

ピクセルレベルラベルを用いた適応。[22] に着想を得て、まず特徴差分マップ  $d(i, u)$  を使用して擬似フーバー損失  $\phi(u)$  を計算します。

$$\phi(u) = ((\frac{1}{C} \sum_i^C |d(i, u)|)^2 + 1)^{\frac{1}{2}} - 1. \quad (6)$$

擬似Huber損失  $\phi(u)$  は差分マップとして設計されており、訓練や拡張が容易です。ピクセルレベルラベルを用いた再構築損失関数  $\mathcal{L}_{px}$  は「プッシュ・プル損失」として記述され、

$$\mathcal{L}_{px} = \frac{1}{HW} \sum_u^{HW} (1 - y(u))\phi(u) - \alpha \log(1 - \exp(-\frac{1}{HW} \sum_u^{HW} y(u)\phi(u))), \quad (7)$$

ここで、最初の項は再構築された正常な特徴を抽出された特徴に引き寄せ、2番目の項は再構築された異常な特徴を元の特徴から遠ざけます。 $y(u)$  はピクセルレベルラベル（正常サンプルは0、異常は1）であり、 $\alpha$  は重み項です。

画像レベルラベルを用いた適応。異常サンプルには異常領域と正常領域の両方が含まれる可能性があるため、異常サンプルのすべての領域を異常領域として扱うとモデルが混乱します。 $\phi(u)$  の値が大きいほど異常領域である可能性が高いことを考慮し、まず  $\phi(u)$  の最大値  $k$  個を収集し、その平均を画像の異常スコアとして計算します。

$$q = \frac{1}{k} \sum \text{top\_k}(\phi). \quad (8)$$

次に、画像レベルの損失関数  $\mathcal{L}_{img}$  は次のように計算されます。

$$\mathcal{L}_{img} = (1 - y)q - \alpha y \log(1 - \exp(-q)), \quad (9)$$

ここで  $y$  は画像レベルのラベル（正常サンプルは0、異常は1）であり、 $\alpha$  は重み項です。 $\mathcal{L}_{img}$  では、最初の項は正常サンプルの再構築特徴量を抽出された特徴量に近づけ、2番目の項は異常の再構築特徴量を抽出された特徴量から遠ざけます。

## 4 Experiment

### 4.1 Dataset

MVTec-AD [4] は、15 種類のカテゴリを有する多カテゴリ・多欠陥産業異常検出データセットです。ground-truth には画像ラベルと異常セグメンテーションが含まれます。正常サンプルのみのケースでは、元の設定に従い正常

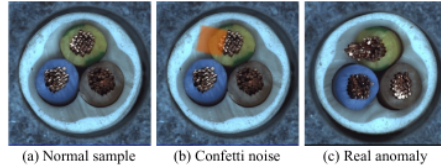


図3：正常サンプルにコンフェッティノイズを追加して生成した合成異常例。

**Adaptation with pixel-level labels.** Inspired by [22], we firstly calculate a pseudo-Huber loss,  $\phi(u)$ , using the feature difference map,  $\mathbf{d}(i, u)$ .

$$\phi(u) = ((\frac{1}{C} \sum_i^C |\mathbf{d}(i, u)|)^2 + 1)^{\frac{1}{2}} - 1. \quad (6)$$

The pseudo-Huber loss,  $\phi(u)$ , is designed as a difference map, which is easy to train and extend. Then the reconstruction loss function with pixel-level labels is denoted as  $\mathcal{L}_{px}$  and could be described as a “push-pull loss” as,

$$\mathcal{L}_{px} = \frac{1}{HW} \sum_u^{HW} (1 - \mathbf{y}(u))\phi(u) - \alpha \log(1 - \exp(-\frac{1}{HW} \sum_u^{HW} \mathbf{y}(u)\phi(u))), \quad (7)$$

where the first term pulls the reconstructed normal features to the extracted features, and the second term pushes the reconstructed anomalous features away from the original features,  $\mathbf{y}(u)$  is the pixel-level label (0 for normal sample and 1 for anomaly) and  $\alpha$  is a weight term.

**Adaptation with image-level labels.** Since anomaly samples could contain both anomalous and normal regions, simply treating all regions of anomaly samples as anomalous regions confuses the model. Considering that larger values in  $\phi(u)$  are more likely to be anomalous regions, we firstly collect  $k$  maximum values of  $\phi(u)$ , then calculate their mean as the anomaly score of the image.

$$q = \frac{1}{k} \sum \text{top-}k(\phi). \quad (8)$$

Then the image-level loss,  $\mathcal{L}_{img}$ , could be calculated as,

$$\mathcal{L}_{img} = (1 - y)q - \alpha y \log(1 - \exp(-q)), \quad (9)$$

where  $y$  is the image-level label (0 for normal sample and 1 for anomaly) and  $\alpha$  is a weight term. In  $\mathcal{L}_{img}$ , the first term pulls the reconstructed features of normal samples towards the extracted features, while the second term pushes the reconstructed features of anomalies away from the extracted features.

## 4 Experiment

### 4.1 Dataset

MVTec-AD [4] is a multi-category, multi-defect, industrial anomaly detection dataset with 15 categories. The ground-truth includes both image labels and anomaly segmentation. In *normal-sample-only case*, we follow the original setting to use normal

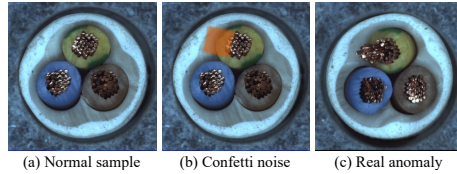


Fig. 3: **Synthetic anomalies** by adding confetti noise on normal samples.