Supplementary Materials

In this supplementary material, we provide additional details that are necessary for reproducing our results as well as additional results that we could not include in the paper due to the page limit.

A. Additional Implementation Details

Table 1 and Table 2 shows a detailed architecture of our TrustMAE and the discriminator used to train it.

As a pre-processing step, we resize the images to 256×256 and scaled the values to be within -1 to 1. During training, we augment our training data using horizontal flips, vertical flips, and random rotations, except for some products, such as cable, metal nut, and transistors, that have fixed orientations, and will be considered defective if it is transformed.

To encourage sharper reconstructions, our model optimizes for a weighted sum of several loss terms, as shown in Eq. 9.

$$\begin{aligned} L_{total} = \lambda_{rec}L_{rec} + \lambda_{sm}L_{sm} + \lambda_{vgg}L_{vgg} + \lambda_{GAN}L_{GAN} \\ + \lambda_{feat}L_{feat} + \lambda_{margin}L_{margin} + \lambda_{trust}L_{trust} \end{aligned} \tag{9}$$

We set the loss weights as follows $\lambda_{rec} = 10, \lambda_{sm} = 10, \lambda_{vgg} = 10, \lambda_{GAN} = 1, \lambda_{feat} = 10$. Each of the loss terms are detailed below:

- Reconstruction loss encourages the output to be close to the input image. We implement it as the mean absolute error between the input and output, as shown below.

$$L_{rec} = \frac{1}{HWC} \sum_{h,w,c} |x - \hat{x}| \tag{10}$$

- SSIM loss matches the luminance, contrast, and structure between two images by matching the mean, standard deviation, and covariance of their patches, as shown in Eq. 11

$$L_{sm}(p, q) = \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)}, \tag{11}$$

  where μ_p, μ_q are means of patch p and q, σ_p, σ_q are the standard deviations of the patch p and q, σ_{pq} is the covariance between the two patches, and c_1, c_2, c_3 are constants that prevents numerical issues caused by the divisions. We use a patch size of 11×11 and set the constants to their recommended defaults $c_1 = 0.01^2, c_2 = 0.03^2, c_3 = c_2/2$.

- VGG feature loss is a form of perceptual loss that encourages feature representations to be similar rather than exact pixel matching [18]. It is defined in Eq. 12

$$L_{vgg} = \sum_{l=1}^{L} \lambda^{(l)} \|\psi^{(l)}(x) - \psi^{(l)}(G(x))\|_1, \tag{12}$$

  where $\psi^{(l)}(x)$ denotes the output at the l-th layer of a pre-trained VGG-19 network given an input x, $G(x)$ denotes the reconstructed output of the auto-encoder, and $\lambda^{(l)}$ are hyperparameters that adjusts the relative importance of each layer l. The lower layers preserves low level features such as edges, while higher layers preserves high level features such as texture and spatial structure. We used the outputs of conv1_2, conv2_2, conv3_4, conv4_4, and conv5_4 layers with layer weights $\lambda^{(1)} = 1/32, \lambda^{(2)} = 1/16, \lambda^{(3)} = 1/8, \lambda^{(4)} = 1/4, \lambda^{(5)} = 1$.

- GAN loss [13] introduces a separate discriminator network D_{disc} that aims to distinguish whether an input image looks real or fake. Our memory auto-encoder G is jointly optimized with the discriminator D_{disc} in an adversarial fashion wherein the discriminator classifies the synthesized reconstructions as fake, while the memory auto-encoder tries to produce images that fools the discriminator into classifying it as real. We adopt the hinge loss [45] formulation as defined in Eq. 13 for the discriminator and Eq. 14 for the auto-encoder G.

$$\begin{aligned} L_{GAN}^{(Disc)} = &- \mathbb{E}[\min(0, -1 + D_{disc}(x))] \\ &- \mathbb{E}[\min(0, -1 - D_{disc}(G(x)))] \end{aligned} \tag{13}$$

$$L_{GAN}^{(G)} = -\mathbb{E}[D_{disc}(G(x))] \tag{14}$$

- GAN feature loss [42, 43] is similar to the VGG feature loss but uses the intermediate layers of the discriminator instead of a VGG network, as shown in Eq. 15, where $D_{disc}^{(l)}$ denotes the intermediate output of the discriminator at layer l. Wang et al. [42] showed that matching the statistics of real images through the discriminator features at multiple scales help stabilize the adversarial training. Another advantage is that the discriminator is trained on the dataset that we care about, which means that the features we are matching are more appropriate for the dataset we are using [43], as opposed to using features extracted by the VGG network that were optimized for ImageNet.

$$L_{feat} = \sum_{l=1}^{L} \| D_{disc}^{(l)}(x) - D_{disc}^{(l)}(G(x)) \|_1 \qquad (15)$$

B. Baselines Implementation Details

We adopted most of the baseline performance values from Bergmann et al. [2], Dehaene et al. [6], Huang et al. [16], and Liu et al. [22].

For the Memory auto-encoder (MemAE) [12] baseline (which our model was built on top of), we trained the model patterned after their publicly available code[1]. We adapted their network architecture and changed the 3D convolutions to 2D. We trained the network on images sized 256×256 for 200 epochs using the Adam optimizer with a learning rate of $1e^{-4}$. The memory size was set to 128 with a latent dimension of 256.

The baselines AE-SSIM [3] and AE-L2 [2] follows the setting of Dehaene et al. [6]. They used the network architecture of Bergmann et al. [3] with a latent dimension of 100 and trained for 300 epochs with a learning rate of $1e^{-4}$. The images were resized to 128×128 except for textures, which required larger resolutions due to high frequency content, thus, the texture images were first resized to 512×512, before cropping random patches of 128×128 for training.

C. Additional Results

Table 3 shows the results of our ablation study on each of the classes in MVTec dataset, while Table 4 shows the results of our comparison with the baselines on each of the classes in the MVTec dataset. Overall, our method with both trust region memory updates and spatial perceptual distance achieves competitive performance across a large range of noise levels.

We also show additional qualitative results in Figures 1, 2, and 3. We can observe that the large values in the error map computed with our spatial perceptual distance matches well with the ground truth defect segmentation.

[1] https://github.com/donggong1/memae-anomaly-detection

Table 1: Network architecture of our TrustMAE. The abbreviations are as follows: N denotes number of filters, K denotes kernel size, S denotes stride, P denotes padding, and CBN denotes conditional batch normalization.

Input $\rightarrow$ Output Shape	Layer Information
Encoder	
$(h, w, 3) \rightarrow (h, w, 32)$	Conv-(N32, K7 $\times$ 7, S1, P3), CBN, ReLU
$(h, w, 32) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	Conv-(N64, K3 $\times$ 3, S2, P1), CBN, ReLU
$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	Conv-(N128, K3 $\times$ 3, S2, P1), CBN, ReLU
$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	Conv-(N256, K3 $\times$ 3, S2, P1), CBN, ReLU
$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	Conv-(N512, K3 $\times$ 3, S2, P1), CBN, ReLU
$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Conv-(N512, K3 $\times$ 3, S2, P1), CBN, ReLU
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Residual Block: Conv-(N512, K3 $\times$ 3, S1, P1), CBN, ReLU
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Residual Block: Conv-(N512, K3 $\times$ 3, S1, P1), CBN, ReLU
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Residual Block: Conv-(N512, K3 $\times$ 3, S1, P1), CBN, ReLU
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Memory Module $\in \mathbb{R}^{512 \times M}$
Decoder	
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Residual Block: Conv-(N512, K3 $\times$ 3, S1, P1), CBN, ReLU
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Residual Block: Conv-(N512, K3 $\times$ 3, S1, P1), CBN, ReLU
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Residual Block: Conv-(N512, K3 $\times$ 3, S1, P1), CBN, ReLU
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	ConvTrans-(N512, K4 $\times$ 4, S2, P1), CBN, ReLU
$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	ConvTrans-(N256, K4 $\times$ 4, S2, P1), CBN, ReLU
$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	ConvTrans-(N128, K4 $\times$ 4, S2, P1), CBN, ReLU
$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	ConvTrans-(N64, K4 $\times$ 4, S2, P1), CBN, ReLU
$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (h, w, 32)$	ConvTrans-(N32, K4 $\times$ 4, S2, P1), IN, ReLU
$(h, w, 32) \rightarrow (h, w, 3)$	Conv-(N3, K7 $\times$ 7, S1, P3), Tanh

Table 2: Network architecture of our discriminator.

Input $\rightarrow$ Output Shape	Layer Information
$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 32)$	Conv-(N32, K4 $\times$ 4, S2, P1), LeakyReLU-(0.2)
$(\frac{h}{2}, \frac{w}{2}, 32) \rightarrow (\frac{h}{4}, \frac{w}{4}, 64)$	Conv-(N64, K4 $\times$ 4, S2, P1), LeakyReLU-(0.2)
$(\frac{h}{4}, \frac{w}{4}, 64) \rightarrow (\frac{h}{8}, \frac{w}{8}, 128)$	Conv-(N128, K4 $\times$ 4, S2, P1), LeakyReLU-(0.2)
$(\frac{h}{8}, \frac{w}{8}, 128) \rightarrow (\frac{h}{16}, \frac{w}{16}, 256)$	Conv-(N256, K4 $\times$ 4, S2, P1), LeakyReLU-(0.2)
$(\frac{h}{16}, \frac{w}{16}, 256) \rightarrow (\frac{h}{32}, \frac{w}{32}, 512)$	Conv-(N512, K4 $\times$ 4, S2, P1), LeakyReLU-(0.2)
$(\frac{h}{32}, \frac{w}{32}, 512) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1024)$	Conv-(N1024, K4 $\times$ 4, S2, P1), LeakyReLU-(0.2)
$(\frac{h}{64}, \frac{w}{64}, 1024) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	Conv-(N1, K3 $\times$ 3, S1, P1)

Table 3: Detailed ablation study showing the performance of the different components of our model on each of the products in the MVTec dataset.

Method	Noise %	mean AUC	bottle	cable	capsule	carpet	grid	hazelnut	leather	metal nut	pill	screw	tile	toothbrush	transistor	wood	zipper
ours w/o PD w/o TR	5%	82.61	88.28	71.17	75.16	80.82	96.83	91.04	95.57	64.96	77.29	75.13	83.73	95.56	65.25	94.23	84.17
ours w/o SP	5%	90.73	99.84	89.03	78.96	91.26	97.78	96.88	97.92	88.07	75.76	79.18	96.76	99.44	82.08	99.15	88.83
ours w/o PD	5%	82.03	96.72	70.58	73.37	80.32	91.27	93.96	94.14	59.28	82.93	82.41	76.11	94.44	53.92	94.74	86.22
ours w/o TR	5%	92.51	99.13	90.94	79.04	93.54	98.41	97.22	99.48	88.16	86.99	79.25	99.44	99.44	82.42	100	94.26
ours	5%	92.39	99.37	92.77	78.96	93.47	98.1	98.82	97.27	86.55	84.77	81.73	99.01	96.67	84	100	94.42
ours w/o PD w/o TR	10%	81.27	84.84	69.55	71.64	80.78	95.08	90.69	93.1	59.56	78.56	76.77	81.54	94.44	63.92	92.87	85.66
ours w/o SP	10%	90.02	98.59	88.21	77.8	92.17	96.83	97.36	95.57	87.59	76.82	77.49	95.98	98.33	79.83	99.47	88.32
ours w/o PD	10%	81.14	94.53	72.19	76.55	79.26	89.37	91.94	87.76	62.03	81.09	76.33	79	92.78	57.25	91.85	85.14
ours w/o TR	10%	91.8	99.06	89.84	79.95	91.95	98.58	97.99	98.05	88.64	85.77	77.13	97.46	97.22	81.08	99.83	94.47
ours	10%	91.9	99.38	91.2	77.02	93.77	98.41	99.03	97.59	87.22	83.83	78.93	98.87	97.22	81.33	99.83	94.83
ours w/o PD w/o TR	15%	79.83	85.31	71.86	70.11	75.46	93.65	90.76	92.51	59.09	71.03	74.69	79.35	92.22	63.83	93.72	83.81
ours w/o SP	15%	89.14	97.62	87.09	76.4	90.96	95.24	96.53	95.01	87.22	75.13	76.41	95.6	97.22	80.92	98.64	87.14
ours w/o PD	15%	80.78	97.03	69.6	73.6	75.08	84.29	91.18	90.27	63.45	77.34	81.33	79.15	95.56	58.75	91.74	83.35
ours w/o TR	15%	90.38	98.89	87.12	79.1	92.10	96.67	96.67	96.88	86.27	76.98	77.45	97.32	98.33	78.08	99.15	94.72
ours	15%	91.05	99.37	89.18	73.76	91.57	98.1	99.24	97.79	87.03	82.46	75.13	97.11	98.89	82.17	100	94.01
ours w/o PD w/o TR	20%	78.8	85.94	67.02	67.86	74.92	94.15	88.12	91.67	53.88	69.92	72.45	82.45	92.78	64.83	92.02	84.02
ours w/o SP	20%	88.94	98.28	87.45	76.75	88.45	96.03	96.32	96.55	85.13	77.87	75.05	94.93	96.67	79.17	99.32	86.12
ours w/o PD	20%	79.72	93.75	66.87	73.37	73.94	85.08	92.15	89.45	60.13	79.77	80.25	79.07	94.44	53.75	90.15	83.66
ours w/o TR	20%	88.93	97.19	84.85	77.93	91.79	94.92	96.81	96.55	80.4	78.19	72.33	94.22	98.89	77.25	99.32	93.29
ours	20%	90.32	99.69	89.8	74.15	92.17	98.25	98.54	98.57	84.66	82.09	75.85	92.67	97.22	79.5	98.64	92.93
ours w/o PD w/o TR	30%	79.18	87.34	65.19	70.57	77.05	89.72	91.88	93.88	52.56	66.97	75.89	85.13	90.56	63.08	94.23	83.66
ours w/o SP	30%	87.01	96.41	86.76	75.18	89.51	92.38	95.56	92.06	81.25	75.18	74.69	95.28	91.11	73.75	99.15	86.89
ours w/o PD	30%	79.8	90.31	67.72	67.78	77.89	70.32	93.89	94.86	65.81	82.09	82.21	82.8	92.78	56.25	90.49	81.81
ours w/o TR	30%	87.1	97.03	83.2	74.91	88.68	89.21	96.04	94.66	78.41	75.08	71.41	93.23	96.11	77.42	98.47	92.67
ours	30%	89.89	97.66	89.1	75.08	91.49	95.56	97.92	96.81	83.52	81.61	76.37	93.02	97.78	78.33	100	94.06
ours w/o PD w/o TR	40%	76.96	83.81	66.47	70.42	72.72	83.71	89.93	89.71	52.65	53.64	71.57	81.61	95	61.58	95.25	86.27
ours w/o SP	40%	84.96	94.22	83.93	72.28	85.87	83.49	95.14	94.79	78.79	74.97	74.53	93.73	86.67	71.58	98.3	86.12
ours w/o PD	40%	77.82	91.88	69.59	72.13	64.74	70.16	93.06	90.49	59.94	79.72	78.49	85.41	91.11	56.17	81.32	83.04
ours w/o TR	40%	86.68	96.41	81.95	75.22	89.44	84.96	96.46	96.48	79.07	72.34	71.69	92.6	97.22	76.92	97.45	92.01
ours	40%	89.87	99.34	88.26	76.01	91.11	93.49	98.54	96.35	84.28	79.77	75.97	94.01	97.78	79.33	99.83	93.95

Table 4: Detailed performance on varying noise levels of our model compared with the baselines on each of the products in the MVTec dataset.

Method	Noise %	mean AUC	bottle	cable	capsule	carpet	grid	hazelnut	leather	metal nut	pill	screw	tile	toothbrush	transistor	wood	zipper
Ours	5%	92.39	99.37	92.77	78.96	93.47	98.1	98.82	97.27	86.55	84.77	81.73	99.01	96.67	84	100	94.42
MemAE	5%	79.68	86.93	49.63	78.42	75.76	96.98	95.62	94.27	44.31	77.33	81.95	77.68	95.37	68.58	96.49	75.85
AE-SSIM	5%	76.52	87.81	80.74	71.58	61.93	78.78	78.82	57.68	62.31	83.25	76.01	69.06	91.67	78.17	90.35	79.71
AE-MSE	5%	81.81	92.66	76.85	73.37	71.31	90.98	96.25	86.98	60.51	77.08	76.68	78.44	93.89	68.25	95.08	88.78
Ours	10%	91.9	99.38	91.2	77.02	93.77	98.41	99.03	97.59	87.22	83.83	78.93	98.87	97.22	81.33	99.83	94.83
MemAE	10%	77.26	87.34	49.66	72.77	71.73	94.28	93.8	90.45	43.65	74.08	74.12	73.29	95	64.33	95.98	78.38
AE-SSIM	10%	75.52	87.62	76.45	69.95	60.07	75.19	80.14	57.29	60.51	82.35	77.82	69.7	90.56	77.83	87.46	79.92
AE-MSE	10%	80.09	92.34	74.5	71.89	73.88	88.73	96.74	84.82	53.88	76.1	76.77	72.55	91.11	68.75	93.89	85.4
Ours	15%	91.05	99.37	89.18	73.76	91.57	98.1	99.24	97.79	87.03	82.46	75.13	97.11	98.89	82.17	100	94.01
MemAE	15%	77.89	90.78	47.54	77.1	72.44	95.39	91.06	97.74	36.74	74.97	77.97	73.17	93.33	63.03	96.32	80.77
AE-SSIM	15%	74.32	84.6	75.06	69.18	58.15	73.49	78.33	56.52	61.46	81.51	75.13	69.48	91.11	76.58	86.23	77.92
AE-MSE	15%	79.25	91.72	76.05	69.64	68.02	89.21	96.11	85.36	56.34	73.55	74.25	73.45	92.22	69.33	91.85	81.61
Ours	20%	90.32	99.69	89.8	74.15	92.17	98.25	98.54	98.57	84.66	82.09	75.85	92.67	97.22	79.5	98.64	92.93
MemAE	20%	76.05	88.33	48.12	76.06	74.82	93.01	93.77	91.3	33.21	73.04	74.13	70.61	93.89	55.11	96.15	79.15
AE-SSIM	20%	72.74	85.95	72.96	67.31	57.14	70.48	76.88	53.45	61.17	82.67	73.25	64.39	86.67	73.67	86.67	78.38
AE-MSE	20%	78.37	90.16	75.5	69.25	69.3	88.3	95.56	84.57	52.56	73.4	71.47	71.93	88.89	67.75	93.89	82.99
Ours	30%	89.89	97.66	89.1	75.08	91.49	95.56	97.92	96.81	83.52	81.61	76.37	93.02	97.78	78.33	100	94.06
MemAE	30%	74.06	85.6	48.86	76.45	63.8	91.26	91	94.29	35.92	68.12	74.22	58.87	90.93	59.31	95.81	76.4
AE-SSIM	30%	71.88	83.44	73.66	67.69	58.79	67.62	77.36	52.8	59.56	82.46	74.41	61.5	85.56	72.5	84.55	76.28
AE-MSE	30%	76.33	90.31	71.72	67.39	68.86	87.46	93.86	82.13	49.81	72.55	69.77	65.37	88.33	62.67	91.68	83.04
Ours	40%	89.87	99.34	88.26	76.01	91.11	93.49	98.54	96.35	84.28	79.77	75.97	94.01	97.78	79.33	99.83	93.95
MemAE	40%	74.62	81.51	47.35	77.07	68.57	88.73	90.37	91.62	40.56	65.1	78.56	61.36	90.37	62.14	96.49	79.53
AE-SSIM	40%	69.8	80.94	69.74	66.54	54.94	66.92	74.31	50.98	57.1	81.09	70.28	60.64	87.22	71.42	82.17	72.66
AE-MSE	40%	75.46	88.91	71.2	66.23	67.38	84.29	90.61	80.74	49.24	73.13	67.97	66.17	89.44	62.58	90.15	83.81

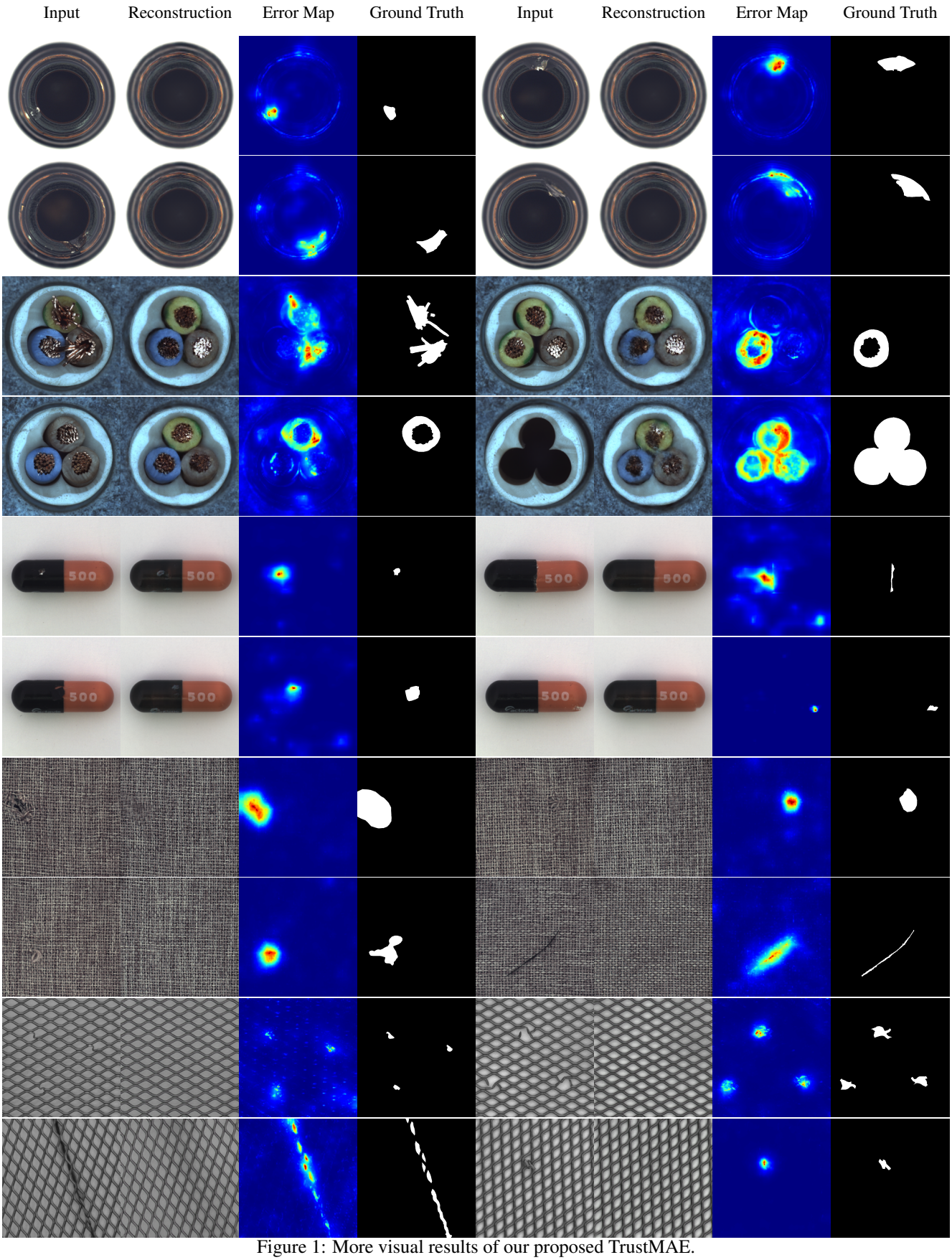

Figure 1: More visual results of our proposed TrustMAE.

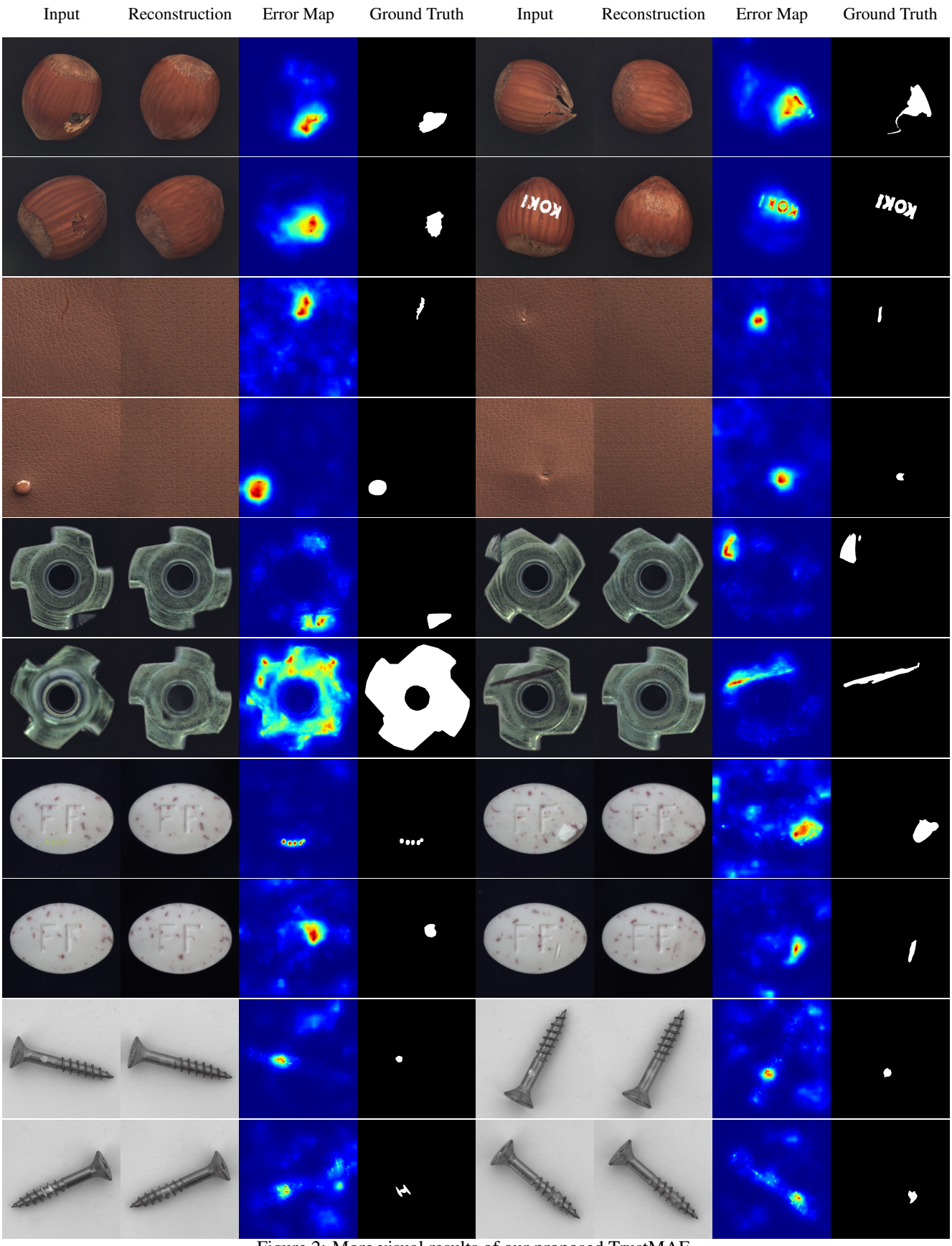

Figure 2: More visual results of our proposed TrustMAE.

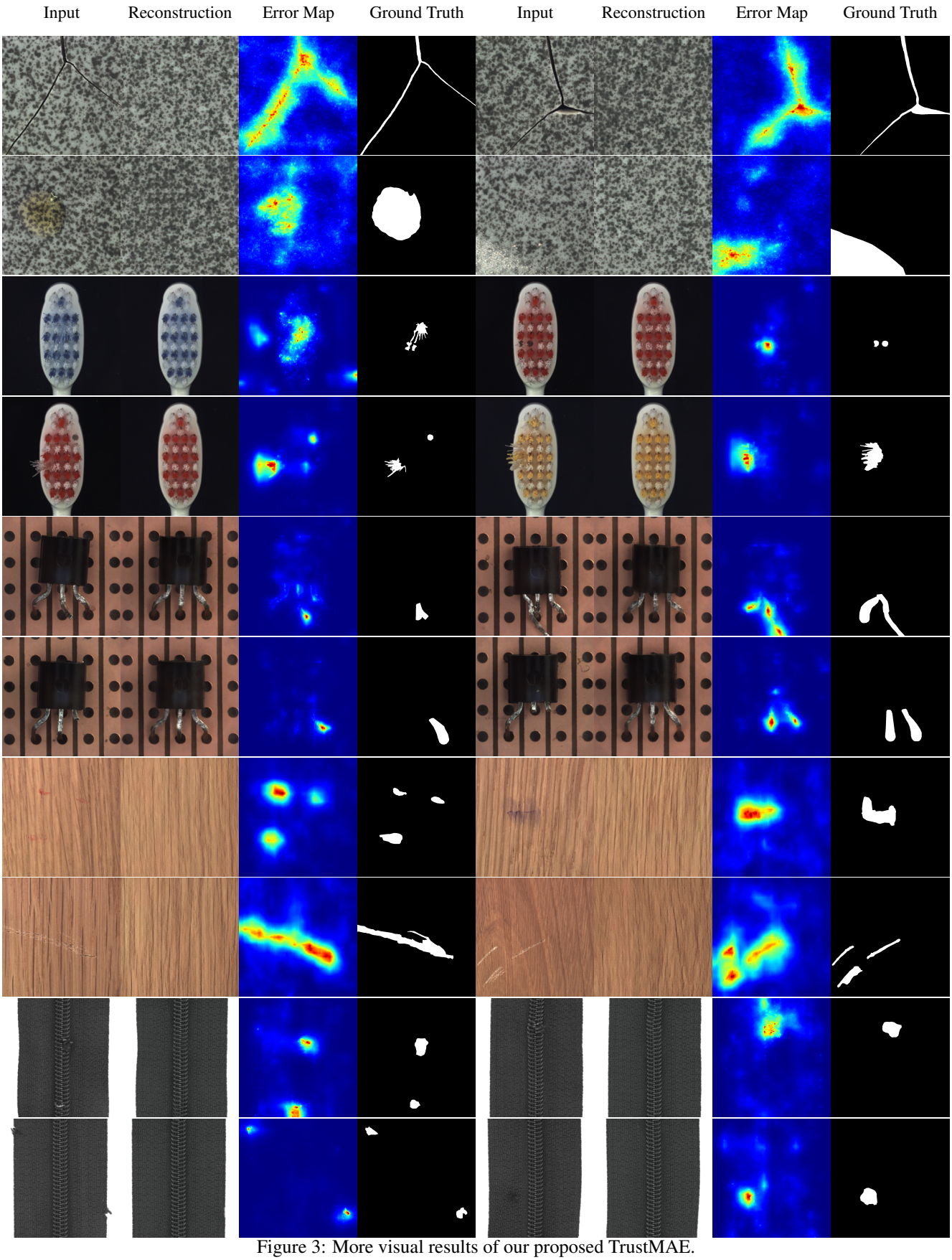

Figure 3: More visual results of our proposed TrustMAE.