A Additional experiment details

A.1 Hyperparameters

The hyperparameters for the self-supervised tasks used in our experiments are given in Tab. 4. $n_{\max}$ is the maximum number of patches added to each image. $h_{\min}, h_{\max}, w_{\min}, w_{\max} \in [0, 1]$ are the bounds on the patch dimensions relative to the image size. $b \in [0, 225]$ is the background brightness. All pixels with absolute brightness distance less than $t_{\text{brightness}}$ to the background brightness are assigned to the background. $t_{\text{object}}, t_{\text{overlap}} \in [0, 1]$ are used for the object and overlap conditions for the patches (see Section 3). The patch resize-scale is clipped to the range $[s_{\min}, s_{\max}]$ in addition to the conditions given in Section 3. y_0 and k give the midpoint and steepness of the logistic function used for creating the labels for NSA (logistic).

These parameters encode our assumptions about the unknown real out-distribution (see Sec. 3). Thus they were not tuned in a data-driven way as no validation set containing all possible types of real anomalies was available. These assumptions were chosen based on visual inspection of the input images and self-supervised examples. *E.g.*, for objects that have larger width than height, $w_{\max}$ is higher than $h_{\max}$ and vice versa; for classes with high perceived natural variation y_0 should be larger and k smaller.

Table 4: Hyperparameters for the self-supervised tasks.

				patch size		background constraints				scale	logistic	
			$n_{\max}$	$h_{\min}, h_{\max}$	$w_{\min}, w_{\max}$	b	$t_{\text{brightness}}$	t_{object}	t_{overlap}	$s_{\min}, s_{\max}$	y_0	k
MVTec AD	object	bottle	3	0.06, 0.80	0.06, 0.80	200	60	0.70	0.25	0.7, 1.3	24	1/12
		cable	3	0.10, 0.80	0.10, 0.80	N/A	N/A	N/A	N/A	0.7, 1.3	24	1/12
		capsule	3	0.06, 0.30	0.06, 0.80	200	60	0.70	0.25	0.7, 1.3	4	1/2
		hazelnut	3	0.06, 0.70	0.06, 0.70	20	20	0.70	0.25	0.7, 1.3	24	1/12
		metal nut	3	0.06, 0.80	0.06, 0.80	20	20	0.50	0.25	0.7, 1.3	7	1/3
		pill	3	0.06, 0.40	0.06, 0.80	20	20	0.70	0.25	0.7, 1.3	7	1/3
		screw	4	0.06, 0.24	0.06, 0.24	200	60	0.50	0.25	0.7, 1.3	3	1
		toothbrush	3	0.06, 0.80	0.06, 0.40	20	20	0.25	0.25	0.7, 1.3	15	1/6
		transistor	3	0.06, 0.80	0.06, 0.80	N/A	N/A	N/A	N/A	0.7, 1.3	15	1/6
		zipper	4	0.06, 0.80	0.06, 0.40	200	60	0.70	0.25	0.7, 1.3	15	1/6
	texture	carpet	4	0.06, 0.80	0.06, 0.80	N/A	N/A	N/A	N/A	0.5, 2.0	7	1/3
		grid	4	0.06, 0.80	0.06, 0.80	N/A	N/A	N/A	N/A	0.5, 2.0	7	1/3
		leather	4	0.06, 0.80	0.06, 0.80	N/A	N/A	N/A	N/A	0.5, 2.0	7	1/3
		tile	4	0.06, 0.80	0.06, 0.80	N/A	N/A	N/A	N/A	0.5, 2.0	7	1/3
		wood	4	0.06, 0.80	0.06, 0.80	N/A	N/A	N/A	N/A	0.5, 2.0	15	1/6
rCXR		male	3	0.06, 0.80	0.06, 0.80	0	20	0.70	0.70	0.7, 1.3	4	1/2
		female	3	0.06, 0.80	0.06, 0.80	0	20	0.70	0.70	0.7, 1.3	4	1/2

For FPI (Poisson), we used mixed gradients for seamless cloning. For NSA, we use mixed gradients for all rCXR data and for MVTec AD texture classes. For MVTec AD object classes, we find that OpenCV's [3] seamless cloning method causes artifacts when there are sharp contrast changes (*e.g.*, at the boundary from the object to the background) near the edges of the patch boundary more frequently when using mixed gradients than source gradients. Thus, we only use source gradients for these classes for NSA.

A.2 Comparison of self-supervised tasks

Table 5: Comparison of CutPaste [14], FPI [28], PII [29], and NSA self-supervised tasks.

	CutPaste [14]	FPI [28]	PII[29]	NSA (binary)	NSA (continuous)	NSA (logistic)
Different source and destination images?	✗	✓	✓	✓	✓	✓
Resize patch before blending?	✗	✗	✗	✓	✓	✓
Different source and destination patch locations?	✓	✗	✗	✓	✓	✓
Blending mode	copy-paste	linear interpolation	seamless cloning	seamless cloning	seamless cloning	seamless cloning
Label type	binary	bounded continuous (interpolation factor)	bounded continuous (interpolation factor)	binary	continuous (difference-based)	bounded continuous (difference-based)
Loss	BCE-loss	BCE-loss	BCE-loss	BCE-loss	MSE-loss	BCE-loss

Table 6: Comparison of the original patch-selection procedures for CutPaste [14], FPI [28], PII [29], and our method. We use our patch-selection procedure for our re-implementations of CutPaste, FPI, and PII. See Sec. 3 for more details on our method.

	CutPaste [14]	FPI [28] and PII [29]	Ours
Patch size	area ratio between patch and image sampled from $(0.02, 0.15)$	width and height relative to image dimensions sampled from $U(0.1, 0.4)$	width and height relative to image dimensions sampled from truncated $\text{Gamma}(2, 0.1)$
Patch aspect ratio	sampled from $(0.3, 1) \cup (1, 3.3)$	square, except when truncated by the image boundary	any ratio resulting from the above
Location restrictions	entire patch must appear in the full image	patch center must lie within the core 80% of the image dimensions	patch must contain part of the object and object portions at source and destination must overlap

B Additional results

B.1 Additional ablation studies

To validate our design choices we conducted several ablation studies beyond comparing different label definitions and comparing NSA to simpler baseline synthetic anomalies (see Sec. 4.2). Additional variants of NSA (logistic) considered were:

A Do not use foreground constraints for any classes. Note that in the original, foreground constraints were not applied to cable, transistor, and textures so these experiments do not need to be duplicated.

B Only use a single patch per training example instead of a random number of patches.

C Generate patch shapes as for CutPaste [14]:

1. sample the area ratio between the patch and the full image from $(0.02, 0.15)$,
2. determine the aspect ratio by sampling from $(0.3, 1) \cup (1, 3.3)$,
3. sample location such that patch is contained entirely within the image.

(Use single patch, uniform distributions, no foreground constraints, no resizing.)

D Mask the patches with a union of 5 random ellipses to achieve non-rectangular patch shapes.

For these experiments we report image-level and pixel-level AUROC % (Tab. 7). The results back-up our design choices as the final version outperforms all three variants. Specifically, the experiments show that

A using foreground constraints is most important for classes where the images contain a lot of background due to the shape of the objects (*e.g.*, screw and capsule),

B using a random number of patches performs slightly better than using a single patch,

C our patch-selection procedure leads to much better overall performance of NSA than the patch selection procedure described in [14], and

D beyond the diverse sizes and aspect ratios the shape of the patches is not important. This could be due to the fact that because of Poisson blending rectangular patches due not necessarily create rectangular anomalies, so NSA with rectangular patches already creates various non-rectangular anomalies.

Table 7: Image-level and pixel-level AUROC % for MVTec AD and standard error across five different random seeds for NSA (logistic) variants.

		Image-level AUROC %					Pixel-level AUROC %				
	NSA (logistic) variants	final	A	B	C	D	final	A	B	C	D
object	bottle	97.7 ± 0.3	97.5 ± 0.5	97.1 ± 0.4	97.1 ± 0.7	98.4 ± 0.2	98.3 ± 0.1	98.4 ± 0.1	97.9 ± 0.1	97.4 ± 0.2	98.9 ± 0.0
	cable	94.5 ± 1.0	–	96.1 ± 1.0	92.4 ± 2.0	91.8 ± 1.4	96.0 ± 1.4	–	94.7 ± 2.7	96.7 ± 0.5	86.8 ± 2.4
	capsule	95.2 ± 1.7	91.7 ± 1.9	89.7 ± 1.3	84.6 ± 0.8	95.6 ± 0.7	97.6 ± 0.9	96.4 ± 0.4	95.2 ± 1.4	92.7 ± 0.9	97.1 ± 0.2
	hazelnut	94.7 ± 1.1	93.1 ± 0.8	92.2 ± 2.7	85.2 ± 2.5	94.1 ± 2.0	97.6 ± 0.6	97.5 ± 0.7	93.6 ± 0.9	94.5 ± 1.2	97.5 ± 0.4
	metal nut	98.7 ± 0.7	99.0 ± 0.6	97.7 ± 1.0	94.4 ± 1.0	99.3 ± 0.3	98.4 ± 0.2	98.5 ± 0.1	96.5 ± 0.9	97.0 ± 0.3	98.2 ± 0.4
	pill	99.2 ± 0.6	99.0 ± 0.2	97.8 ± 0.5	94.5 ± 1.6	96.9 ± 1.0	98.5 ± 0.3	97.5 ± 0.2	90.5 ± 4.5	92.8 ± 2.2	97.1 ± 1.0
	screw	90.2 ± 1.4	77.8 ± 3.3	85.3 ± 3.4	56.3 ± 1.8	90.3 ± 1.0	96.5 ± 0.1	92.9 ± 0.6	95.6 ± 0.8	82.6 ± 1.6	96.2 ± 0.2
	toothbrush	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.7 ± 0.2	100.0 ± 0.0	94.9 ± 0.7	93.8 ± 1.1	91.7 ± 2.8	94.4 ± 0.9	95.3 ± 0.2
	transistor	95.1 ± 0.2	–	93.7 ± 1.5	91.2 ± 1.7	93.2 ± 0.8	88.0 ± 1.8	–	83.8 ± 0.9	83.1 ± 2.2	86.0 ± 1.1
	zipper	99.8 ± 0.1	100.0 ± 0.0	99.8 ± 0.3	98.9 ± 1.1	99.9 ± 0.1	94.2 ± 0.3	94.1 ± 0.2	94.0 ± 0.3	94.0 ± 0.3	94.9 ± 0.1
	average	96.5 ± 0.3	–	94.9 ± 0.6	89.4 ± 0.4	96.0 ± 0.2	96.0 ± 0.4	–	93.3 ± 0.9	92.3 ± 0.5	94.8 ± 0.3
texture	carpet	95.6 ± 0.6	–	88.7 ± 2.8	87.4 ± 5.7	97.2 ± 1.3	95.5 ± 2.3	–	95.8 ± 5.0	88.3 ± 4.5	98.0 ± 0.7
	grid	99.9 ± 0.1	–	100.0 ± 0.0	98.6 ± 0.8	100.0 ± 0.0	99.2 ± 0.1	–	98.4 ± 0.7	92.5 ± 2.0	99.4 ± 0.0
	leather	99.9 ± 0.1	–	99.9 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	99.5 ± 0.1	–	99.3 ± 0.5	99.5 ± 0.1	99.7 ± 0.0
	tile	100.0 ± 0.0	–	100.0 ± 0.0	100.0 ± 0.0	99.9 ± 0.1	99.3 ± 0.0	–	98.5 ± 0.4	95.6 ± 2.1	98.2 ± 0.4
	wood	97.5 ± 1.5	–	91.4 ± 4.3	91.4 ± 2.5	98.0 ± 0.3	90.7 ± 1.9	–	86.5 ± 3.9	?856 ± 2.0	92.4 ± 0.6
	average	98.6 ± 0.3	–	96.0 ± 0.7	95.5 ± 1.5	99.0 ± 0.3	96.8 ± 0.7	–	95.7 ± 1.6	92.3 ± 1.1	97.5 ± 0.1
	overall average	97.2 ± 0.3	–	95.3 ± 0.5	91.4 ± 0.5	97.0 ± 0.2	96.3 ± 0.4	–	94.1 ± 0.8	92.3 ± 0.5	95.7 ± 0.2

B.2 Per-region overlap

The AU-PRO$_{0.3}$ metric is defined as the area under the per-region overlap (PRO) curve for false positive rates up to 30 % [2]. To calculate PRO, we decompose the ground-truth label maps into M connected components such that $C_{j,k}$ gives the set of anomalous pixels in a connected component k of label map j. Let P_j denote the predicted anomalous pixels when using a threshold t. [2] defines PRO as:

$$\text{PRO} = \frac{1}{M} \sum_j \sum_k \frac{|P_j \cap C_{j,k}|}{|C_{j,k}|} \tag{17}$$

Unlike pixel-level AUROC, AU-PRO assigns equal weight to small and large anomalies. This is desirable for practical applications where precise localization of small anomalies is at least as important as localization of large anomalies.

In Tab. 8 we report AU-PRO$_{0.3}$ scores for our models from Tab. 2 as well as the scores for PaDiM [6] for reference. Note that unlike our method, PaDiM relies on ImageNet pretraining, so this is not a fair comparison. The authors of CutPaste [14] and DRAEM [36] did not report AU-PRO for their method but we hope that future methods that learn from scratch can compare their localization performance to our AU-PRO scores.

Table 8: AU-PRO$_{0.3}$ % for MVTec AD defect localization and standard error across five different random seeds. Scores are calculated for 256×256 resampled image and mask. Best scores between PaDiM-WR50-Rd550 [6] and NSA within standard error are bold-faced. Note that PaDiM uses pretrained ImageNet features.

		SOTA	Our Experiments					
		PaDiM [6]	CutPaste (end-to-end)	FPI	PII	NSA (binary)	NSA (continuous)	NSA (logistic)
object	bottle	**94.8**	91.2	66.0	79.0	93.0 ± 0.9	89.9 ± 1.1	92.9 ± 0.3
	cable	88.8	59.8	51.9	55.7	**87.6** ± 3.4	85.4 ± 2.1	**89.9** ± 1.0
	capsule	**93.5**	83.5	79.9	67.6	91.8 ± 0.8	79.9 ± 9.0	**91.4** ± 2.2
	hazelnut	92.6	81.3	71.4	90.9	**93.6** ± 0.4	**93.1** ± 1.3	**93.6** ± 0.9
	metal nut	85.6	54.4	72.2	91.5	**94.9** ± 0.2	90.8 ± 1.1	**94.6** ± 0.6
	pill	92.7	83.1	50.4	65.2	93.7 ± 0.9	**92.5** ± 3.5	**96.0** ± 0.5
	screw	**94.4**	72.6	69.8	78.4	90.6 ± 1.3	80.6 ± 10.3	90.1 ± 0.3
	toothbrush	**93.1**	88.1	60.3	66.8	91.2 ± 0.6	89.0 ± 1.8	90.7 ± 1.0
	transistor	**84.5**	68.5	55.4	57.4	72.6 ± 4.4	63.3 ± 1.2	75.3 ± 2.4
	zipper	**95.9**	84.9	81.2	86.6	88.9 ± 0.5	83.6 ± 3.3	89.2 ± 0.3
	average	**91.6**	76.7	65.8	73.9	89.8 ± 0.8	84.8 ± 2.8	90.4 ± 0.5
texture	carpet	**96.2**	50.4	21.6	93.5	84.0 ± 11.8	71.1 ± 8.2	85.0 ± 6.2
	grid	94.6	91.5	86.0	95.9	**96.5** ± 0.1	94.2 ± 0.8	**96.8** ± 0.4
	leather	97.8	83.7	84.1	98.1	**98.9** ± 0.1	**98.6** ± 0.4	**98.7** ± 0.1
	tile	86.0	54.4	42.0	83.2	**93.9** ± 0.9	90.3 ± 2.5	**95.3** ± 0.5
	wood	**91.1**	64.0	41.7	81.7	**89.2** ± 2.4	**86.1** ± 5.7	85.3 ± 3.7
	average	**93.2**	68.8	55.1	90.5	**92.5** ± 2.0	88.1 ± 1.3	**92.2** ± 1.4
overall average		**92.1**	74.1	62.3	79.4	90.7 ± 0.4	85.9 ± 2.1	91.0 ± 0.6