




# Semantic Novelty Detection via Relational Reasoning Supplementary Material

Francesco Cappio Borlino<sup>\*,1,2</sup>, Silvia Bucci<sup>\*,1</sup>, and Tatiana Tommasi<sup>1,2</sup>

<sup>1</sup> Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

<sup>2</sup> Italian Institute of Technology, Italy

{francesco.cappio,silvia.bucci,tatiana.tommasi}@polito.it

## 1 Implementation details

We start from a standard ResNet-18 [3], pretrained on ImageNet1k [1], which we use as feature extractor  $f_\theta$  by removing the original final classification layer. Our relational module  $r_\gamma$  has the same structure of the transformer in ViT [2]: we use 4 multi-head self-attention encoder blocks, a number that allows to trade-off performance and time complexity (the number of blocks highly influences the total number of learnable parameters of the network). The features extracted by the backbone are passed through an FC projection layer before entering the transformer. The transformer input sequence is obtained concatenating the learnable label token and the representations of a pair of samples  $[z_l, z_i, z_j]$ . The output token  $v_l$  is then selected and passed through a final FC layer which represents the regression head  $c_\delta$ .

The transformer procedure is summarized in the following equations:

$$z^0 = [z_l; z_i; z_j] \tag{1}$$

$$\tilde{z}^b = \text{MSA}(\text{LN}(z^{b-1})) + z^{b-1}, \quad b = 1 \dots B \tag{2}$$

$$z^b = \text{MLP}(\text{LN}(\tilde{z}^b)) + \tilde{z}^b, \quad b = 1 \dots B \tag{3}$$

$$v_l = \text{LN}(z_l^B) . \tag{4}$$

We train our network on ImageNet1k in an end-to-end manner using the MSE loss (Eq. 1 in the main paper) applied to the output of the regression head. Our training procedure uses 13k iterations with a batch size of 4096, where each element of the batch is an image pair. The learning rate uses a linear warmup for 500 iterations and then is fixed to 0.008. We use LARS optimizer [4] with momentum 0.9 and weight decay  $5 \cdot 10^{-5}$ . We build image pairs by selecting each image of the dataset as anchor and associating it with a randomly chosen sample with the same label to create *positive* pairs and samples of different labels to create *negative* pairs. All experimental results are averaged over three runs.

We summarize in Algorithm 1 and 2 the training and evaluation procedure of ReSeND.

---

\* equal contributions

---

**Algorithm 1** ReSeND train procedure

---

**Require:**  $\mathcal{S}, \mathcal{T}, f_\theta : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^d, r_\gamma, c_\delta$ **procedure** CREATE\_PAIRS( $\mathcal{S}$ )   $pairs = []$   **for each**  $(\mathbf{x}^s, y^s)$  **in**  $\{\mathcal{S}\}$  **do**     $pairs.append((rand\_same\_class(y^s), \mathbf{x}^s, 1))$      $pairs.append((rand\_diff\_class(y^s), \mathbf{x}^s, 0))$   **return**  $pairs$ **procedure** MAIN()  **for**  $epoch$  **in**  $range(n\_epochs)$  **do**     $pairs = create\_pairs(\mathcal{S})$      $shuffle(pairs)$     **for**  $iter$  **in**  $range(itsers\_epoch)$  **do**       $pairs\_batch = next\_batch(pairs)$        $\mathbf{x}_1, \mathbf{x}_2, labels = pairs\_batch$        $\mathbf{z}_1 = f_\theta(\mathbf{x}_1)$        $\mathbf{z}_2 = f_\theta(\mathbf{x}_2)$        $feats\_pairs = (\mathbf{z}_1, \mathbf{z}_2)$        $predictions = c_\delta(r_\gamma(feats\_pairs))$        $MSE\_loss = \mathcal{L}(predictions, labels)$       **Update**  $\theta, \gamma, \delta \leftarrow \nabla MSE\_loss$ ▷ Eq. 1 (main)

---

---

**Algorithm 2** ReSeND eval procedure

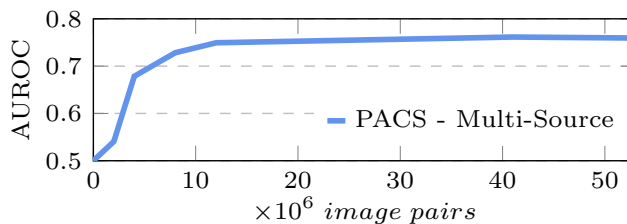
---

**Require:**  $\mathcal{S}, \mathcal{T}, f_\theta : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^d, r_\gamma, c_\delta$ **procedure** COMPUTE\_PROTOTYPES( $\mathcal{S}$ )   $prototypes = zeros((|\mathcal{Y}_s|,))$    $counters = zeros(|\mathcal{Y}_s|)$   **for each**  $(\mathbf{x}^s, y^s)$  **in**  $\{\mathcal{S}\}$  **do**     $\mathbf{z}^s = f_\theta(\mathbf{x}^s)$      $prototypes[y^s] += \mathbf{z}^s$      $counters[y^s] += 1$   **for**  $i$  **in**  $range(|\mathcal{Y}_s|)$  **do**     $prototypes[i] /= counters[i]$   **return**  $prototypes$ **procedure** MAIN()   $normality\_scores = []$    $prototypes = compute\_prototypes(\mathcal{S})$   **for each**  $\mathbf{x}^t$  **in**  $\{\mathcal{T}\}$  **do**     $\mathbf{z}^t = f_\theta(\mathbf{x}^t)$      $pairs = (prototypes, \mathbf{z}^t.repeat())$      $predictions = c_\delta(r_\gamma(pairs))$      $score = max(softmax(predictions))$      $normality\_scores.append(score)$ 

---

## 2 Further Analysis

**Number of image pairs.** Our learning objective is based on the use of image pairs randomly created at training time by coupling samples from the training dataset. Even if the total number of image pairs that could be formed from ImageNet1k dataset is very high ( $\sim 820 \times 10^9$ ), in Fig. 1 we show that ReSeND converges after having seen a relatively small portion of them.



**Fig. 1.** Performance trend increasing the number of image pairs.

## References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
4. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv:1708.03888 (2017)