Supplementary Material

Few-shot Anomaly Detection via Personalization

A. Implementation details

Concept learning. Unless specified otherwise, we maintain the original hyperparameter choices from LDM (Rombach et al., 2021). The batch size is set to 4, and the base learning rate is set to 5.0×10^{-4}. Both MVTec-AD (Bergmann et al., 2019) and VisA (Zou et al., 2022) datasets are resized to a resolution of 512×512. All results are obtained after 3,000 optimization steps.

For normal-aware concept learning, α, regularization hyperparameter for aligning state prompts with images, is set to 0.003. We find that applying large α value results over-fitting to normal state prompt, *e.g.*, "a photo of a flawless $\mathbf{c}_n^*$", with the given image. For anomaly-aware concept learning, we first synthesize pseudo-anomalies via pre-trained text-to-image diffusion model (Meng et al., 2021). Specifically, with the given reference images, we set the *strength* parameter, *i.e.*, the amount of noise initially added to the given image, as 0.5. The guidance scale and number of inference steps are set to 7.5 and 30 respectively. We explore diverse amount of noise, and set which is distinguishable with normal samples while maintaining the high-level features of the images. In Figure 2 and Figure 4, pseudo-anomalies with different strength is shown. Hyperparameter α is set to 0.002 and γ, distance regularization term between anomaly prompts and pseudo-anomalies, is set and 0.8. We explore several γ values, while small γ values results normal state optimization, *i.e.*, minimizing the distance between CLIP embeddings of the images and normal state prompt, unstable.

Overview of WinCLIP+. We present the details for incorporating visual features (*i.e.*, feature maps) in computing anomaly detection score which is described in Section 3.3. WinCLIP+ (Jeong et al., 2023) introduces the *reference association* module, which enables the storage and retrieval of memory features $\mathbf{R}$ for a given set of images $\mathcal{D} \coloneqq \{\mathbf{x}_i\}_{i=1}^{K}$ based on cosine similarity. Using the reference association module along with the corresponding features $\mathbf{F} \in \mathbb{R}^{h \times w \times d}$ extracted from a query image (e.g., patch-level features), we can generate a prediction $\mathbf{M} \in [0,1]^{h \times w}$ for each pixel. The prediction is computed as follows:

$$\mathbf{M}_{ij} \coloneqq \min_{r \in \mathbf{R}} \tfrac{1}{2}(1 - \langle \mathbf{F}_{ij}, r \rangle). \tag{8}$$

To compute the prediction, three different features are incorporated: small-scale feature $\mathbf{F}^{\mathtt{W}}\mathtt{s}$, mid-scale feature $\mathbf{F}^{\mathtt{W}}\mathtt{m}$, and penultimate feature $\mathbf{F}^{\mathtt{P}}$. By applying the reference association module, we obtain three reference memories: $\mathbf{R}_{\mathtt{s}}^{\mathtt{W}}$, $\mathbf{R}_{\mathtt{m}}^{\mathtt{W}}$, and $\mathbf{R}^{\mathtt{P}}$. Then we compute the average of multi-scale prediction (8), and it is given as:

$$\mathbf{M}^{\mathtt{W}} \coloneqq \tfrac{1}{3}(\mathbf{M}^{\mathtt{P}} + \mathbf{M}_{\mathtt{s}}^{\mathtt{W}} + \mathbf{M}_{\mathtt{m}}^{\mathtt{W}}). \tag{9}$$

Subsequently, the maximum value of $\mathbf{M}^{\mathtt{W}}$ is incorporated into the ADP anomaly detection score (7). This score captures complementary information derived from the spatial features of the few-shot references. The complete form of ADP anomaly detection ($\mathtt{ADP}_{ad}$) is as follows:

$$\mathtt{ADP}(\mathbf{x})_{ad} \coloneqq \frac{1}{2}\Big(\mathtt{ADP}(\mathbf{x}) + \max_{ij} \mathbf{M}_{ij}^{\mathtt{W}}\Big). \tag{10}$$

Anomaly detection with learned concepts. For anomaly detection, we generate 20 pseudo-anomalies for pseudo-validation set. We set *strength* parameter, *i.e.*, the amount of noise initially added to the given image, as 0.5. The guidance scale and number of inference steps are set to 7.5 and 30 respectively. We employ the data pre-processing pipeline from OpenCLIP (Ilharco et al., 2021) for both MVTec-AD and VisA datasets. This pipeline includes channel-wise standardization using the pre-computed mean `[0.48145466, 0.4578275, 0.40821073]` and standard deviation `[0.26862954, 0.26130258, 0.27577711]` after normalizing each RGB image to the range of $[0, 1]$. Additionally, we set the input resolution to be 224 by default, regardless of the original size of the input image. When reproducing the results for WinCLIP+ (Jeong et al., 2023), we follow the same pre-processing pipeline to ensure compatibility in our experiments.

Datasets. MVTec-AD comprises 15 sub-datasets with a total of 5,354 images, where 1,725 of which are in the test set. 15 sub-datasets are further divided into 10 object categories and 5 texture categories. VisA consists of 12 sub-datasets with 10,821 images in total. Anomalous images in VisA contain a variety of imperfections, including surface defects and structural defects. We follow the index given by Zou et al. (2022) for splitting the VisA dataset into train and test sets.

Computation. We use 64 CPU cores (Intel Xeon CPU @ 2.90GHz) and 1 GPU (NVIDIA GeForce RTX 3090 24GB GPU) for performing concept learning. The training for 3,000 optimization steps takes approximately 1.5 hours for each class. We need two times of concept learning *i.e.*, normal-aware concept learning and anomaly-aware concept learning, which takes similar time for each concept learning. When considering the entire MVTec-AD dataset, which consists of 15 classes, the complete concept learning process takes 45 hours using a single GPU. For anomaly detection (Section 3.3), we use same machine which takes approximately 1 hour for the entire dataset both on MVTec-AD and VisA.

B. Additional quantitative results

Class-wise comparison. We provide a detailed anomaly detection (AD) performance, specifically in terms of class-wise AUROC (%). For the 2-shot and 4-shot scenarios, we report the mean and standard deviation over three random seeds for WinCLIP+ (Jeong et al., 2023), ADP and ADP$_\ell$, while other baselines (SPADE (Cohen & Hoshen, 2020), PaDiM (Defard et al., 2021) and PatchCore (Roth et al., 2022)) are from those reported by Jeong et al. (2023). The class-wise AUROC (%) results for the MVTec-AD dataset are presented in Table 3 for the 2-shot and 4-shot settings. Similarly, the class-wise AUROC results for the VisA dataset can be found in Table 4 for the 2-shot and 4-shot settings. Additionally, we compare our 8-shot AD results with other 8-shot AD methods on MVTec-AD dataset in Table 5. The results for TDG+ (Sheynin et al., 2021), DiffNet+ (Rudolph et al., 2021) and RegAD (Huang et al., 2022) are from the work of Huang et al. (2022).

Comparison with many-shot methods. Table 6 provides a comparison with the full-shot results of various prior works on the MVTec-AD dataset. In the 4-shot scenario, ADP surpasses the performance of CutPaste (Li et al., 2021), a recent full-shot method for AD and is competitive with Metaformer (Wu et al., 2021). While PatchCore (Roth et al., 2022) shows gratifying performance with full-shot, ADP outperforms to prior works such as MKD (Salehi et al., 2021) and P-SVDD (Yi & Yoon, 2020) with large margin. Furthermore, our 2-shot ADP achieves superior performance compared to recent few-shot AD methods such as DiffNet+ (Rudolph et al., 2021), TDG+ (Sheynin et al., 2021), and RegAD (Huang et al., 2022), despite utilizing only 2 shots instead of the 16 shots used by these methods.

Comparison with textual inversion. Table 7 and 8 present a class-wise comparison between standard textual inversion (referred to as "TI" in Table 7 and 8) and the utilization of learned concepts. The evaluation is conducted on 4-shot anomaly detection tasks in MVTec-AD and VisA datasets, respectively. The results are represented by $\mathbf{c}_n^*$ and $\mathbf{c}_a^*$, which indicate the outcomes obtained by incorporating only $\mathbf{c}_n^*$ and $\mathbf{c}_a^*$ in the text prompts. Furthermore, $\mathbf{c}_n^* + \mathbf{c}_a^*$ represents the combination of both concepts via ADP (Section 3.3). In general, the inclusion of concepts leads to a notable improvement in anomaly detection performance. While the utilization of only $\mathbf{c}_a^*$ does not yield significant enhancements, combining $\mathbf{c}_n^*$ and $\mathbf{c}_a^*$ proves to be mutually beneficial. Specifically, the incorporation of learned concepts proves effective in identifying fine-grained anomalies, such as the "Capsule" class in the MVTec-AD dataset or the "PCB" classes in the VisA dataset.

Table 3. Comparison of anomaly detection (AD) in terms of class-wise AUROC (%) on MVTec-AD for 2- and 4-shot.

	2-shot						4-shot					
Data \ Method	SPADE	PaDiM	PatchCore	WinCLIP+	**ADP**	**ADP$_\ell$**	SPADE	PaDiM	PatchCore	WinCLIP+	**ADP**	**ADP$_\ell$**
Bottle	99.5$_{\pm 0.1}$	98.5$_{\pm 1.0}$	99.2$_{\pm 0.3}$	93.3$_{\pm 0.1}$	97.1$_{\pm 1.5}$	95.1$_{\pm 0.4}$	99.5$_{\pm 0.2}$	98.8$_{\pm 0.2}$	99.2$_{\pm 0.3}$	93.4$_{\pm 0.3}$	98.9$_{\pm 0.4}$	97.2$_{\pm 0.7}$
Cable	76.2$_{\pm 5.2}$	62.3$_{\pm 5.9}$	91.0$_{\pm 2.7}$	82.6$_{\pm 0.2}$	85.7$_{\pm 1.5}$	86.2$_{\pm 1.8}$	83.4$_{\pm 3.1}$	70.0$_{\pm 6.1}$	91.0$_{\pm 2.7}$	83.0$_{\pm 0.0}$	87.9$_{\pm 2.5}$	88.3$_{\pm 3.1}$
Capsule	70.9$_{\pm 6.1}$	64.3$_{\pm 3.0}$	72.8$_{\pm 7.0}$	84.2$_{\pm 9.0}$	85.0$_{\pm 12.9}$	85.3$_{\pm 12.1}$	78.9$_{\pm 5.5}$	65.2$_{\pm 2.5}$	72.8$_{\pm 7.0}$	84.4$_{\pm 9.4}$	83.4$_{\pm 11.9}$	84.0$_{\pm 11.5}$
Carpet	98.3$_{\pm 0.4}$	97.8$_{\pm 0.5}$	96.6$_{\pm 0.5}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$	98.6$_{\pm 0.2}$	97.9$_{\pm 0.4}$	96.6$_{\pm 0.5}$	100$_{\pm 0.0}$	99.9$_{\pm 0.1}$	100$_{\pm 0.0}$
Grid	41.3$_{\pm 3.6}$	67.2$_{\pm 4.2}$	67.7$_{\pm 8.3}$	99.2$_{\pm 0.0}$	97.4$_{\pm 0.7}$	98.6$_{\pm 0.0}$	44.6$_{\pm 6.6}$	68.1$_{\pm 3.8}$	67.7$_{\pm 8.3}$	99.1$_{\pm 0.2}$	98.0$_{\pm 2.5}$	99.5$_{\pm 0.6}$
Hazelnut	96.2$_{\pm 2.1}$	90.8$_{\pm 0.8}$	93.2$_{\pm 3.8}$	97.0$_{\pm 0.6}$	98.8$_{\pm 0.9}$	98.3$_{\pm 0.9}$	98.4$_{\pm 1.3}$	91.9$_{\pm 1.2}$	93.2$_{\pm 3.8}$	97.5$_{\pm 0.1}$	99.4$_{\pm 0.5}$	98.9$_{\pm 0.4}$
Leather	100$_{\pm 0.0}$	97.5$_{\pm 0.9}$	97.9$_{\pm 0.7}$	100$_{\pm 0.0}$	93.1$_{\pm 11.9}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$	98.5$_{\pm 0.2}$	97.9$_{\pm 0.7}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$
Metal nut	77.0$_{\pm 7.9}$	54.8$_{\pm 3.8}$	77.7$_{\pm 8.5}$	95.5$_{\pm 0.3}$	99.7$_{\pm 0.3}$	99.1$_{\pm 0.1}$	77.8$_{\pm 5.7}$	60.7$_{\pm 5.2}$	77.7$_{\pm 8.5}$	95.7$_{\pm 0.3}$	99.4$_{\pm 0.5}$	99.6$_{\pm 0.2}$
Pill	84.8$_{\pm 0.9}$	59.1$_{\pm 6.4}$	82.9$_{\pm 2.9}$	90.0$_{\pm 0.2}$	95.2$_{\pm 0.4}$	95.2$_{\pm 1.0}$	86.7$_{\pm 0.3}$	54.9$_{\pm 2.7}$	82.9$_{\pm 2.9}$	90.1$_{\pm 0.1}$	95.2$_{\pm 0.3}$	94.9$_{\pm 0.6}$
Screw	46.6$_{\pm 2.2}$	54.0$_{\pm 4.4}$	49.0$_{\pm 3.8}$	96.5$_{\pm 0.2}$	91.9$_{\pm 5.6}$	94.8$_{\pm 3.3}$	50.5$_{\pm 5.4}$	50.0$_{\pm 4.1}$	49.0$_{\pm 3.8}$	96.8$_{\pm 0.3}$	90.9$_{\pm 2.6}$	94.1$_{\pm 2.1}$
Tile	99.9$_{\pm 0.1}$	93.3$_{\pm 1.1}$	98.5$_{\pm 1.0}$	99.4$_{\pm 0.0}$	99.5$_{\pm 0.2}$	99.6$_{\pm 0.1}$	100$_{\pm 0.0}$	93.1$_{\pm 0.6}$	98.5$_{\pm 1.0}$	99.4$_{\pm 0.0}$	99.8$_{\pm 0.1}$	99.7$_{\pm 0.1}$
Toothbrush	78.6$_{\pm 3.2}$	87.6$_{\pm 4.2}$	85.9$_{\pm 3.5}$	94.0$_{\pm 0.6}$	88.5$_{\pm 3.8}$	95.0$_{\pm 4.3}$	78.8$_{\pm 5.2}$	89.2$_{\pm 2.5}$	85.9$_{\pm 3.5}$	93.8$_{\pm 0.2}$	96.6$_{\pm 4.2}$	98.6$_{\pm 1.0}$
Transistor	83.4$_{\pm 3.8}$	81.3$_{\pm 3.7}$	72.8$_{\pm 6.3}$	82.4$_{\pm 0.4}$	82.3$_{\pm 5.7}$	87.5$_{\pm 2.4}$	81.4$_{\pm 2.1}$	82.4$_{\pm 6.5}$	90.0$_{\pm 4.3}$	83.0$_{\pm 0.3}$	89.3$_{\pm 2.9}$	90.0$_{\pm 1.9}$
Wood	99.2$_{\pm 0.4}$	96.9$_{\pm 0.5}$	98.3$_{\pm 0.6}$	100$_{\pm 0.0}$	99.9$_{\pm 0.3}$	99.9$_{\pm 0.1}$	98.9$_{\pm 0.6}$	97.0$_{\pm 0.2}$	98.3$_{\pm 0.6}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$
Zipper	93.3$_{\pm 2.9}$	86.3$_{\pm 2.6}$	94.0$_{\pm 2.1}$	92.4$_{\pm 4.4}$	95.5$_{\pm 5.6}$	95.8$_{\pm 5.4}$	95.1$_{\pm 1.3}$	88.3$_{\pm 2.0}$	94.0$_{\pm 2.1}$	95.4$_{\pm 0.7}$	94.9$_{\pm 6.5}$	98.8$_{\pm 0.3}$
Mean	82.9$_{\pm 2.6}$	78.9$_{\pm 3.1}$	86.3$_{\pm 3.3}$	93.8$_{\pm 1.0}$	94.4$_{\pm 1.2}$	95.4$_{\pm 0.9}$	84.8$_{\pm 2.5}$	80.4$_{\pm 2.5}$	88.8$_{\pm 2.6}$	94.1$_{\pm 0.7}$	95.8$_{\pm 1.1}$	96.2$_{\pm 0.8}$

Table 4. Comparison of anomaly detection (AD) in terms of class-wise AUROC (%) on VisA for 2- and 4-shot.

	2-shot						4-shot					
Data \ Method	SPADE	PaDiM	PatchCore	WinCLIP+	**ADP**	**ADP$_\ell$**	SPADE	PaDiM	PatchCore	WinCLIP+	**ADP**	**ADP$_\ell$**
Candle	91.3$_{\pm 3.3}$	75.8$_{\pm 2.1}$	85.3$_{\pm 1.5}$	95.3$_{\pm 0.4}$	94.1$_{\pm 1.6}$	95.1$_{\pm 1.6}$	92.8$_{\pm 2.1}$	77.5$_{\pm 1.6}$	87.8$_{\pm 0.8}$	95.4$_{\pm 0.7}$	92.5$_{\pm 1.5}$	94.0$_{\pm 1.2}$
Capsules	71.7$_{\pm 11.2}$	51.7$_{\pm 4.6}$	57.8$_{\pm 5.4}$	82.2$_{\pm 5.5}$	84.4$_{\pm 4.1}$	84.7$_{\pm 4.1}$	73.4$_{\pm 7.1}$	52.7$_{\pm 3.4}$	63.4$_{\pm 5.4}$	81.8$_{\pm 6.7}$	87.3$_{\pm 0.7}$	87.4$_{\pm 0.5}$
Cashew	97.3$_{\pm 1.4}$	74.6$_{\pm 3.6}$	93.6$_{\pm 0.6}$	88.9$_{\pm 0.8}$	91.5$_{\pm 4.3}$	91.6$_{\pm 3.9}$	96.4$_{\pm 1.3}$	77.7$_{\pm 3.2}$	93.0$_{\pm 1.5}$	88.9$_{\pm 0.9}$	91.7$_{\pm 1.7}$	91.7$_{\pm 2.1}$
Chewinggum	93.4$_{\pm 1.0}$	82.7$_{\pm 2.1}$	97.8$_{\pm 0.6}$	94.6$_{\pm 0.3}$	98.2$_{\pm 0.5}$	98.1$_{\pm 0.7}$	93.5$_{\pm 1.4}$	83.5$_{\pm 3.7}$	98.3$_{\pm 0.3}$	95.1$_{\pm 0.1}$	97.7$_{\pm 0.6}$	97.9$_{\pm 0.1}$
Fryum	90.5$_{\pm 3.9}$	69.2$_{\pm 9.0}$	83.4$_{\pm 2.4}$	87.7$_{\pm 0.3}$	93.6$_{\pm 1.4}$	91.9$_{\pm 2.1}$	92.9$_{\pm 1.6}$	71.2$_{\pm 5.9}$	88.6$_{\pm 1.3}$	87.7$_{\pm 0.4}$	94.6$_{\pm 2.0}$	94.0$_{\pm 1.9}$
Macaroni1	69.1$_{\pm 8.2}$	62.2$_{\pm 5.0}$	75.6$_{\pm 4.6}$	91.1$_{\pm 0.6}$	91.1$_{\pm 3.7}$	92.9$_{\pm 3.4}$	65.8$_{\pm 1.2}$	65.9$_{\pm 3.9}$	82.9$_{\pm 2.7}$	91.3$_{\pm 0.8}$	91.4$_{\pm 3.3}$	91.9$_{\pm 2.0}$
Macaroni2	58.3$_{\pm 4.4}$	50.8$_{\pm 2.9}$	57.3$_{\pm 5.6}$	74.7$_{\pm 1.5}$	76.1$_{\pm 4.7}$	76.7$_{\pm 5.2}$	56.7$_{\pm 3.2}$	55.0$_{\pm 2.9}$	61.7$_{\pm 1.8}$	74.6$_{\pm 1.7}$	71.7$_{\pm 3.4}$	72.5$_{\pm 2.4}$
PCB1	86.7$_{\pm 1.1}$	62.4$_{\pm 10.8}$	71.5$_{\pm 20.0}$	87.7$_{\pm 0.4}$	80.1$_{\pm 13.4}$	83.9$_{\pm 9.4}$	83.4$_{\pm 8.5}$	82.6$_{\pm 1.5}$	84.7$_{\pm 6.7}$	88.1$_{\pm 0.3}$	87.7$_{\pm 1.5}$	90.4$_{\pm 1.7}$
PCB2	70.3$_{\pm 8.1}$	66.8$_{\pm 2.0}$	84.3$_{\pm 1.7}$	61.9$_{\pm 1.6}$	71.3$_{\pm 3.0}$	71.1$_{\pm 2.9}$	71.7$_{\pm 7.0}$	73.5$_{\pm 2.4}$	84.3$_{\pm 1.0}$	63.1$_{\pm 1.5}$	74.3$_{\pm 2.7}$	73.8$_{\pm 2.1}$
PCB3	75.8$_{\pm 5.7}$	67.3$_{\pm 3.8}$	84.8$_{\pm 1.2}$	70.2$_{\pm 0.5}$	64.0$_{\pm 1.0}$	67.0$_{\pm 2.6}$	79.0$_{\pm 4.1}$	65.9$_{\pm 1.9}$	87.0$_{\pm 1.1}$	70.1$_{\pm 1.2}$	67.8$_{\pm 9.6}$	71.4$_{\pm 6.4}$
PCB4	86.1$_{\pm 8.2}$	69.3$_{\pm 13.7}$	94.3$_{\pm 3.2}$	83.0$_{\pm 5.2}$	86.3$_{\pm 10.6}$	90.4$_{\pm 6.2}$	95.4$_{\pm 2.3}$	85.4$_{\pm 2.0}$	95.6$_{\pm 1.6}$	85.6$_{\pm 4.1}$	96.7$_{\pm 0.8}$	97.1$_{\pm 0.9}$
Pipe fryum	78.1$_{\pm 3.0}$	75.3$_{\pm 1.8}$	93.5$_{\pm 1.3}$	93.3$_{\pm 0.1}$	98.3$_{\pm 1.9}$	98.9$_{\pm 1.1}$	79.3$_{\pm 0.9}$	82.9$_{\pm 2.2}$	96.4$_{\pm 0.7}$	93.4$_{\pm 0.0}$	99.1$_{\pm 0.2}$	99.2$_{\pm 0.4}$
Mean	80.7$_{\pm 5.0}$	67.4$_{\pm 5.1}$	81.6$_{\pm 4.0}$	84.2$_{\pm 0.2}$	85.7$_{\pm 0.9}$	86.9$_{\pm 0.9}$	81.7$_{\pm 3.4}$	72.8$_{\pm 2.9}$	85.3$_{\pm 2.1}$	84.6$_{\pm 0.4}$	87.7$_{\pm 0.3}$	88.4$_{\pm 0.4}$

Table 5. Comparison of anomaly detection (AD) with existing 8-shot AD in terms of class-wise AUROC (%) on MVTec-AD for 8-shot.

	8-shot					
Data \ Method	TDG+	DiffNet+	RegAD	WinCLIP+	**ADP**	**ADP$_\ell$**
Bottle	70.3	99.4	99.8	93.7$_{\pm 0.1}$	99.4$_{\pm 0.3}$	97.5$_{\pm 1.0}$
Cable	74.7	87.9	80.6	83.0$_{\pm 0.1}$	88.0$_{\pm 1.9}$	88.5$_{\pm 2.4}$
Capsule	44.7	78.6	76.3	90.9$_{\pm 1.4}$	93.1$_{\pm 1.7}$	93.0$_{\pm 1.5}$
Carpet	78.2	78.5	98.5	100$_{\pm 0.0}$	99.5$_{\pm 0.7}$	99.7$_{\pm 0.4}$
Grid	87.6	78.5	91.5	99.0$_{\pm 0.5}$	98.2$_{\pm 1.7}$	99.4$_{\pm 0.4}$
Hazelnut	82.8	97.9	96.5	97.7$_{\pm 0.1}$	99.5$_{\pm 0.7}$	99.1$_{\pm 0.5}$
Leather	93.5	92.2	100	100$_{\pm 0.0}$	100$_{\pm 0.0}$	100$_{\pm 0.0}$
Metal nut	68.7	67.6	98.3	95.8$_{\pm 0.4}$	99.6$_{\pm 0.4}$	99.6$_{\pm 0.2}$
Pill	67.9	82.1	80.6	90.1$_{\pm 0.1}$	95.6$_{\pm 0.5}$	94.9$_{\pm 0.5}$
Screw	99.0	75.0	63.4	96.9$_{\pm 0.3}$	91.2$_{\pm 0.8}$	94.5$_{\pm 1.0}$
Tile	87.4	99.6	97.4	99.5$_{\pm 0.1}$	99.8$_{\pm 0.1}$	99.8$_{\pm 0.0}$
Toothbrush	57.6	60.8	98.5	93.5$_{\pm 0.2}$	99.3$_{\pm 1.3}$	98.8$_{\pm 1.3}$
Transistor	71.5	63.3	93.4	83.4$_{\pm 0.1}$	90.0$_{\pm 2.4}$	90.6$_{\pm 1.9}$
Wood	98.4	99.4	99.4	100$_{\pm 0.0}$	100$_{\pm 0.1}$	100$_{\pm 0.0}$
Zipper	66.3	87.3	94.0	96.1$_{\pm 0.2}$	99.2$_{\pm 0.2}$	99.2$_{\pm 0.2}$
Mean	76.6	83.2	91.2	94.6$_{\pm 0.1}$	96.8$_{\pm 0.4}$	97.0$_{\pm 0.2}$

Table 6. Comparison with existing many-shot AD methods in terms of AUROC (%) on MVTec-AD.

Methods	Setup	AD
ADP (ours)	2-shot	94.4
ADP (ours)	4-shot	95.8
ADP (ours)	8-shot	96.8
ADP (ours)	16-shot	97.1
TDG+	16-shot	78.0
DiffNet+	16-shot	87.3
RegAD	16-shot	92.7
MKD	full-shot	87.7
P-SVDD	full-shot	92.1
CutPaste	full-shot	95.2
Metaformer	full-shot	95.8
PatchCore	full-shot	99.6

Table 7. Comparison of anomaly detection (AD) in terms of class-wise AUROC (%) with naïve textual inversion and across the use of learned concepts in MVTec-AD for 4-shot. Naïve textual inversion is denoted as "TI".

Data \ Method	TI	$\mathbf{c}_n^*$	$\mathbf{c}_a^*$	$\mathbf{c}_n^* + \mathbf{c}_a^*$
Bottle	91.5	99.4	97.9	98.6
Cable	76.3	90.3	90.9	90.6
Capsule	61.8	88.5	90.0	88.6
Carpet	100	100	100	100
Grid	99.6	99.0	93.7	95.2
Hazelnut	94.4	99.7	98.4	99.8
Leather	88.0	100	100	100
Metal nut	98.8	98.0	99.2	98.9
Pill	84.5	94.4	95.4	95.1
Screw	92.1	84.6	86.2	88.8
Tile	99.4	99.5	99.6	99.8
Toothbrush	75.0	99.4	92.8	100
Transistor	71.3	86.9	86.0	86.3
Wood	98.1	100	100	100
Zipper	98.5	98.5	98.3	98.4
Mean	88.6	95.9	95.2	96.0

Table 8. Comparison of anomaly detection (AD) in terms of class-wise AUROC (%) with naïve textual inversion and across the use of learned concepts in VisA for 4-shot. Naïve textual inversion is denoted as "TI".

Data \ Method	TI	$\mathbf{c}_n^*$	$\mathbf{c}_a^*$	$\mathbf{c}_n^* + \mathbf{c}_a^*$
Candle	97.0	90.0	91.4	90.9
Capsules	88.0	88.5	77.6	87.5
Cashew	76.6	92.4	91.5	92.6
Chewinggum	97.4	98.8	96.9	97.6
Fryum	51.1	96.3	96.1	96.4
Macaroni1	84.9	92.9	81.7	87.7
Macaroni2	66.4	62.0	69.0	67.9
PCB1	60.5	85.0	91.4	88.8
PCB2	65.9	77.6	63.6	72.9
PCB3	68.0	69.4	69.9	75.0
PCB4	88.9	94.3	96.1	96.3
Pipe fryum	97.6	98.1	99.0	99.0
Mean	78.5	87.1	85.3	87.7

C. Additional qualitative results

In Figure 2-5, we present additional qualitative results of pseudo-anomalies synthesized using the pre-trained text-to-image diffusion model (Meng et al., 2021) for both VisA (Zou et al., 2022) and MVTec-AD (Bergmann et al., 2019) datasets. We adjust the level of noise added to the reference image, denoted as `S`. We explore four different noise level, 0.1, 0.3, 0.5 and 0.7. Figure 2 and Figure 4 showcase the pseudo-anomalies conditioned with a simple text prompt, as described in Section 3.2, for the VisA and MVTec-AD datasets, respectively. On the other hand, Figure 3 and Figure 5 demonstrate the pseudo-anomalies conditioned with a prompt incorporating $\mathbf{c}_a^*$, as described in Section 3.3, for the VisA and MVTec-AD datasets, respectively. Overall, incorporating $\mathbf{c}_a^*$ in the conditioning prompt generates more fine-grained anomalies compared to the simple text prompt. Synthesizing pseudo-anomalies using pre-trained text-to-image diffusion models allows for better control over different noise levels and prompts, depending on the context.

D. Prompt templates

Below we provide the list of text templates used when learning the state-aware concept and detecting anomaly where $S \in \{S_n, S_c\}$ are state templates and $\mathbf{c} \in \{\mathbf{c}_n, \mathbf{c}_a\}$ are concepts:

- "a photo of a $S(\mathbf{c})$",
- "a rendering of a $S(\mathbf{c})$.",
- "a cropped photo of the $S(\mathbf{c})$.",
- "the photo of a $S(\mathbf{c})$.",
- "a photo of a clean $S(\mathbf{c})$.",
- "a photo of a dirty $S(\mathbf{c})$.",
- "a dark photo of the $S(\mathbf{c})$.",
- "a photo of my $S(\mathbf{c})$.",
- "a photo of the cool $S(\mathbf{c})$.",
- "a close-up photo of a $S(\mathbf{c})$.",
- "a bright photo of the $S(\mathbf{c})$.",
- "a cropped photo of a $S(\mathbf{c})$.",
- "a photo of the $S(\mathbf{c})$.",
- "a good photo of the $S(\mathbf{c})$.",
- "a photo of one $S(\mathbf{c})$.",
- "a close-up photo of the $S(\mathbf{c})$.",
- "a rendition of the $S(\mathbf{c})$.",
- "a photo of the clean $S(\mathbf{c})$.",
- "a rendition of a $S(\mathbf{c})$.",
- "a photo of a nice $S(\mathbf{c})$.",
- "a good photo of a $S(\mathbf{c})$.",
- "a photo of the nice $S(\mathbf{c})$.",
- "a photo of the small $S(\mathbf{c})$.",
- "a photo of the weird $S(\mathbf{c})$.",
- "a photo of the large $S(\mathbf{c})$.",
- "a photo of a cool $S(\mathbf{c})$.",
- "a photo of a small $S(\mathbf{c})$.",

E. Limitation and future work

Despite its strong performances in few-shot AD, we expect the effectiveness of current ADP may saturate earlier as more normal samples become available, *e.g.*, compared to other approaches such as PatchCore (Roth et al., 2022): the current technique of textual inversion is known to fall short with many samples, *e.g.*, more than 4-5 in practice (Gal et al., 2022). Making textual inversion to extract better concepts from many samples would be an interesting future work itself, not only in the context of AD but also in the context of generative modeling.

F. Potential negative social impact

Abilities in performing anomaly-aware few-shot personalization could be potentially misused in face identification and generation. This may raise several privacy issues, for example, one can extract someone's personal information very efficiently from cameras in public spaces. It is an interesting research direction to protect information from personalization techniques like textual inversion or ADP.

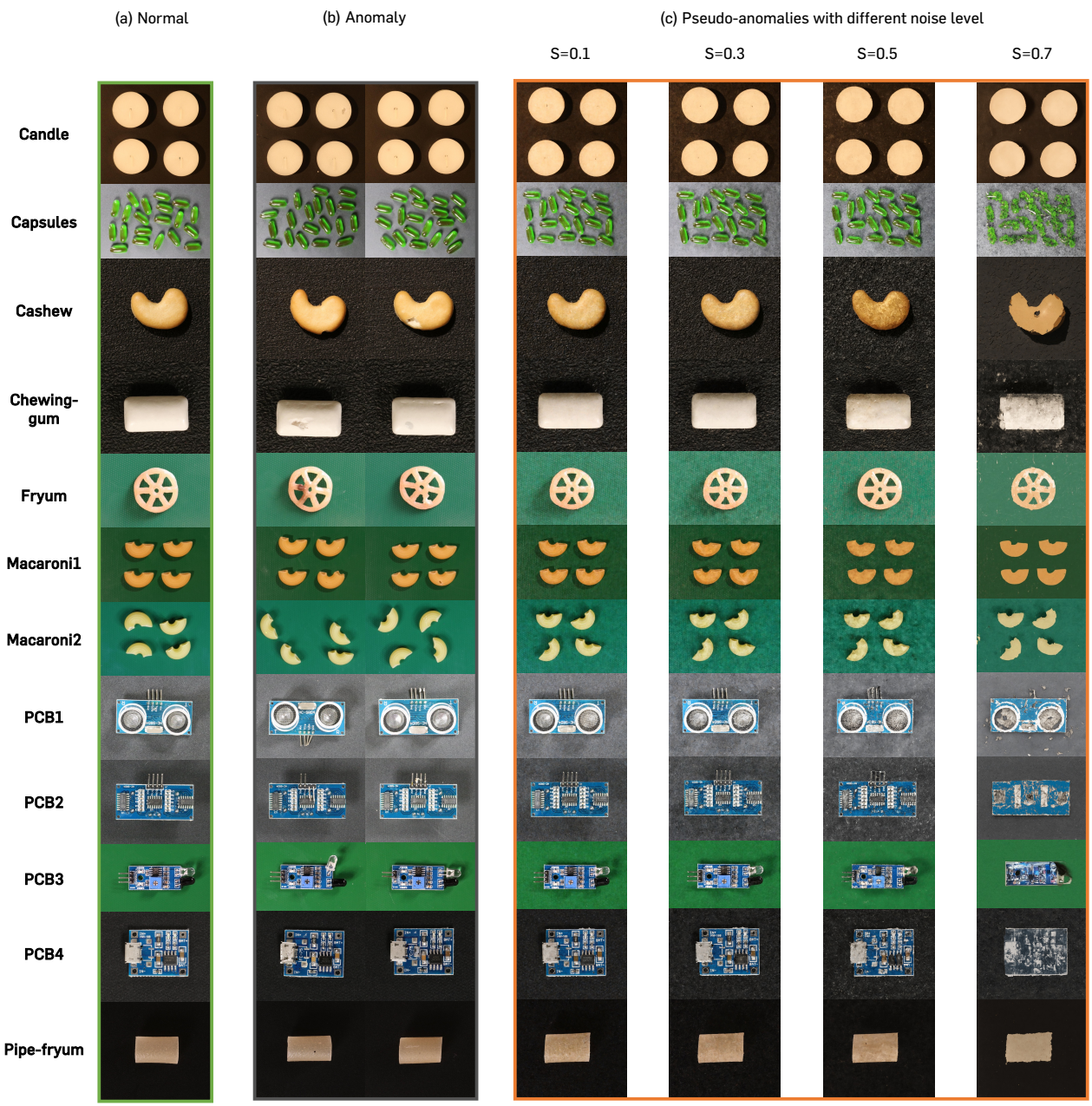

Figure 2. Visualization of (a) normal, (b) anomaly and (c) pseudo-anomalies synthesized via pre-trained text-to-image diffusion model with different noise level (S) in VisA. Pseudo-anomalies are generated with **simple prompt text** such as **"a photo with damage"**.

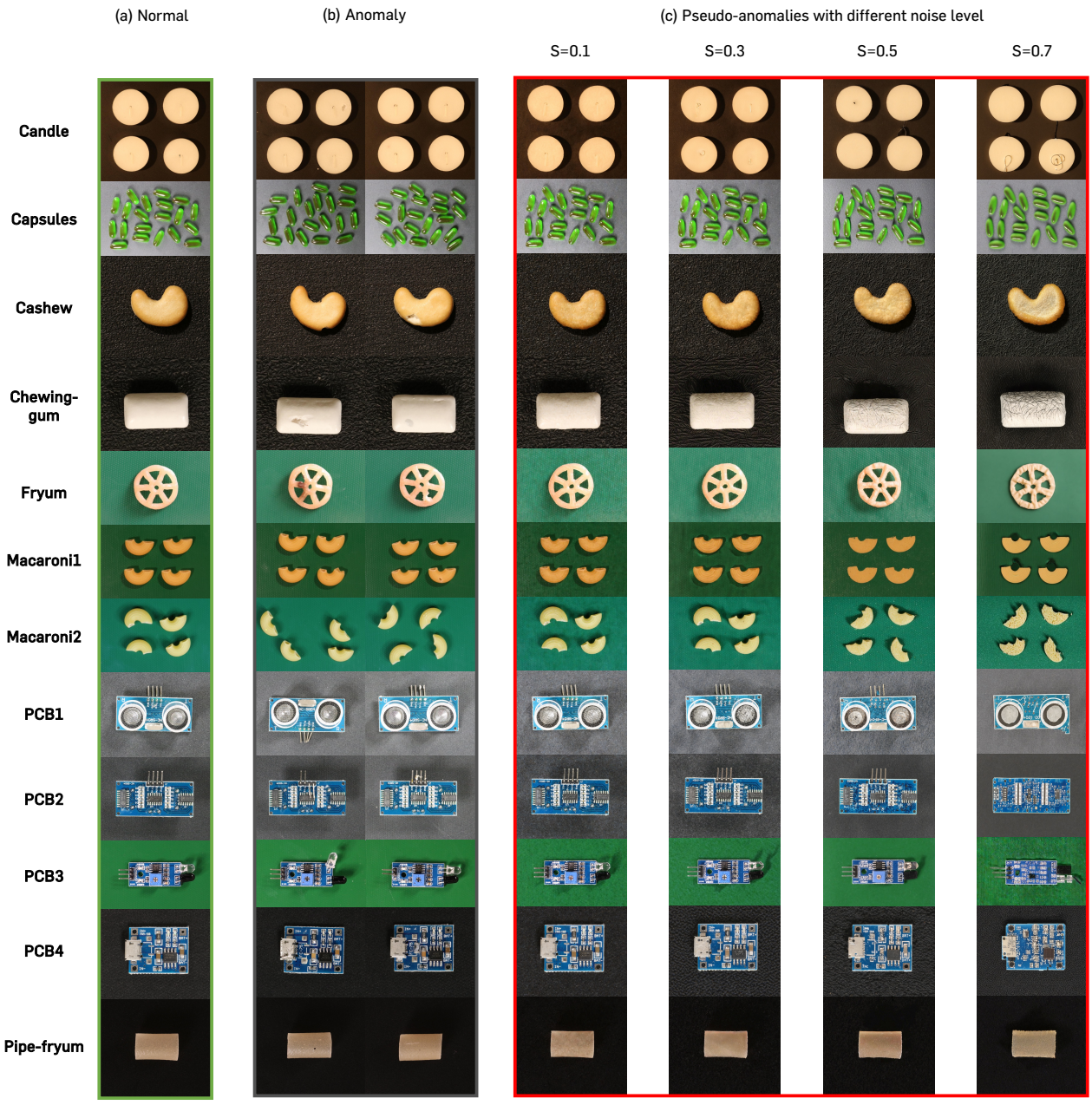

Figure 3. Visualization of (a) normal, (b) anomaly and (c) pseudo-anomalies synthesized via pre-trained text-to-image diffusion model with different noise level (S) in VisA. Pseudo-anomalies are generated with **prompts incorporating $\mathbf{c}_a^*$**, such as **"a photo of a $\mathbf{c}_a^*$ with damage"**.

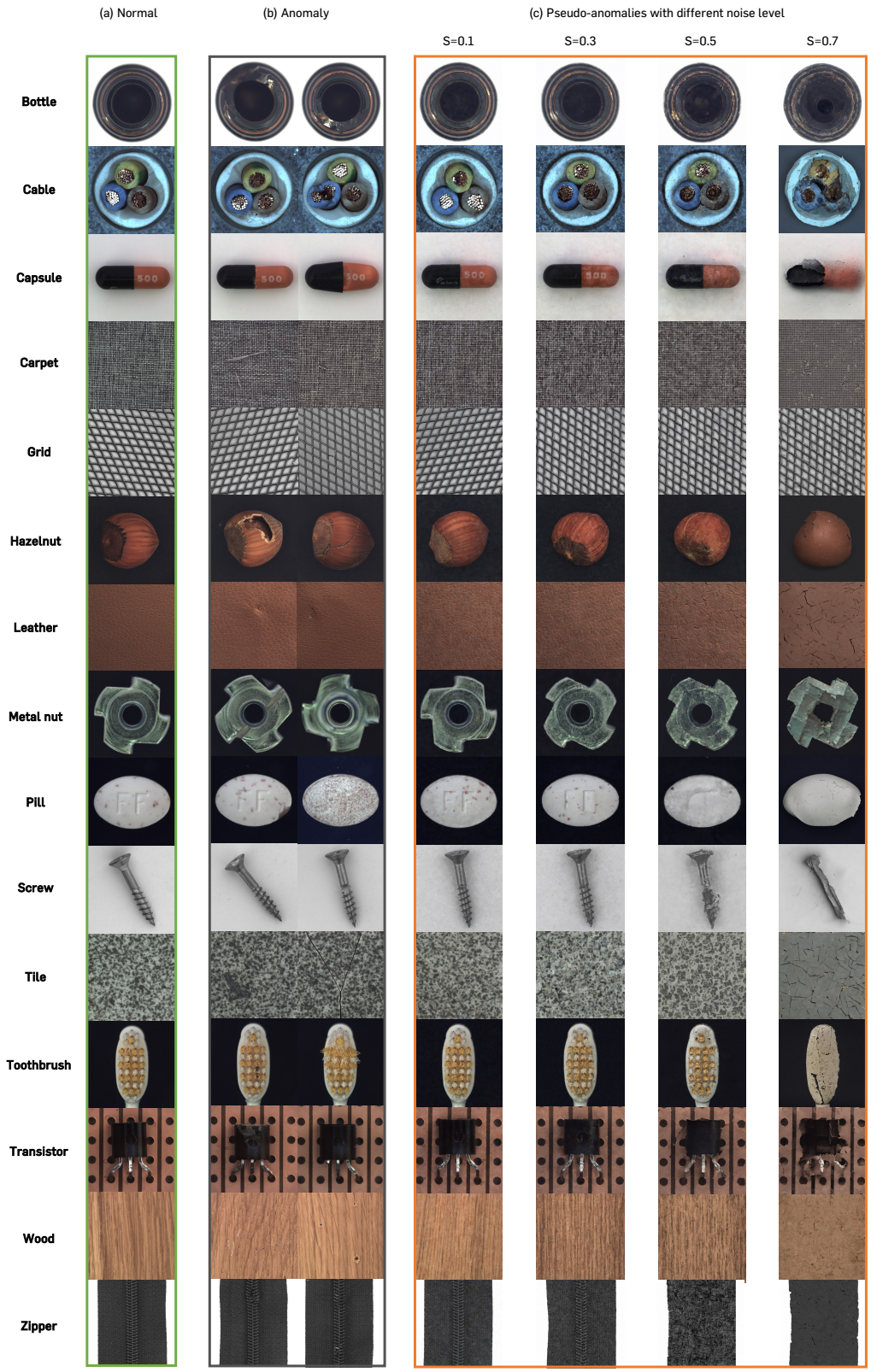

Figure 4. Visualization of (a) normal, (b) anomaly and (c) pseudo-anomalies synthesized via pre-trained text-to-image diffusion model with different noise level (S) in MVTec-AD. Pseudo-anomalies are generated with **simple prompt text** such as **"a photo with damage"**.

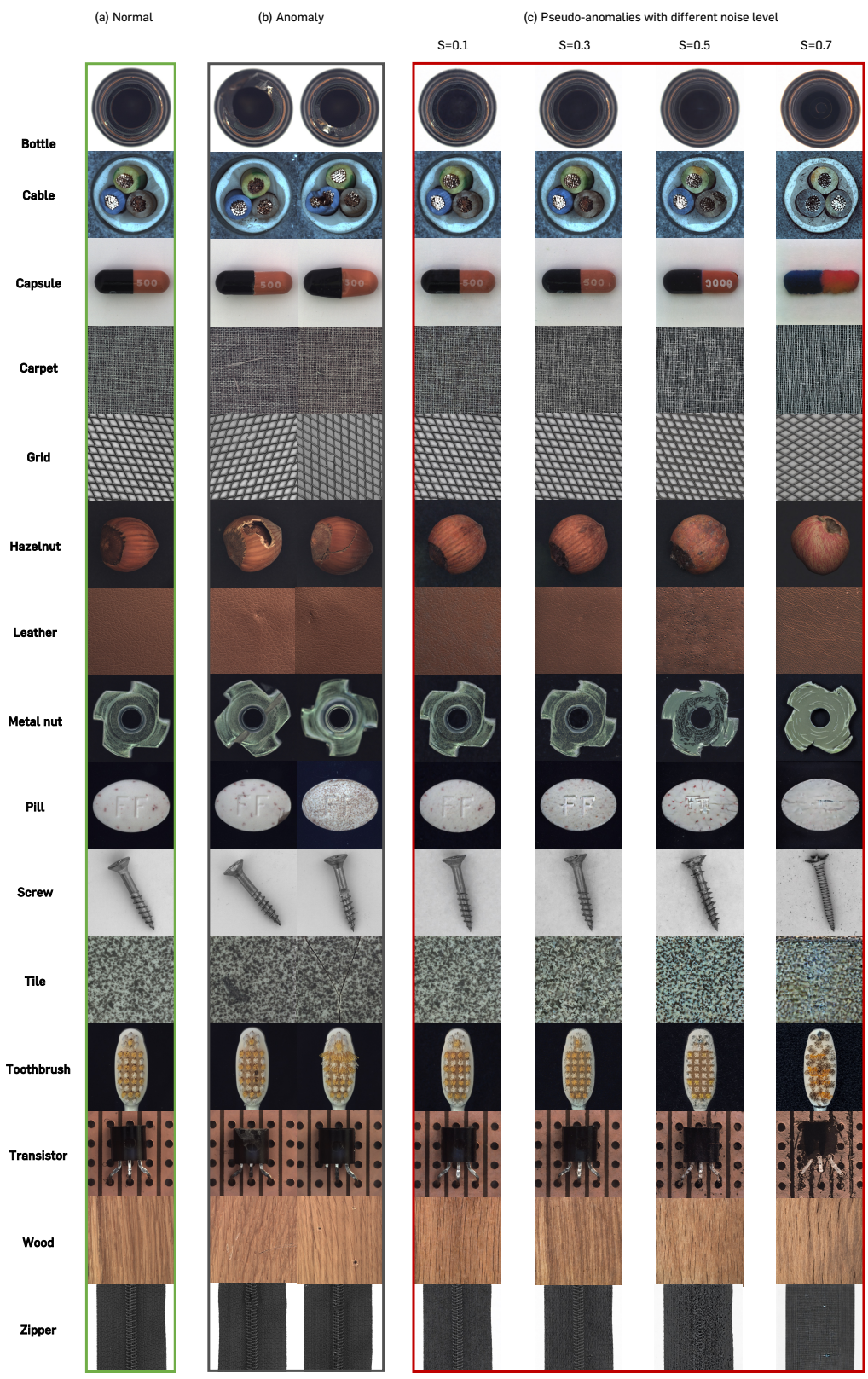

Figure 5. Visualization of (a) normal, (b) anomaly and (c) pseudo-anomalies synthesized via pre-trained text-to-image diffusion model with different noise level (S) in MVTec-AD. Pseudo-anomalies are generated with **prompts incorporating** $\mathbf{c}_a^*$, such as **"a photo of a** $\mathbf{c}_a^*$ **with damage"**.