Latent Space Autoregression for Novelty Detection: Supplementary Materials

Davide Abati Angelo Porrello Simone Calderara Rita Cucchiara

University of Modena and Reggio Emilia,
Via P.Vivarelli, 10, Modena, Italy
name.surname@unimore.it

1. On the implementation details

Architectures and hyperparameters employed for each experiment are reported in Tab. 1, in terms of the type of blocks, autoregressive layers, mini-batch size, learning rate and weight of the log-likelihood objective. All intermediate layers are Leaky ReLU activated. The objective function is optimized using Adam [2]. All hyperparameters are tuned on a held-out validation set, by minimizing the raw objective (Eq. 4 with $\lambda = 1$).

2. On the log-likelihood objective

In this section, we detail how the log-likelihood term (Eq. 4 in the main paper) has been computed and optimized. Importantly, as mentioned in the main paper, we model each CPD through a multinomial. To this aim, we firstly need that the encoder acts as a bounded function. To achieve such desideratum, we simply employ a sigmoidal activation, ensuring that latent representations $\mathbf{z} = f(\mathbf{x}; \theta_f)$ reside in $[0, 1]^d$. Therefore, for each z_j with $j = 1, 2, \dots, d$, we perform a linear quantization of the space $[0, 1]$ in B bins (where B is a hyperparameter). This latter step provides for z_j a B-dimensional categorical distribution $\phi(z_j)$, highlighting the correct bin to which z_j belongs. For each CPD, such distribution will serve as ground truth for the estimator $h(\mathbf{z}; \theta_h)$, the latter coherently predicting d distributions $p(z_j|\mathbf{z}_{<j})$ across the B bins, employing a softmax activation. This way, as shown in Eq. 1, the $\mathcal{L}_{\text{LLK}}$ loss turns out to be a valid likelihood term, defined as the cross-entropy loss between each one of the estimated CPD and their categorical counterparts:

$$\mathcal{L}_{\text{LLK}}(\theta_f, \theta_h) = \mathbb{E}_{\mathbf{x} \sim P} \left[-\sum_{j=1}^{d} \sum_{k=1}^{B} \phi(z_j)_k \log(p(z_j|\mathbf{z}_{<j})_k) \right]. \tag{1}$$

It is worth noting that multinomials are just one of the plausible models for the CPDs. Indeed, if we replace them with Gaussians, the overall framework would leave standing. However, as we observed in different trials, this choice does not yield considerable improvements but rather numerical instabilities, as described in prior works [5].

3. On the relations to Variational Autoencoders

Our framework yields some similarities with the Variational Autoencoder (VAE) [3]. Indeed, they both approximate the integral of Eq. 1 in the main paper through the minimization of the reconstruction error under a regularization con-

	MNIST	CIFAR-10	UCSD Ped2	ShanghaiTech	DR(eye)VE
Input Shape	1,28,28	3,32,32	1,8,32,32*	3,16,256,512	1,16,160,256
Encoder Network	$\mathrm{D}^{32}_{2,2}$ $\mathrm{D}^{64}_{2,2}$ FC^{64} FC^{64}	2D $\mathrm{Conv}^{32}_{3\mathrm{X}3}$ R^{32} $\mathrm{D}^{64}_{2,2}$ $\mathrm{D}^{128}_{2,2}$ $\mathrm{D}^{256}_{2,2}$ FC^{256} FC^{64}	$\mathrm{D}^{8}_{1,2,2}$ $\mathrm{D}^{12}_{2,1,1}$ $\mathrm{D}^{18}_{1,2,2}$ $\mathrm{D}^{27}_{2,1,1}$ $\mathrm{D}^{40}_{1,2,2}$ TFC^{64}	$\mathrm{D}^{8}_{1,2,2}$ $\mathrm{D}^{16}_{1,2,2}$ $\mathrm{D}^{32}_{2,2,2}$ $\mathrm{D}^{64}_{1,2,2}$ $\mathrm{D}^{64}_{2,2,2}$ TFC^{512} TFC^{64}	$\mathrm{D}^{8}_{1,2,2}$ $\mathrm{D}^{16}_{1,2,2}$ $\mathrm{D}^{32}_{2,2,2}$ $\mathrm{D}^{64}_{1,2,2}$ $\mathrm{D}^{64}_{2,2,2}$ TFC^{512} TFC^{64}
Decoder Network	FC^{64} FC^{64} $\mathrm{U}^{32}_{2,2}$ $\mathrm{U}^{16}_{2,2}$ 2D $\mathrm{Conv}^{1}_{1\mathrm{x}1}$	FC^{256} FC^{256} $\mathrm{U}^{128}_{2,2}$ $\mathrm{U}^{64}_{2,2}$ $\mathrm{U}^{32}_{2,2}$ R^{32} 2D $\mathrm{Conv}^{3}_{1\mathrm{x}1}$	TFC^{64} $\mathrm{U}^{40}_{1,2,2}$ $\mathrm{U}^{27}_{1,2,2}$ $\mathrm{U}^{18}_{2,1,1}$ $\mathrm{U}^{12}_{1,2,2}$ $\mathrm{U}^{8}_{2,1,1}$ 3D $\mathrm{Conv}^{1}_{1\mathrm{x}1}$	TFC^{64} TFC^{512} $\mathrm{U}^{64}_{2,2,2}$ $\mathrm{U}^{32}_{1,2,2}$ $\mathrm{U}^{16}_{2,2,2}$ $\mathrm{U}^{8}_{1,2,2}$ $\mathrm{U}^{8}_{1,2,2}$ 3D $\mathrm{Conv}^{3}_{1\mathrm{x}1}$	TFC^{64} TFC^{512} $\mathrm{U}^{64}_{2,2,2}$ $\mathrm{U}^{32}_{1,2,2}$ $\mathrm{U}^{16}_{2,2,2}$ $\mathrm{U}^{8}_{1,2,2}$ $\mathrm{U}^{8}_{1,2,2}$ 3D $\mathrm{Conv}^{1}_{1\mathrm{x}1}$
Estimator Network	MFC^{32} MFC^{32} MFC^{32} MFC^{32} MFC^{100}	MFC^{32} MFC^{32} MFC^{32} MFC^{32} MFC^{100}	MSC^{4} MSC^{4} MSC^{4} MSC^{4} MSC^{100}	MSC^{4} MSC^{4} MSC^{100}	MSC^{4} MSC^{4} MSC^{4} MSC^{4} MSC^{100}
Mini Batch	256	256	2760	8	16
Learning Rate	10^{-4}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
λ	1	0.1	0.1	1	1

*Patches extracted from input clips having shape 1,16,256,384.

Table 1. Architectural and optimization hyperparameters of each setting. We denote with D^{C}_{S} (downsampling), U^{C}_{S} (upsampling) and R^{C} (residual) the parametrizations for the employed building blocks (see Fig. 1ii in the main paper). On the one hand, C is the number of output channels, whereas S is the stride of the first convolution in the block. Additionally, FC^{C} and TFC^{C} denote dense layers and temporally-shared full connections respectively (in this case, C is the number of output features). Finally, we refer to MFC^{C} and MSC^{C} for the proposed autoregressive layers, illustrated in Fig. 3 in the manuscript. For a comprehensive description of each type of layer, please refer to Sec. 3.1 of the main paper.

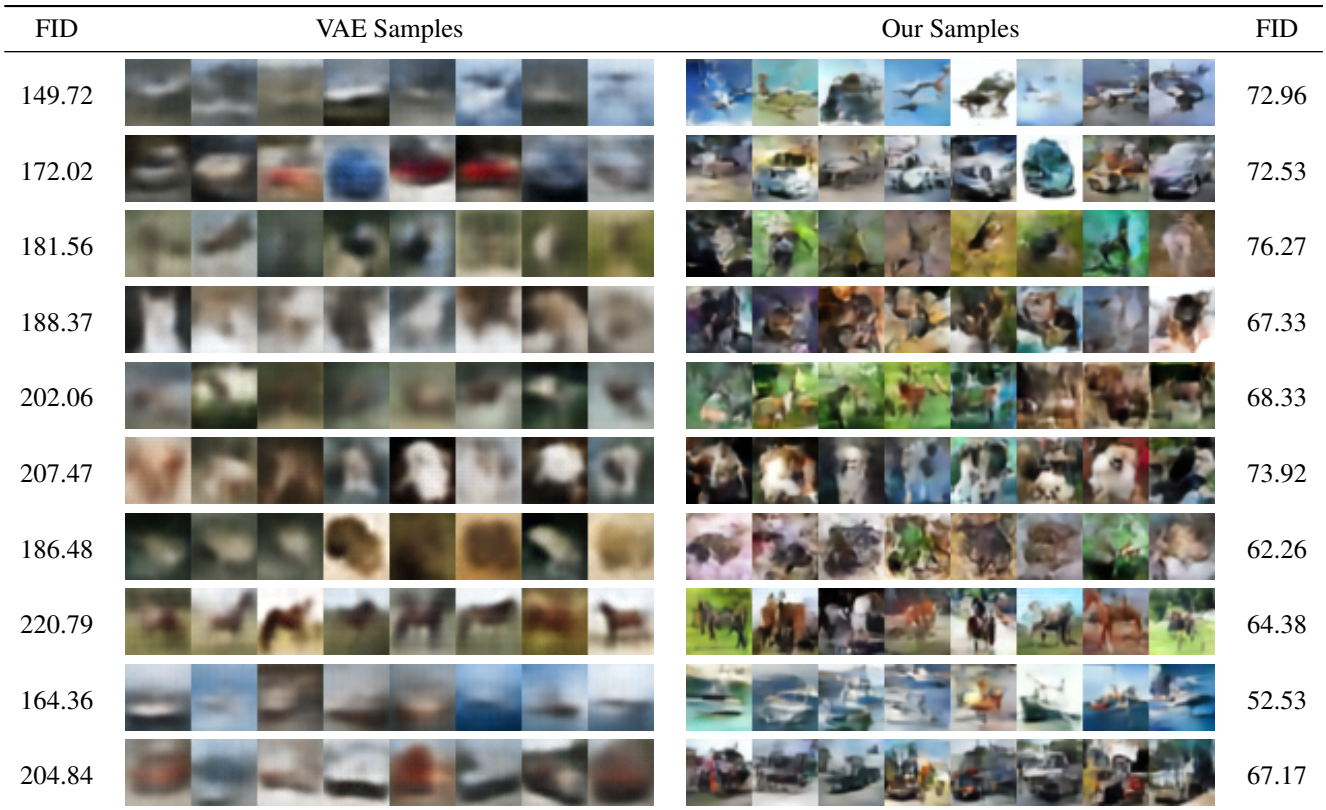


Figure 1. For all CIFAR-10 classes (organized in different rows), we report images sampled from VAEs (left) and the proposed autoencoders with autoregressive priors. As can be seen, our samples visually exhibit fine-grained details and sharpness, differently from the heavily blurred ones coming from VAEs. Finally, the over-regularization arising from VAE is confirmed when looking at FID scores (at the extremes of the figure, the lower, the better).

straint involving a prior distribution on latent vectors. However, it is worth noting several fundamental distinctions. Firstly, our model does not provide an explicit strategy to sample from the posterior distribution, thus resulting in a deterministic mapping from the input to the hidden representation. Secondly, while VAE specifies an explicit and adamant form for modeling the prior $p(\mathbf{z})$, in our formulation its landscape is free from any assumption and directly learnable as a result of the estimator's autoregressive nature. On this point, our proposal leads to two beneficial aspects. First, as the VAE forces the codes' distribution to match the prior, their differential entropy converges to be the same as the prior. This behavior results in approximately stationary entropies across different settings (appreciable in Fig. 2 in the main paper, where we discuss the intuition behind the entropy minimization within a novelty detection task). Secondly, the employment of a too simplistic prior may lead to over-regularized representations, whereas our proposal is less prone to such risk. Empirical evidence of such behavior can also be appreciated in Fig. 1, where we draw new samples from VAE and our model, both of which has been trained on CIFAR-10. All settings being equal, our hallucinations are visually much more realistic than the ones coming from VAEs, the latter leading to over-smooth shapes and lacking any details, as further confirmed by the substantial differences in Fréchet Inception Distance (FID) scores [1].

4. On the dual nature of novelty

In this section, we stress how significant is the presence of both terms for obtaining a highly discriminative novelty score (NS, Eq. 9 in the main paper): namely the reconstruction error (REC), modeling the memory capabilities, and the

	LLK	REC	NS
MNIST	0.926	0.949	**0.975**
CIFAR-10	0.627	0.603	**0.641**
UCSD Ped2	0.933	0.909	**0.954**
ShanghaiTech	0.695	**0.726**	0.725
DR(eye)VE	0.917	0.863	**0.926**

Table 2. For each setting, AUROC performances under three different novelty scores: i) the log-likelihood term (LLK), ii) the reconstruction term (REC), and iii) the proposed scheme accounting for both (NS).

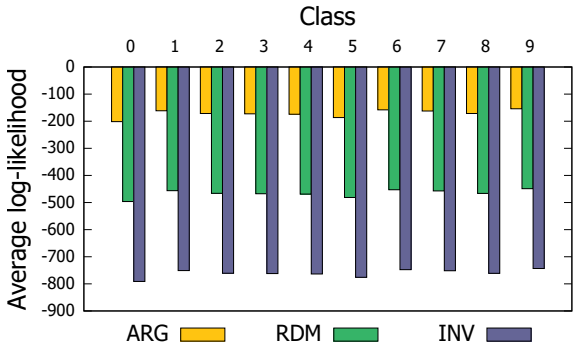


Figure 2. Sample training log-likelihood of a Bayesian Network modeling the distribution of latent codes produced by the encoder of our model trained on MNIST digits. When the BN structure resembles the autoregressive order imposed during training, a much higher likelihood is achieved. This behavior is consistent in all classes and supports the capability of the encoder to produce codes that respect a pre-imposed autoregressive structure.

log-likelihood term (LLK), capturing the surprisal inducted from latent representations. Aiming to reinforce this latter point, just briefly illustrated in Fig. 4 of the manuscript, we report in Tab. 2 performances - expressed in AUROC - delivered by different scoring strategies on each setting mentioned in the main paper. Except for ShanghaiTech, we systematically observe a reward in accounting for both aspects. Furthermore, for MNIST and CIFAR-10, we find particularly interesting the gap in performance arising from our reconstruction error w.r.t. the one arising from the denoising autoencoder (DAE) variants (0.942 and 0.590 for the two datasets respectively, as reported in Tab. 1 of the main paper). In this respect, we gather new evidence supporting that surprisal minimization acts as a novelty-oriented regularizer for the overall architecture, as it improves the discriminative capability of the reconstruction (as already conjectured in Sec. 4.1 of the main paper).

5. On the causal structure of representations

We now investigate the capability of our encoder to produce representations that respect the autoregressive causal structure imposed by the LLK loss (mentioned in Sec. 3 of the main paper). To this aim, we extract representations out of the ten models trained on MNIST digits and fit their distribution using a structured density estimator. Specifically, we employ Bayesian Networks (BNs) with different autoregressive structures. In this respect, each BN is modeled with Linear Gaussians [4], s.t. each CPD $p(z_i|Pa(z_i))$ with $i = 1, 2, \ldots, d$ is given by:

$$p(z_i|Pa(z_i)) = \mathcal{N}(z_i \mid w_0^{(i)} + \sum_{z_j \in Pa(z_i)} w_j^{(i)} z_j, \sigma_i^2), \quad (2)$$

where each $w_j^{(i)}$, σ_i^2 are learnable parameters. We indicate with $Pa(z_i)$ the parent variables of z_i in the BN. The previous equation holds for all nodes, except for the root one, which is modeled through a Gaussian distribution. Concerning the BN structure, we test:

- Autoregressive order: the BN structure follows the autoregressive order imposed during training, namely $Pa(z_i) = \{z_j \mid j = 1, 2, \ldots, i-1\}$
- Random order: the BN structure follows a random autoregressive order.
- Inverse order: the BN structure follows an autoregressive order which is the inverse with respect to the one imposed during training, namely $Pa(z_i) = \{z_j \mid j = i+1, i+2, \ldots, d\}$

It is worth noting that, as the three structures exhibit the same number of edges and independent parameters, the difference in their fitting capabilities is only due to the causal order imposed over variables.
Fig. 2 reports the sample training log-likelihood of all BN models. Remarkably, the autoregressive order delivers a better fit, supporting the capability of the encoder network to extract features with learned autoregressive properties. Moreover, to show that this result is not due to overfitting or other lurking behaviors, we report in Tab. 3 log-likelihoods for training, validation and test set.

6. On the entropy minimization

To provide an additional grasp about the role of the representation's entropy minimization, we focus on a single MNIST digit (class 7) and report in Fig. 3 some randomly

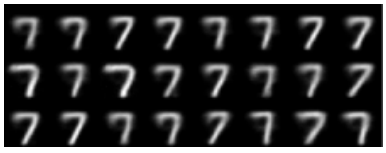
Loss weight	Reconstructions
$\lambda = 0.01$	
$\lambda = 1$	
$\lambda = 100$	

Figure 3. MNIST reconstructions delivered by different values of λ, the latter controlling the impact of the differential entropy minimization.

		Classes									
		0	1	2	3	4	5	6	7	8	9
ARG	Train	-201.60	-161.60	-171.43	-172.73	-174.17	-186.48	-158.22	-162.37	-171.65	-154.11
	Val	-200.96	-160.38	-170.10	-172.29	-173.85	-185.25	-157.22	-162.20	-171.42	-154.02
	Test	-200.89	-159.73	-169.64	-170.75	-172.40	-184.27	-157.74	-161.65	-170.10	-152.70
RDM	Train	-496.33	-456.34	-466.16	-467.47	-468.90	-481.21	-452.95	-457.10	-466.39	-448.84
	Val	-495.69	-455.11	-464.83	-467.02	-468.58	-479.98	-451.95	-456.93	-466.15	-448.75
	Test	-495.62	-454.47	-464.37	-465.48	-467.13	-479.00	-452.48	-456.38	-464.83	-447.43
INV	Train	-791.06	-751.07	-760.89	-762.20	-763.63	-775.94	-747.68	-751.83	-761.12	-743.57
	Val	-790.42	-749.84	-759.56	-761.75	-763.31	-774.71	-746.68	-751.66	-760.88	-743.48
	Test	-790.35	-749.20	-759.11	-760.22	-761.86	-773.73	-747.21	-751.12	-759.56	-742.16

Table 3. Sample log-likelihood obtained by different BN structures when fitting MNIST representations. Each BN is trained on latent codes computed from the training set of a single class, following either the autoregression order (ARG), a random order (RDM) or the order inverse to autoregression (INV). We report the log-likelihood also on the validation and test set. For train-val-test split, see Sec 4.1 of the paper. Only "normal" test samples are used in this evaluation.

sampled reconstructions from the training set. Such reconstructions are learned under three different regularization regimes, represented by different weights on the log-likelihood objective (λ, Eq. 4 in the main paper). As shown in Fig. 3, higher degrees of regularization (i.e., stricter constraints on entropy) deliver near mode-collapsed reconstructions, losing sharp variations in favor of capturing fewer prototypes for the input distribution.

7. On the complexity of autoregressive layers

In this section, we briefly discuss the complexity of Masked Fully Connected (MFC) and Masked Stacked Convolution (MSC) layers (Fig. 3 of the main paper)[1]: adhering to the notation introduced in Sec. 3 from the main paper, MFC exhibits $\frac{d^2+d}{2} \cdot ci \cdot co + d \cdot co$ trainable parameters and a computational complexity $\mathcal{O}(d^2 \cdot ci \cdot co)$. MSC, instead, features $\frac{3d^2+d}{2} ci \cdot co + d \cdot c_o$ free parameters and a time complexity $\mathcal{O}(d^2 \cdot ci \cdot co \cdot t)$.

8. On the localizations and novelty scores in video anomaly detection

We show in Fig. 4 other qualitative evidence of the behavior of our model in video anomaly detection settings, namely UCSD Ped2 and ShanghaiTech.

References

[1] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems*, pages 6626–6637, 2017. 2

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. 1

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014. 1

[4] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. 3

[5] A. van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *International Conference on Machine Learning*, 2016. 1

[1] We refer to the type 'B' of both layers, since it is an upper bound to the type 'A'

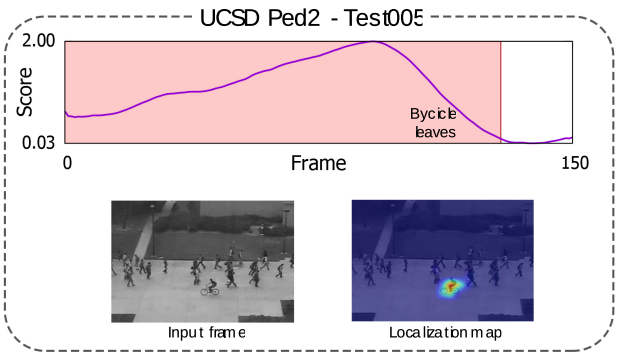


Figure 4. Novelty scores and localizations maps for several test clips from UCSD Ped2 (left) and ShanghaiTech (right).