Generative Pseudo-label Refinement for Unsupervised Domain Adaptation
Supplementary Material

Pietro Morerio[1], Riccardo Volpi[1], Ruggero Ragonesi[1,2], Vittorio Murino[1,3,4]

{pietro.morerio,riccardo.volpi,ruggero.ragonesi,vittorio.murino}@iit.it

[1]Pattern Analysis & Computer Vision - Istituto Italiano di Tecnologia

[2]Università degli Studi di Genova, Italy

[3]Computer Science Department - Università di Verona, Italy

[4]Huawei Technologies Ltd., Ireland Research Center

1. Architectures and Hyperparameters

We provide here a detailed description of the networks used for our experiments.

Figure 1, Figure 2 and Figure 3 depict the architectures used for C (classifier), G (GAN's generator) and D (GAN's discriminator), respectively, in the different benchmark experiments.

We report in the following the hyperparameters associated with the same experiments. We use Adam optimizer [2] in all the experiments, and set the learning rate to train pre-train C on data from the source distribution to $3 \cdot 10^{-4}$. For the cGAN pre-training, we set the learning rate for training both G and D to 10^{-5}. When running Algorithm 1, we set $\eta = 10^{-5}$ and $\delta = 5 \cdot 10^{-5}$.

Architectural choices, as well as hyperparameter tuning, were carried out with the goal of making GANs converge.

2. Are deeper architectures more resistant against shift noise?

In [3] the authors provide empirical evidence that deeper models (e.g. Residual Networks [1]) are more robust against uniform label noise than shallow architectures. We investigated whether such resilience of deep models arises also with shift noise. Our experiments led us to exclude such hypothesis. We considered the split MNIST $\rightarrow$ SVHN, where shift noise is very significant, and repeated the experiment of Table 2: we trained different ResNets (from scratch) with different depths on target samples corrupted by shift noise; we observe that despite improved capacity of the models, they overfit the noisy labelled samples and are not able to reduce δ_A. (see Table **??** and Figure 4).

Architecture	**Shift noise**		**Classifier**	
	a	δ_A	a	δ_A
C	0.3005	0.3739	0.3212	0.3741
ResNet-50			0.2998	0.3741
ResNet-101			0.3004	0.3739
ResNet-152			0.2997	0.3736

Table 1. MNIST $\rightarrow$ SVHN: we observe no improvements in accuracy wrt shallower classifiers that we trained in the paper. Even δ_A's do not sink as it happens for generative models.

3. Generated images

We report in Figures 5, 6, 7, 8 and 9 samples generated by G after the training procedure defined by Algorithm 1, for the splits SVHN $\rightarrow$ MNIST, MNIST $\rightarrow$ SVHN, MNIST $\rightarrow$ MNIST-M, MNIST $\rightarrow$ USPS and USPS $\rightarrow$ MNIST, respectively. For each experiment, we randomly generated 20 samples associated with the different classes and reported them in the Figures, where each row is related to a different class.

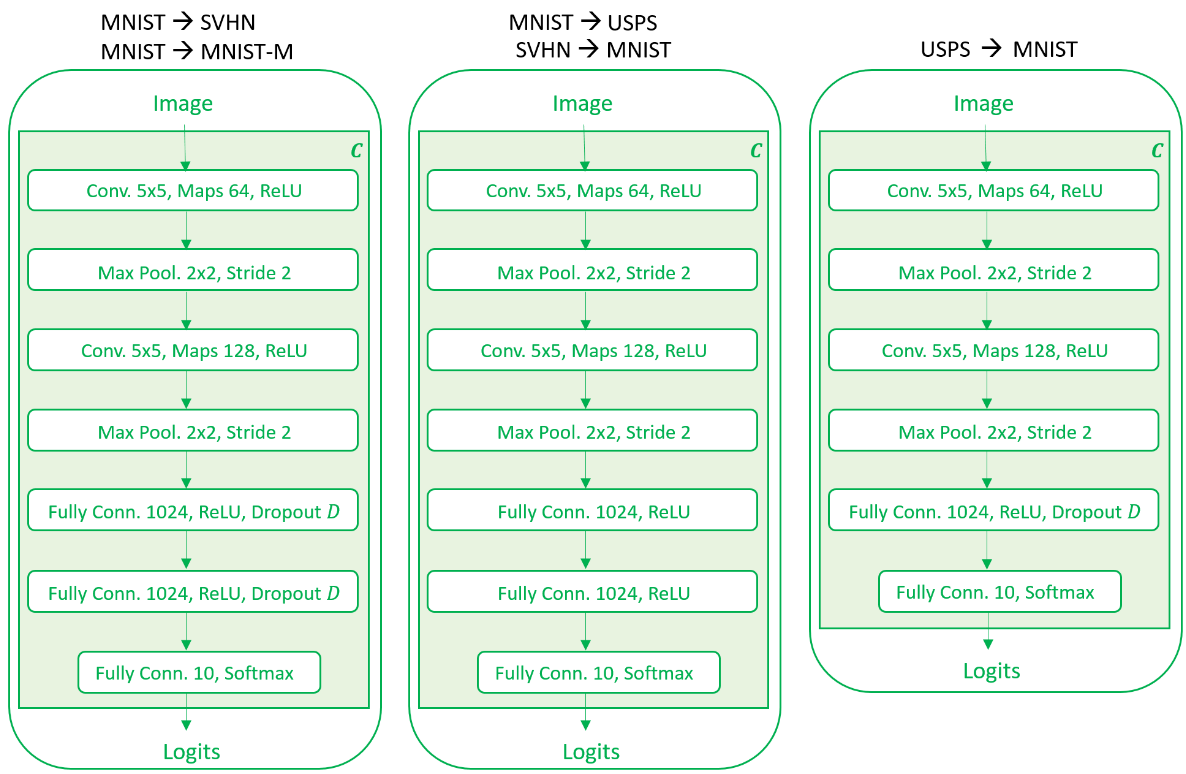

Figure 1. Architectures for the classifier C (see Figure 4 in the paper).

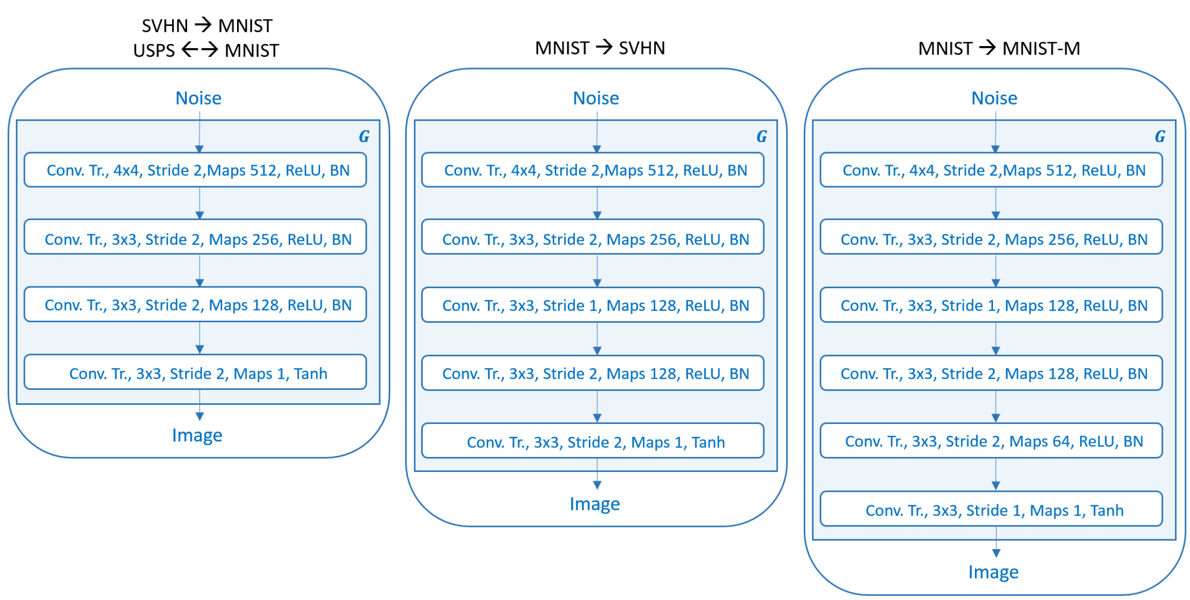

Figure 2. Architectures for the generator G (see Figure 4 in the paper).

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 1
[2] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 1

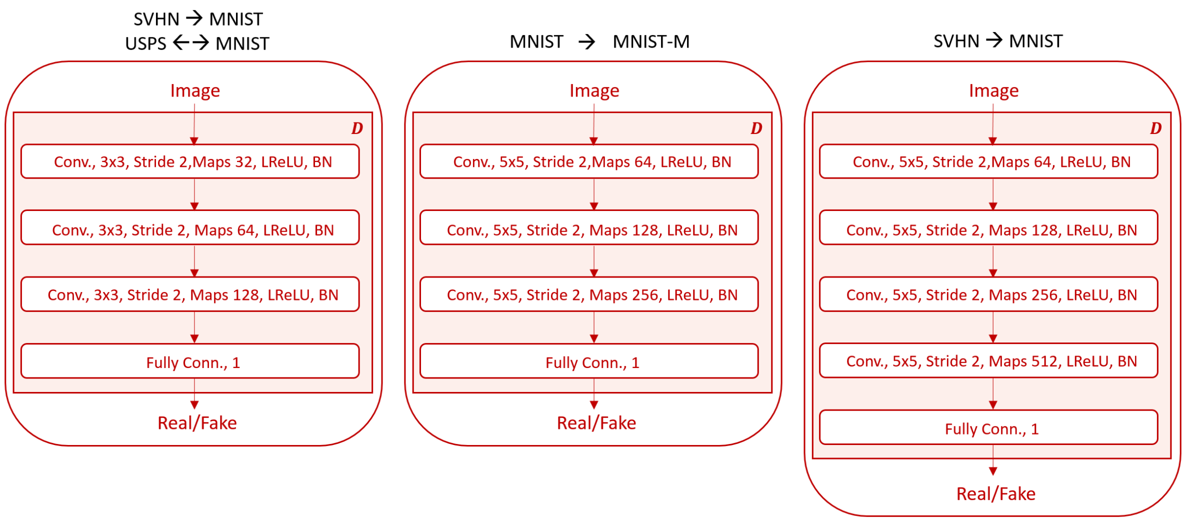

Figure 3. Architectures for the discriminator D (see Figure 4 in the paper).

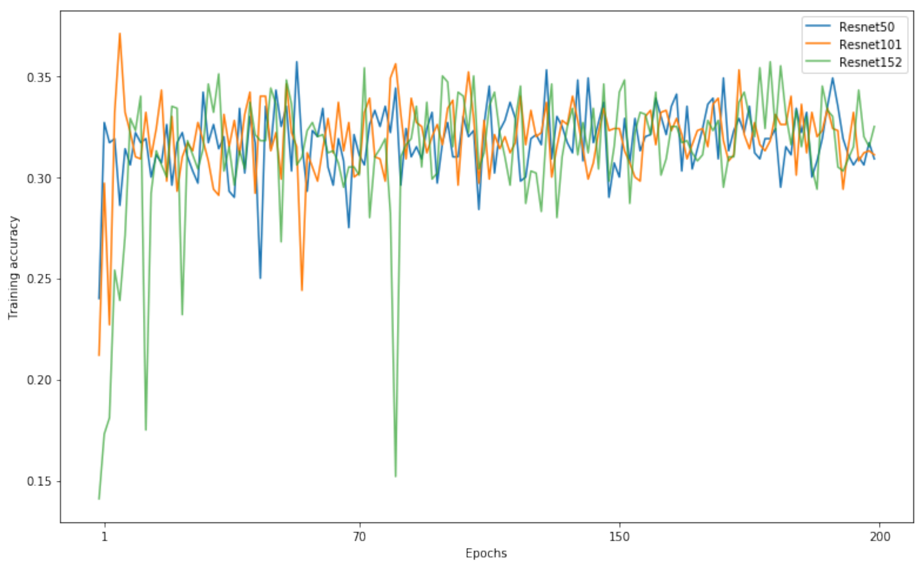

Figure 4. Training on the shift noise: we evaluate the accuracy on *clean training set* at the end of each epoch. Despite ResNet models are deeper and more resistant to uniform noise [3], they are not robust against shift noise. Indeed, accuracy on the *noisy training set* reaches about 100% pointing out that the models overfit noise.

[3] D. Rolnick, A. Veit, S. J. Belongie, and N. Shavit. Deep learning is robust to massive label noise. *CoRR*, abs/1705.10694, 2017. 1, 3

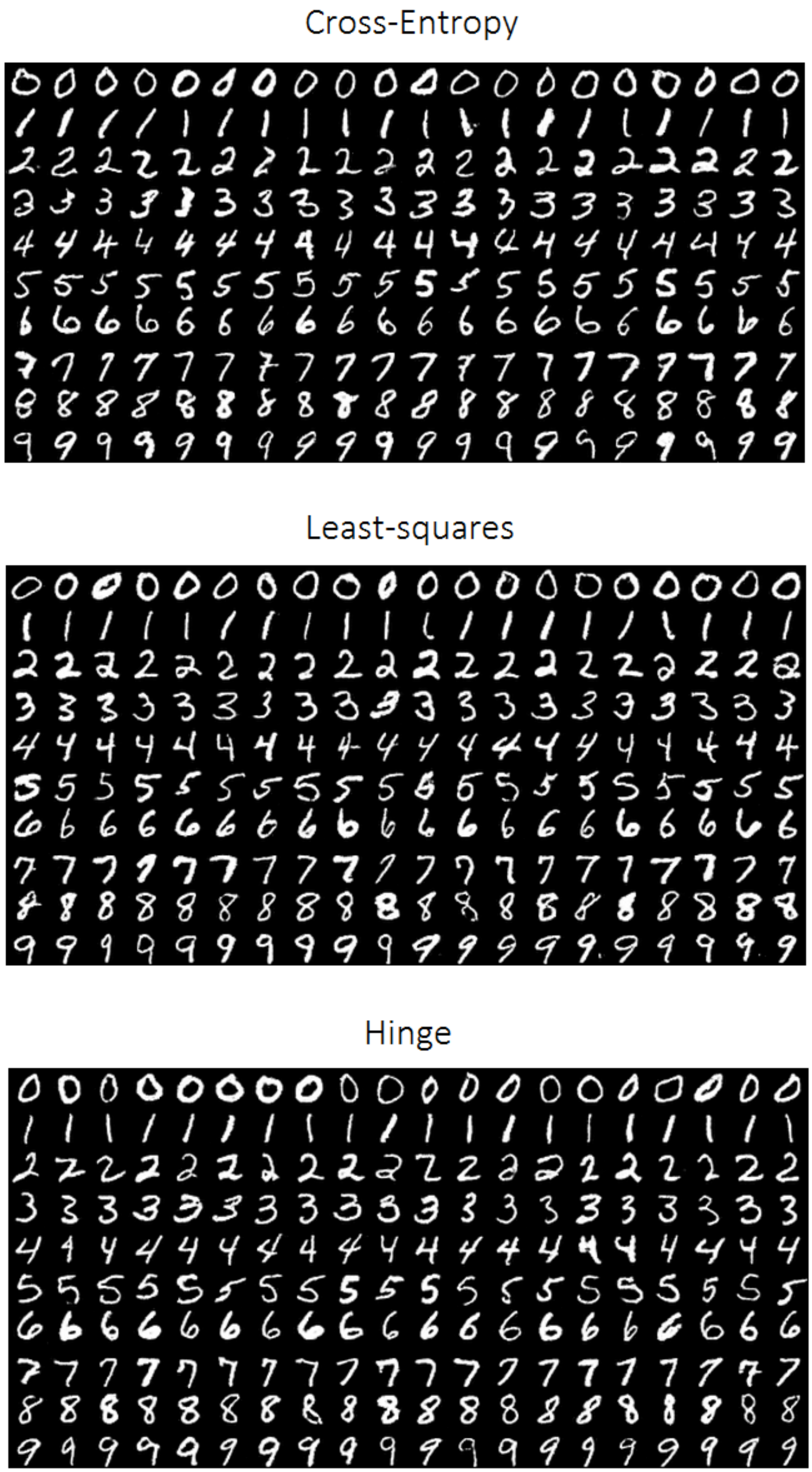

Figure 5. MNIST samples generated by G, trained with Algorithm 1 (SVHN $\rightarrow$ MNIST split). Each row is related to a different label code (from *top* to *bottom*, 0 to 9).

Figure 6. SVHN samples generated by G, trained with Algorithm 1 (MNIST $\rightarrow$ SVHN split). Each row is related to a different label code (from *top* to *bottom*, 0 to 9).

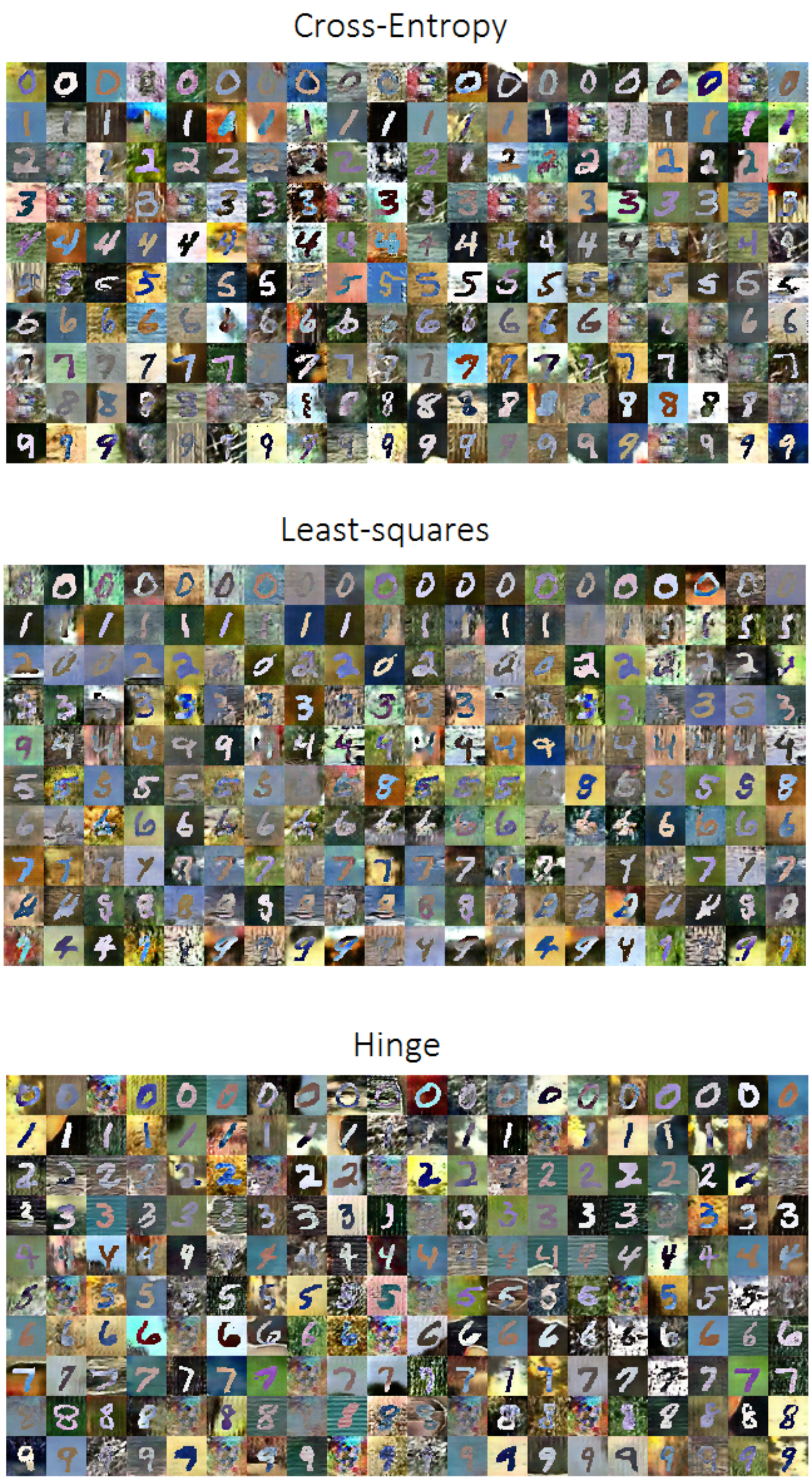

Figure 7. MNIST-M samples generated by G, trained with Algorithm 1 (MNIST $\rightarrow$ MNIST-M split). Each row is related to a different label code (from *top* to *bottom*, 0 to 9).

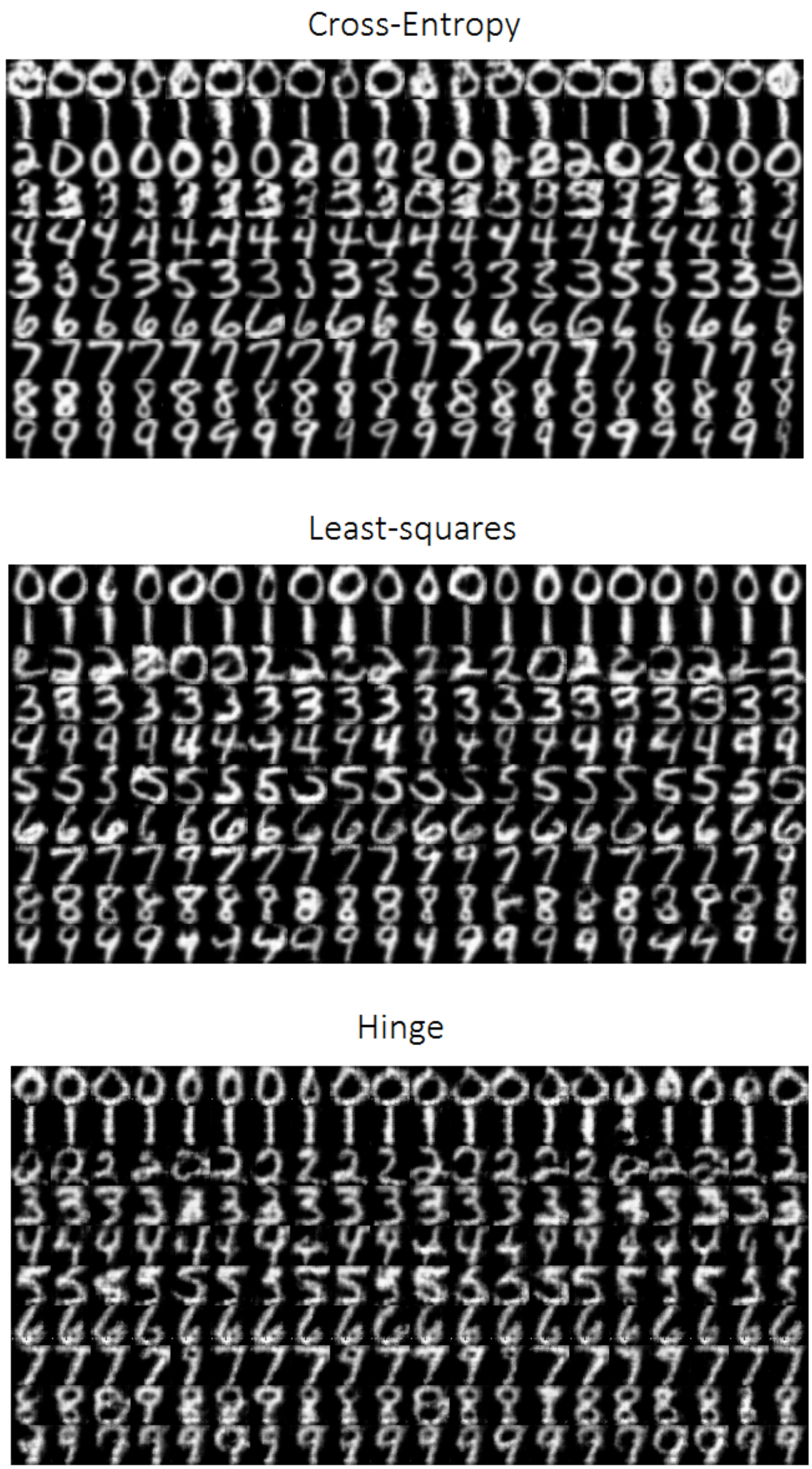

Figure 8. USPS samples generated by G, trained with Algorithm 1 (MNIST $\rightarrow$ USPS split). Each row is related to a different label code (from *top* to *bottom*, 0 to 9).

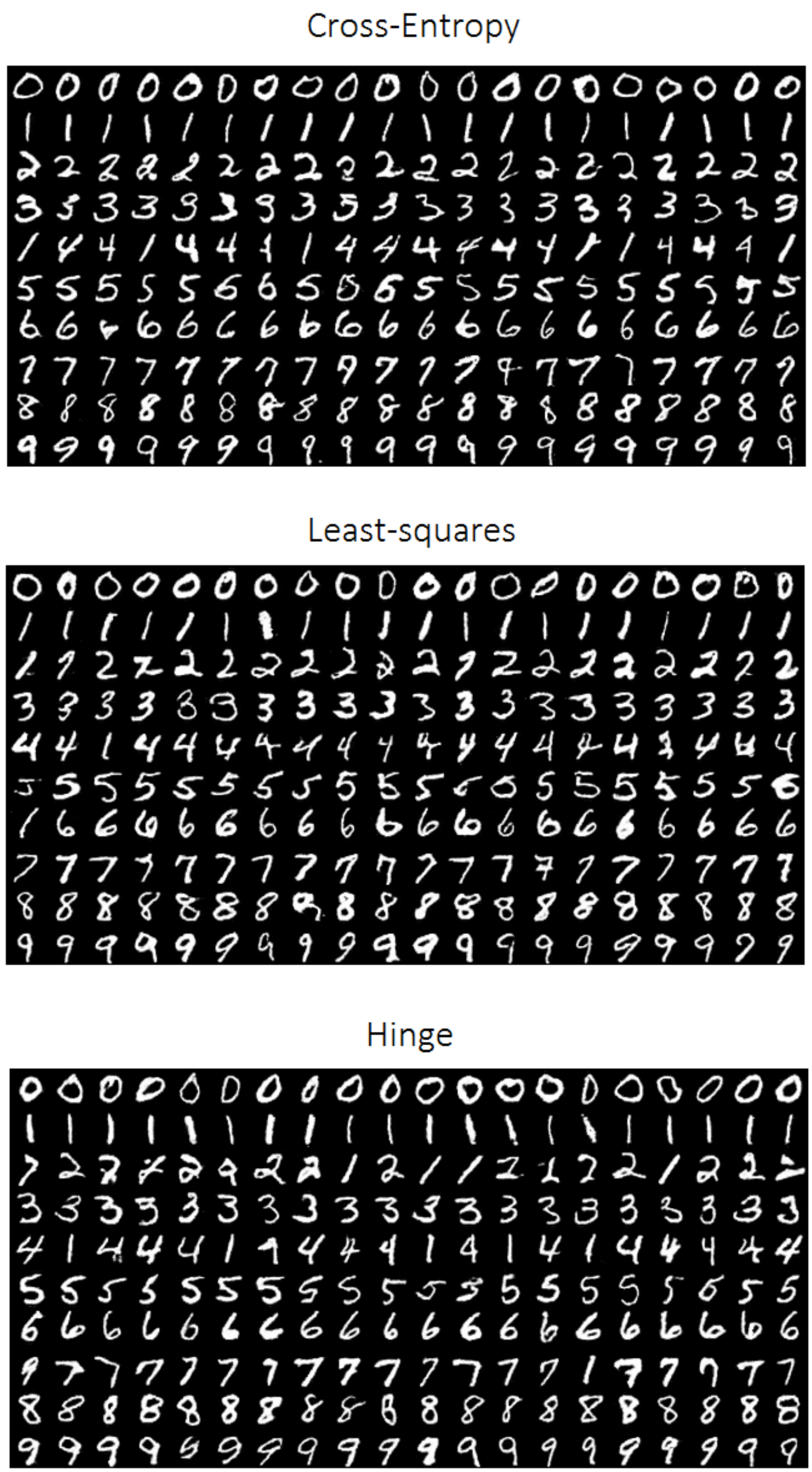

Figure 9. MNIST samples generated by G, trained with Algorithm 1 (USPS $\rightarrow$ MNIST split). Each row is related to a different label code (from *top* to *bottom*, 0 to 9).