2), *GroupNorm*
$28\times28\times64$	*Conv* (3×3, stride=2), *GroupNorm*, *ELU*, *Conv* (3×3), *BatchNorm*, *ELU* *Skip connect Conv* (1×1, stride=2), *GroupNorm*
$14\times14\times128$	*Conv* (3×3, stride=2), *GroupNorm*, *ELU*, *Conv* (3×3), *GroupNorm*, *ELU* *Skip connect Conv* (1×1, stride=2), *GroupNorm*
$7\times7\times256$	*Conv* (3×3, stride=2), *GroupNorm*, *ELU*, *Conv* (3×3), *GroupNorm*, *ELU* *Skip connect Conv* (1×1, stride=2), *GroupNorm*
$7\times7\times256$	*Conv* (1×1, stride=2), *GroupNorm*, *Tanh*

Table A: The architectures of feature extractor.

A.1 Industrial Inspection

We adopt the "leave-one-out" training strategy for obtaining the results on each of the categories of MVTec-AD. All experiments are performed with the same settings and hyperparameters. We resize all images to 128×128, and do not perform any data augmentation. We adopt a simple reduced-sized ResNet as the feature extractor as shown in Table A. Following [11], we adopt group normalization (denoted as GroupNorm) instead of batch normalization, and use Exponential Linear Unit (ELU) as the activation function. We empirically observed that using Tanh as the final activation function can remarkably improve the numerical stability of the sparse coding stage as the magnitude of the feature values is effectively bounded by the final activation function.

We adopt Adam as the optimizer, with a consistent learning rate of 1e-4. We do not apply any net regularization methods like dropout or weight decay in training. The weights of reconstruction η_0 and contrastive divergence η_1 are set to 1.0 and 0.25, respectively. Each training batch contains 4 randomly sampled training tasks with 10 query ($M = 10$) for each task. All training can be conducted on a single NVIDIA Tesla A100 GPU.

We perform 8 steps of sparse coding in the adaptive sparse coding layer, with an initial $\lambda_{\text{max}} = 0.3$ and a final $\lambda_\star = 0.05$. We perform 5 steps of Langevin dynamics with a step size of $\beta = 1.0$ to synthesize negative samples, which we show in the examples of Figure 2 and Figure F to be sufficient for producing hard negative samples.

A.2 Video Anomaly Detection

We resize each frame to 240×320. No data augmentation is performed. All other hyperparameters equal to those applied in industrial inspection experiments.

A.3 MAML-AE

We adopt a full-convolutional auto-encoder network to construct the MAML-AE. A 10 layer ResNet [18] as the encoder, and consecutive transpose-convolutional layers with batch normalization and RELU activation function as the decoder to recover the feature resolution. The hyperparameters of the episodic training of MAML AE are exactly the same with those of our methods. We perform 5 steps of gradient descent as the inner-loop adaption.

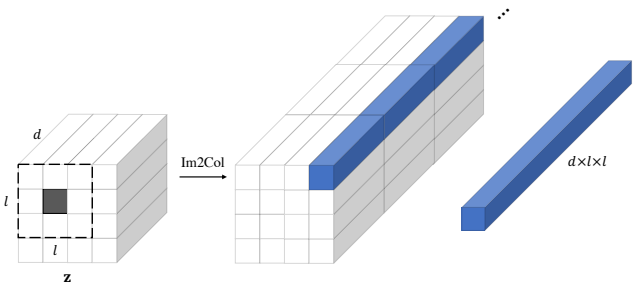

Figure A: Illustration of sparse coding layer with $l \times l$ receptive filed.

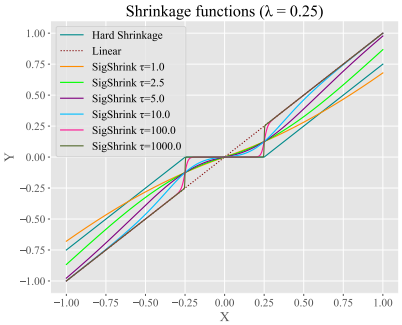

Figure B: Shrinkage functions.

We directly use the energy score in (Eqn. 8) as the anomaly score. When evaluating AUC-ROC and mIoU, we obtain the normalized energy score for each sample by uniformly normalize to the scores of all samples in test set.

A.4 Initializing Negative Sample

In the synthesis of negative samples, we randomly place at most three random patches to each normal image. Each patch is created by one of the following:

- **Random uniform:** A random patch with the values of each pixel sampled from a uniform distribution. The minimum and maximum values of the uniform distribution is equal to those of the pixel values of the images.
- **Random consistent:** A random patch with consistent pixel values sampled from a uniform distribution. The minimum and maximum values of the uniform distribution is equal to those of the pixel values of the images.
- **Random copy-paste:** A random patch randomly cropped from the same image.
- **Random blurring:** Applying Gaussian blurring to a random patch of the normal image.

Denoting $\mathcal{U}(a, b)$ as an uniform distribution with minimum and maximum values of a and b, respectively, the relative size of the random patch w.r.t. the original image is randomly sampled from $\mathcal{U}(0.0025, 0.025)$, and the aspect ratio is randomly sampled from $\mathcal{U}(0.01, 100.0)$. See Fig. E for illustrations of synthesized negative samples and the corresponding generative sequences of Langevin dynamics.

A.5 Illustrations

Sparse coding layer with receptive field. We illustrate sparse coding with receptive field in Fig. A. An $l \times l$ sliding window is applied spatially to the feature map, and the feature volume in each window is flatten as a feature vector to perform sparse coding. In this way, the context information of each position is effectively considered in the adaptive sparse coding layer.

Shrinkage functions. We plot different shrinkage function in Figure B. Beside the desired property of sparse projection, the SigShrink we use enjoys the advantages of being differential and having non-zero gradients everywhere.

A.6 Inference Steps

B Additional ablations

B.1 Robustness to Pose Variations

To validate the robustness of our method to pose variations, we perform additional experiments with pose variations and provide the quantitative comparisons in Table B. We report results by performing random $\pm 90^o$ rotation to samples during testing stage when adapting the trained model to novel categories. We observe that the textures categories (leather and grid) are robust to any pose variations in both the query and support samples. For the hazelnut category, where there are intrinsically significant pose variations across samples, performing any new rotations does not influence the results. On the other hand, for the pill category, where all samples are aligned horizontally (see examples in Appendix, Figure D), performing rotation to only query samples results in performance drop. Performing rotation to both query and support samples help to recover the performance. The

Algorithm 2 Inference steps of the proposed few-shot adaptive anomaly detection on a testing task indexed by i.

1: **Given**: Feature extractor $\mathbf{\Psi}$ with trained parameter θ.
2: **Given**: The support set $\{\mathbf{s}_k^i\}_{k=1}^K$ and query samples $\{\mathbf{q}_m^i\}_{m=1}^M$ to be tested.
3: Extract normal feature tensor $\mathbf{Z}^i \in \mathbb{R}^{K\times d\times h'\times w'}$.
4: Reshape the normal feature tensor into a matrix, and use it as the task-specific dictionary $\mathbf{D}^i \in \mathbb{R}^{d\times Kh'w'}$.
5: Estimate the anomaly score of a test sample $\mathbf{q}_m^i$ by its energy score using Eqn 8, with $E_\theta(\mathbf{q}_m^i;\mathbf{D}^i) = ||\mathbf{\Psi}(\mathbf{q}_m^i;\theta) - \mathbf{D}^i\boldsymbol{\mathcal{S}}\big(\mathbf{\Psi}(\mathbf{q}_m^i;\theta);\mathbf{D}^i\big)||^2$, where a higher energy score indicates that $\mathbf{q}_m^i$ is more likely to be an abnormal sample.
6: The pixel-wise anomaly map of a test sample $\mathbf{q}_m^i$ can be obtain by visualizing $\nabla_{\mathbf{q}_m^i} - E_\theta(\mathbf{q}_m^i;\mathbf{D}^i)$, where abnormal regions show higher magnitude.

above results suggest the robustness of our methods under the condition of sufficient sample diversity in the support set.

Category	Leather		Grid		Hazelnut		Cable		Pill	
Original	0.42	0.98	0.12	0.81	0.40	0.94	0.24	0.87	0.22	0.88
Rotate query	0.41	0.98	0.12	0.80	0.40	0.94	0.21	0.82	0.10	0.71
Rotate query and support	0.42	0.98	0.11	0.81	0.40	0.94	0.23	0.85	0.19	0.86

Table B: Robustness to pose variations. We report both mIoU (left) and AUC-ROC (right) values.

B.2 Comparison to Image Differencing

Image differencing and its variances are an intuitive approach for detecting image anomaly. We perform image differencing and background subtraction on the industrial inspection and video anomaly detection task, respectively, and report performance in Table C. Our method demonstrates clear advantages.

Category	Leather		Hazelnut		Cable		UCSD Ped1	UCSD Ped2	CUHK
Image differencing	0.02	0.57	0.12	0.77	0.07	0.65	65.34	59.03	57.12
Ours	0.42	0.98	0.40	0.94	0.24	0.87	77.42	91.22	75.32

Table C: Comparing with image differencing. We report mIoU (left) and AUC-ROC (right) for industrial inspection task and AUC-ROC for video anomaly detection task.

B.3 Robustness Against Contaminated Training Data

While it is a common practice among machine learning practitioners to assume clean training data, this may not be true in real world applications. In this section, we evaluate the robustness of our method against contaminated training data by inserting certain amount of abnormal samples. As shown in Table D, we progressively contaminate the normal training data by increasing the amount of abnormal data from 1% to 10%. Our proposed method is in general robust to data contamination, where contamination under 5% only decreases the performance slightly. The network is still able to perform decently in the extreme case of 10% contamination.

B.4 Leveraging Temporal Information

An EBM is agnostic to the underlying backbone network architecture, therefore it is straightforward to incorporate temporal information into the image-based anomaly detection framework described in Section 5.2. We replace the 2D convolutions in the first four layers with 3D convolutions, and have the model accepts a 5-frame input instead. Without heavy tuning, we report 5-shot performance on both '1 frame' (original, without temporal) and '5 frames' (with temporal) in Table E. The results show that when temporal information is added, it leads to further improvements.

B.5 Adaptive Sparse Coding Layer with Auto-encoder

Auto-encoders are widely used in anomaly detection with the localization performed by measuring the reconstruction error. An straightforward baseline of few-shot anomaly detection can be constructed by plugging the proposed adaptive sparse coding layer as the bottleneck of an autoencoder. However, we empirically find it

Contamination	Leather		Grid		Hazelnut		Cable		Pill	
0%	0.42	0.98	0.12	0.81	0.40	0.94	0.24	0.87	0.22	0.88
1%	0.42	0.98	0.11	0.80	0.40	0.94	0.23	0.86	0.21	0.88
2%	0.41	0.98	0.11	0.80	0.40	0.93	0.23	0.84	0.21	0.87
5%	0.40	0.96	0.11	0.78	0.39	0.93	0.22	0.84	0.20	0.87
10%	0.38	0.95	0.10	0.76	0.37	0.90	0.21	0.82	0.19	0.85

Table D: Performance evaluation against data contamination. We contaminate the normal training data by inserting increasing percentages of abnormal data.

Category	UCSD Ped1	UCSD Ped2	CUHK	Sh-Tech
1 frame	78.12	92.00	83.41	79.64
5 frames	79.06	91.94	84.27	80.80

Table E: Performance comparisons with temporal information incorporated.

very difficult to train a decoder that can generalize well to unseen classes. As visualized in Fig. C, the poor generalization of the decoder consistently results in drastically high reconstruction error when applied to novel test tasks. Quantitatively, this simple baseline consistently delivers nearly 0 mIoU, therefore is not included in the comparisons. On the other hand, utilizing only an encoder and using the backward gradient of the EBM to localize the anomaly demonstrate outstanding robustness on novel tasks that are unseen during the training stage as we discuss in Sec. 4.

C Qualitative Results

We present in Figure D additional anomaly localization results of categories with different anomalies in the MVTec-AD dataset.

D Additional Figures

Common practice of sampling from EBMs is computationally costly, requiring as many as 50 steps of Langevin Dynamics when initialized with full noise, instead of a synthesized negative sample as shown in Fig. E. Some examples are provided in Fig. F.

Figure C: Applying the proposed adaptive sparse coding layer to an auto-encoder for few-shot anomaly detection and localization suffers severely from the poor generalization of the decoder, which delivers consistently high reconstruction error on novel tasks.

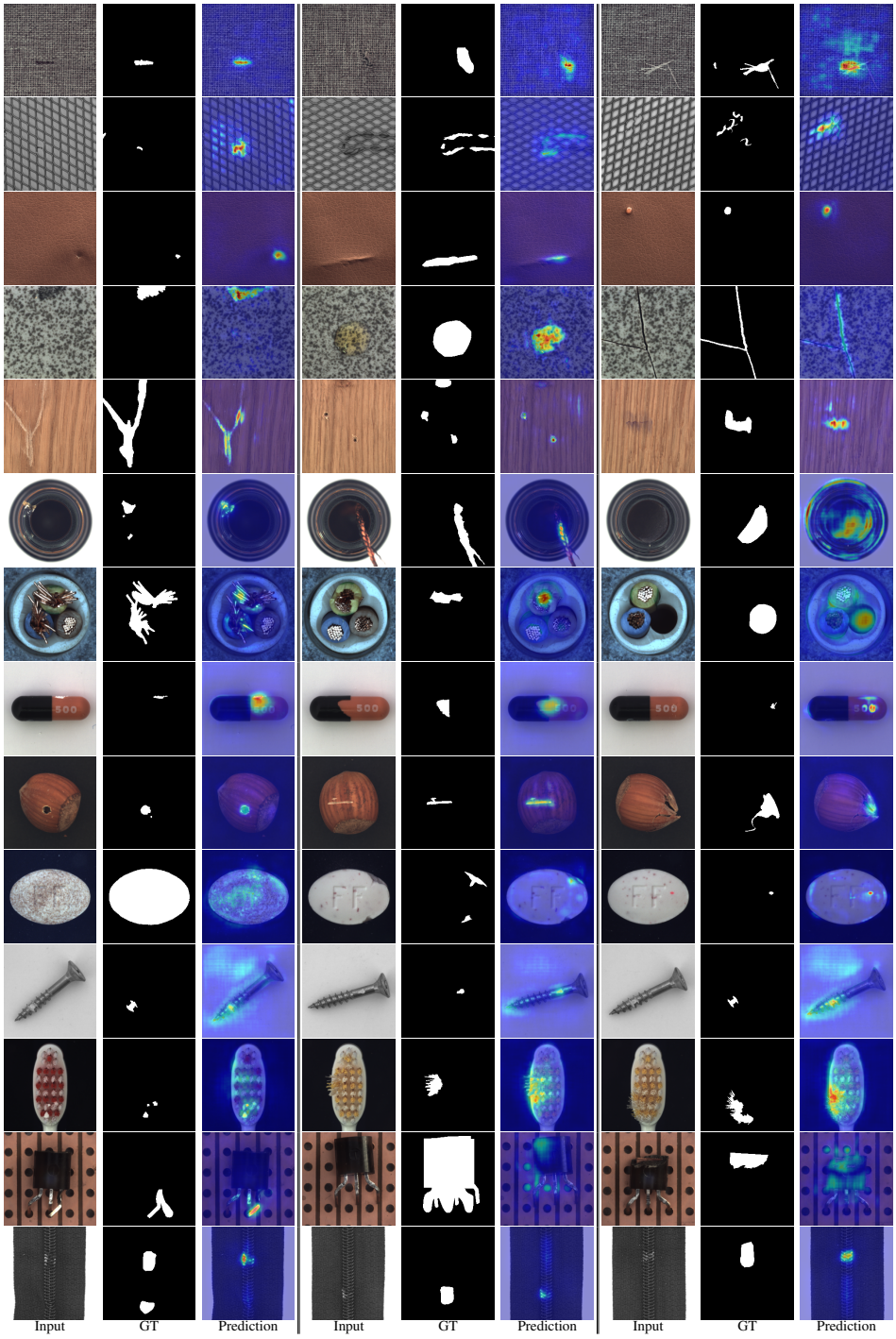

Figure D: Visualizations of anomaly localization on industrial inspection data. All results are obtained by adapting the model using 10 normal samples only.

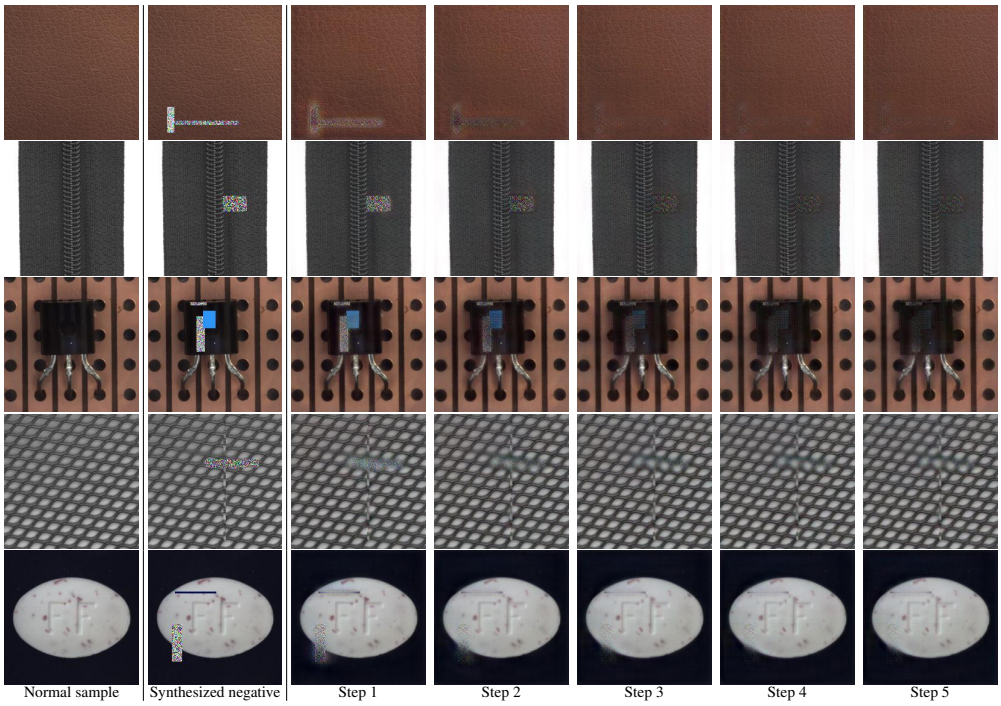

Figure E: Sampling outputs after a few steps of Langevin Dynamics starting from each synthesized negative sample. 5 steps of Langevin dynamics are sufficient to quickly generate hard negative samples with minor artifacts.

Figure F: Figure from [52]. Generated sequences from the process of Langevin Dynamics. Initialization from noise usually requires around 50 steps to synthesize the images with desired quality (images are shown every 5 steps).