Appendix

The appendix is organised as follows:

- Appendix A presents a detailed description of InstanceGM-SS and experimental details of self-supervision. Appendix A.1 contains the experimental details for the self-supervision method DINO, and Appendix A.2 contains the experimental details of InstanceGM with self-supervision (InstaceGM-SS).

- Appendix B shows the motivation behind the use of the continuous Bernoulli distribution.

- Appendix C shows the results on high IDN rates for CIFAR10 dataset.

A. Self-supervision and experimental details

In addition to training the proposed method from scratch, we also adapt self-supervised learning to pre-train the feature extractor part in the classifier $q(Y|X)$, denoted as InstanceGM-SS in Table 2. In particular, we employ DINO [8] to self-supervisedly learn a feature extractor using the unlabelled data from the training set of Red Mini-Imagenet (DINO is a self-supervision method that uses self-distillation). Such integration allows our proposed method to be fairly compared with other label noise learning approaches that rely on self-supervision, such as PropMix [12].

A.1. Experimental details of self-supervision DINO

We trained the self-supervised model on Red Mini-Imagenet for 500 epochs on PreAct-ResNet-18 (PRN18). We used the same set of hyper-parameters as provided by DINO. The method follows the teacher-student setting where the weights of the teacher network are exponentially weighted averaged from the student network [19]. It includes the teacher model temperature for warmup as 0.04 and 0.07 for training, and warmup teacher epochs as 50. The L2 weight decay regularisation is 0.000001, and batch size is 51 per gpu. The initial learning rate is set to 0.3 and the minimum learning rate is set to 0.0048, with the training warm up number of epochs set as 10. In addition,DINO needs various augmented views of the input image. It includes multi-crop strategy [7] with high-resolution global and low-resolution local views. The two versions of global crops views are considered with scale values of 0.14 and 1. Moreover, the six different local crops views are considered having scale values of 0.05 and 0.14, with teacher momentum as 0.996.

A.2. Experimental details of InstanceGM-SS

When we use the self-supervised trained classifier for InstanceGM-SS, we slightly change the settings to train the Red Mini-Imagenet, and results could be find in Table 2. In particular, we use the self-supervised PreAct-ResNet-18 as a classifier with the latent representation Z of size 25. We train the network for 80 epochs with the learning rate reduced by 10 after 50 epochs. The warmup stage is reduced to 15 epochs. Otherwise, the previous settings for Red Mini-ImageNet without self-supervision are kept the same.

B. Motivation of using continuous Bernoulli distribution

To explain the motivation behind the use of the continuous Bernoulli likelihood for image reconstruction, we refer to the variational inference technique. In particular, we denote x as an observable variable, e.g., input images, while z as a hidden (or latent) variable. For simplicity, we assume that both x and z are scalars. In variational inference, e.g. VAE, the objective is to maximise the evidence lower bound (ELBO) or minimise the variational-free energy w.r.t. ϕ – the parameter of the variational posterior $q_{\phi}(z|x)$:

$$\min_{\phi} \underbrace{\mathbb{E}_{q_{\phi}(z|x)}[-\ln p_{\theta}(x|z)]}_{\text{Reconstruction loss}} + \beta \, \mathrm{KL}[q_{\phi}(z|x) || p(z)], \tag{7}$$

where $p(z)$ is the prior of z (for example, a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$), $\beta \in \mathbb{R}_{+}$ is a re-weighting factor. In theory, $\beta = 1$, and we use β here to explain common practice in VAE which is described in the following.

The second term in (7) could be evaluated with a closed-form formula for some simple cases of $q_{\phi}(z|x)$ and $p(z)$ or approximated using Monte Carlo sampling. Thus, we would focus on the explanation of the first term – often known as reconstruction loss. Depending on how $p_{\theta}(x|z)$ is modelled, we could have different reconstruction losses, as explained below.

B.1. Gaussian likelihood

If $p_\theta(x|z)$ is a Gaussian distribution: $p_\theta(x|z) = \mathcal{N}(x; \mu(z), \sigma^2(z)))$, the negative log-likelihood term in (7) can be written as:

$$- \ln p_\theta(x|z) = \ln\left(\sigma(z)\sqrt{2\pi}\right) + \frac{1}{\sigma^2(z)}(\mathbf{x} - \mu(\mathbf{z}))^{\mathbf{2}}. \tag{8}$$

The correct form of the reconstruction loss in (8) contains two terms including a "weighted" MSE term. However, common practice simply replaces the whole $-\ln p_\theta(x|z)$ by $(\mathbf{x} - \mu(\mathbf{z}))^{\mathbf{2}}$, resulting in an incorrect formula. As a result, it requires to fine-tune β to some small value to balance the contributions of the first and second terms in (7).

B.2. Bernoulli likelihood

If $p_\theta(x|z)$ is a Bernoulli distribution: $p_\theta(x|z) = \mathcal{B}(\lambda_\theta(z))$ where $x \in \{0, 1\}$ and $\lambda_\theta(z) \in [0, 1]$, then the negative log-likelihood in (7) is:

$$- \ln p_\theta\left(x|z\right) = -x \ln \lambda_\theta\left(z\right) - \left(1 - x\right) \ln\left(1 - \lambda_\theta\left(z\right)\right), \tag{9}$$

resulting in the binary cross-entropy loss (BCE) [13].

Simply implementing the reconstruction loss as BCE results in the pervasive error since the input x must be in {0, 1} [40], which is applicable for black and white images only.

B.3. Continuous Bernoulli likelihood

For colour images, although one can model $p_\theta(x|z)$ as a Gaussian distribution shown in (8), it might be a suboptimal choice since the support of the Gaussian distribution is un-bounded, while image data is bounded. Thus, we use the continuous Bernoulli distribution to model $p_\theta(x|z)$ [40] since the continuous Bernoulli distribution is supported in $[0, 1]$ with only one parameter:

$$p_\theta(x|\lambda_\theta) = C(\lambda_\theta)\lambda_\theta^x(1 - \lambda_\theta)^{1-x}, \text{ where } C(\lambda_\theta) = \begin{cases} \frac{2\tanh^{-1}(1-2\lambda_\theta)}{1-2\lambda_\theta} & \text{if } \lambda_\theta \neq 0.5 \\ 2 & \text{otherwise.} \end{cases} \tag{10}$$

Note that one could also use the Beta distribution whose support space is also $[0, 1]$. The advantage of using the continuous Bernoulli distribution is the simplicity since we need only one parameter per pixel, while the Beta distribution requires double the number of parameters.

C. Experimental Results on CIFAR10 at High IDN Levels

We investigated the performance of InstanceGM on high IDN levels including $0.7, 0.8$ and 0.9. We provided the test classification accuracy on CIFAR10 on Table 7. The competing model results are from [26]. Our InstanceGM shows superior results even in such high noise rate problems. Note that all results at these high noise rate problems are not good but the performance degradation for InstanceGM is lower, compared to the other models.

Table 7. Test accuracy (%) for CIFAR10 at high IDN rates. All the mentioned results of other methods are as presented in the paper [26].

Method	**IDN - CIFAR10**		
	0.7	**0.8**	**0.9**
PTD-F-V [62]	20.35	13.58	09.44
PTM-F-V [26]	18.95	13.89	10.57
IF-F-V [26]	21.09	16.72	10.86
DivideMix [33]	22.13	08.10	04.08
InstanceGM	**47.23**	**29.30**	**11.01**