A EXPERIMENTAL SETUP

Data partitioning and non-i.i.d.-ness For the label-skew setting, we use the Dirichlet splits for CIFAR 10, 100 discussed at Reddi et al. (2020) with $\alpha = 0.1$ in both cases. Notice that we adopt the convention of Hsu et al. (2019) where α is multiplied by the prior probability of the label in the dataset, so, for example, in the case of CIFAR 10 the final concentration parameter is 0.01.

For the covariate shift setting we consider the case of rotation non-i.i.d.-ness. More specifically, we first perform an i.i.d., with respect to the labels, split of the data into 100 and 500 clients for CIFAR 10 and CIFAR 100 respectively. Afterwards, we bin the $[0, 2\pi]$ range into 10 rotation bins and then assign to each client bins according to a Dirichlet distribution with $\alpha = 0.1$. In this case, each client receives one or two bins of rotations. After that bin assignment, each client randomly rotates each image of their local dataset once with an angle for each image sampled i.i.d. from the bins selected at that client. For the evaluation we consider non-rotated images.

For the joint shift setting we mix the two cases above by first performing a non-i.i.d., Dirichlet split, *i.e.*, $\alpha = 0.1$, according to the labels and then apply the non-i.i.d. rotation strategy described above.

Architecture details For all methods we use the same encoder model, a ResNet18 architecture adapted for CIFAR 10/100 by replacing the kernel of the first convolutional layer with a $3 \times 64 \times 3 \times 3$ kernel and removing the max-pooling and last fully connected layer. Furthermore, to better accommodate for the non-i.i.d. issues in the federated learning scenario (Hsieh et al., 2020) we replace batch normalization (Ioffe & Szegedy, 2015) with group normalization (Wu & He, 2018). For the client ID projector, we use a simple MLP on top of the encoder output with a single ReLU hidden layer of 2048 units and 128 output units. For the auxiliary classifier in the case of semi-supervised learning we use a simple linear layer on top of the encoder output.

For our SimCLR and spectral constrastive learning variants, the representations of the encoder are passed through an MLP projector with a single hidden layer of 2048 units and 128 dimensional outputs. The contrastive loss between the two views is measured at the output of the projector.

For our SimSiam baseline we measure the cosine similarity objective on the output of a projector that follows the SimCLR design with the exception that we also add a group normalization layer before the hidden layer, as SimSiam was unstable without it (especially at the unsupervised experiments). For the predictor we use another single hidden layer MLP with 2048 ReLU units and group normalization.

For the data augmentations, in order to create the two views, we follow the standard recipe of random cropping into 32×32 images, followed by a random horizontal flip, a random, with probability 0.8, color distortion with brightness, contrast, saturation factors of 0.4 and a hue factor of 0.1. The final augmentation is a random, with probability 0.2, RGB-to-grayscale transformation.

Optimization details For local optimization we use standard stochastic gradient descent with a learning rate of 0.1 for both CIFAR 10 and CIFAR 100 for, unless mentioned otherwise, a single local epoch and a batch size of 128. After the local optimization on a specific round has been completed, each client communicates to the server the delta between the finetuned parameters and the model communicated from the server to the clients. The server averages these deltas, interprets them as "gradients", and uses them in conjunction with the Adam Kingma & Ba (2014) optimizer in order to update the global model. This is a strategy originally proposed in Reddi et al. (2020). For the server-side Adam we are using the default hyperparameters.

B ADDITIONAL EXPERIMENTS

In this section we consider more baselines for both our unsupervised and semi-supervised setups in the federated setting.

B.1 UNSUPERVISED SETTING

Additional baseline We consider one more baseline for self-supervised learning in the federated setting, FeatARC (Wang et al., 2022), specifically the "Align Only" variant. We omit the clustering approach as it makes additional assumptions compared to our unsupervised learning setup. The

authors report results with a loss-coefficient of $\lambda = 1.0$, which lead to loss divergence in our case, so we report $\lambda = 0.5$, which was stable except for the covariate shift setting. We see in table 3 that adding FeatARC alignment regularization does not result in improved accuracy, contrary to what the FeatARC paper results would lead us to expect. We hypothesise that this is due to the differences in our setup. Whereas FeatARC considers a cross-silo setting with a large number of local update steps, our setting focuses on the cross-device setting with one local epoch per client communication round. We leave a further analysis of FeatARC applicability to this cross-device setting to future work.

Table 3: Test set performance on the unsupervised setting of CIFAR 10. Clients' data is assumed to be fully annotated for LP fine-tuning in the unsupervised case.

Method	Label skew	Covariate shift	Joint shift
Local SimCLR	$79.4_{\pm 0.2}$	$\mathbf{74.3_{\pm 0.3}}$	$71.0_{\pm 0.4}$
Local SimCLR + FeatARC	$70.4_{\pm 0.2}$	$34.4_{\pm -}$	$57.6_{\pm 2.7}$
Federated SimCLR	$\mathbf{85.0_{\pm 0.2}}$	$73.8_{\pm 0.2}$	$\mathbf{74.8_{\pm 0.5}}$
Spectral CL	$76.5_{\pm 1.1}$	$\mathbf{73.5_{\pm 0.4}}$	$68.2_{\pm 0.6}$
Spectral CL + UV	$\mathbf{87.8_{\pm 0.3}}$	$71.7_{\pm 0.5}$	$\mathbf{76.6_{\pm 0.6}}$
SimSiam	$\mathbf{40.0_{\pm 0.5}}$	$\mathbf{39.9_{\pm 0.3}}$	$\mathbf{39.6_{\pm 0.3}}$
SimSiam + UV	$35.4_{\pm 0.4}$	$35.4_{\pm 0.2}$	$34.5_{\pm 0.3}$
Supervised	$89.6_{\pm 0.1}$	$78.3_{\pm 0.4}$	$76.3_{\pm 1.1}$

TinyImagenet dataset To demonstrate the scalability of our theoretical results and model design stemming from our MI perspective, we also consider the more challenging task of self-supervised pretraining on TinyImagenet. It consists of 100k training examples and 10k test examples, each beloging to one of 200 classes. We apply our federated CIFAR 10 setting to this dataset as well, i.e., we partition the training dataset to 100 clients with either the covariate shift or joint shift non-i.i.d. strategies. We sample 10 clients per round in order to optimize the models and each client performs one local epoch of updates. The encoder mdoel we use is a Compact Convolutional Transformer Hassani et al. (2021) in the "CCT-4/3×2" variant, i.e. with 4 transformer encoder layers and a 2-layer convolutional feature extractor with a 3x3 kernel size. The results with the different methods can be seen at table 4.

Table 4: Test set performance (%) on the unsupervised setting of TinyImagenet with 100 clients after 50k rounds. Clients' data is assumed to be fully annotated for LP fine-tuning in the unsupervised case.

Method	Label skew	Covariate shift	Joint shift
Local SimCLR	33.3	**30.3**	29.6
Federated SimCLR	**38.0**	30.0	**31.6**
Spectral CL	34.0	28.4	27.9
Spectral CL + UV	**39.7**	**29.5**	**32.4**
SimSiam	**10.6**	**4.7**	0.5
SimSiam + UV	0.5	0.5	0.5
Supervised	44	36.6	33.0

Overall, we see that the results are consistent with our intuitions and story in the case of contrastive methods; the biggest gains from the additional UV loss are in the case of label skew and joint shift. SimSiam generally underperformed in this setting, which is also consistent with our observations in the case of unsupervised learning on CIFAR 10/100, probably due to representation collapse, given that in our setting we use group normalization instead of batch normalization.

B.2 SEMI-SUPERVISED SETTING

Additional pseudo-labelling baselines We provide more results on our partially labeled (with 10% labeled data on each client) semi-supervised setting by also considering baselines that perform pseudo-labelling as a means for semi-supervised learning. The two methods we consider are SemiFed (Lin et al., 2021) and CBAFed (Li et al., 2023b). For both of these settings we have the following modifications that bring them in line with our semi-supervised setup.

For SemiFed we do not make use of an ensemble of client models in order to impute the missing labels but rather assign a pseudo-label to the datapoint based on the received server model on each client. In this way, our proposed methods and SemiFed have similar communication costs and privacy, as exchanging models directly trained on local data between clients reduces the overall privacy. For CBAFed, we do not use residual weight connection, in order to have a consistent optimization strategy for all our methods, but do use the class balanced adaptive threshold strategy. We follow the setup described in Appendix F.5 of (Li et al., 2023b) to train a model with partially labeled clients.

From what we can see it table 5 and table 6, our conclusion about the usefulness of the UV loss (c.f. proposition 2) applies to this setting as well. While SemiFed underperforms when trained without the UV loss, it manages to improve upon the fully supervised baseline and be comparable to the other methods when we add it back. On CIFAR 10, adding the UV loss yields a significant 16.7% improvement in the case of label skew and on CIFAR 100, while it gets a more modest 6% improvement, it manages to outperform all other methods. CBAFed performs worse than self-supervised methods albeit also benefits from adding the UV loss in all the conducted experiments.

Table 5: Test set performance ($\%$) on the semi-supervised setting of CIFAR 10 with 10% labelled data on each client along with standard error over 5 seeds for all experiments except of CBAFed which have one seed only. We use the corresponding labelled subset for the LP.

Method	Label skew	Covariate shift	Joint shift
Local SimCLR	$74.5_{\pm 0.3}$	$\mathbf{49.1_{\pm 1.3}}$	$45.8_{\pm 1.4}$
Federated SimCLR	$\mathbf{78.0_{\pm 0.2}}$	$\mathbf{50.3_{\pm 1.1}}$	$\mathbf{49.9_{\pm 1.4}}$
Spectral CL	$74.2_{\pm 0.3}$	$48.0_{\pm 0.7}$	$45.4_{\pm 1.5}$
Spectral CL + UV	$\mathbf{79.6_{\pm 0.3}}$	$\mathbf{49.7_{\pm 1.0}}$	$\mathbf{49.8_{\pm 1.1}}$
SimSiam	$75.3_{\pm 0.4}$	$46.8_{\pm 0.7}$	$40.5_{\pm 0.9}$
SimSiam + UV	$\mathbf{80.4_{\pm 0.2}}$	$\mathbf{50.0_{\pm 1.2}}$	$\mathbf{44.3_{\pm 1.0}}$
SemiFed	$60.0_{\pm 4.5}$	$18.6_{\pm 1.8}$	$37.2_{\pm 0.9}$
SemiFed + UV	$\mathbf{76.7_{\pm 1.2}}$	$\mathbf{24.0_{\pm 2.2}}$	$\mathbf{45.1_{\pm 2.0}}$
CBAFed	66.3	45.9	34.8
CBAFed + UV	**74.1**	**48.2**	**36.2**
Supervised	$75.1_{\pm 0.2}$	$48.1_{\pm 0.9}$	$42.7_{\pm 1.7}$

TinyImagenet dataset To demonstrate the scalability of our semi-supervised model design stemming from our MI perspective, we also consider the more challenging TinyImagenet task in the case of label skew non-i.i.d.-ness with Dirichlet splitting and an $\alpha = 0.1$ multiplied by the prior probability of each class. The setup is similar to our semi-supervised federated CIFAR 10 setting, with 100 clients and 10% labelled data per client. We sample 10 clients per round in order to optimize the models and each client performs one local epoch of updates. We use the same CCT architecture as the unsupervised TinyImagenet experiment. The results with the different methods can be seen in table 7.

We observe similar patterns to our unsupervised TinyImagenet setting, with the biggest gains for the contrastive methods from the UV loss being in the case where some label skew is present. SimSiam did experience representation collapse at the case of label skew, however, by adding to it the UV loss, this was successfully mitigated and improved significantly the performance.

Table 6: Test set performance (%) on the semi-supervised setting of CIFAR 100 with 10% labelled data on each client along with standard error over 5 seeds. We use the corresponding labelled subset for the LP.

Method	Label Skew	Covariate shift	Joint shift
Local SimCLR	$30.3_{\pm 0.2}$	$15.1_{\pm 0.4}$	$13.1_{\pm 0.3}$
Federated SimCLR	$\mathbf{34.5_{\pm 0.3}}$	$14.8_{\pm 0.3}$	$\mathbf{14.6_{\pm 0.3}}$
Spectral CL	$30.1_{\pm 0.2}$	$14.1_{\pm 0.4}$	$12.3_{\pm 0.3}$
Spectral CL + UV	$\mathbf{34.0_{\pm 0.2}}$	$13.7_{\pm 0.3}$	$\mathbf{13.6_{\pm 0.4}}$
SimSiam	$30.7_{\pm 0.2}$	$13.4_{\pm 0.3}$	$12.8_{\pm 0.3}$
SimSiam + UV	$\mathbf{34.3_{\pm 0.1}}$	$13.6_{\pm 0.3}$	$\mathbf{14.0_{\pm 0.4}}$
SemiFed	$29.7_{\pm 0.5}$	$13.3_{\pm 0.2}$	$12.3_{\pm 0.2}$
SemiFed + UV	$\mathbf{35.7_{\pm 0.2}}$	$13.4_{\pm 0.6}$	$\mathbf{13.1_{\pm 0.2}}$
Supervised	$29.6_{\pm 0.3}$	$12.6_{\pm 0.2}$	$12.2_{\pm 0.1}$

Table 7: Test set performance (%) on the semi-supervised setting of TinyImagenet with 100 clients after 50k rounds. We use the corresponding labelled subset for the linear probe.

Method	Label skew	Covariate shift	Joint shift
Local SimCLR	18.5	8.1	6.7
Federated SimCLR	**19.5**	**8.4**	**7.4**
Spectral CL	17.8	**8.3**	6.9
Spectral CL + UV	**18.9**	8.1	**7.5**
SimSiam	0.5	8.1	**6.9**
SimSiam + UV	**20.0**	**8.5**	**6.9**
Supervised	17.9	8.4	7.7

C ALGORITHMS

Algorithm 1 The server side algorithm for our federated SimCLR / Spectral CL / SimSiam with optional user-verification and semi-supervision.

Initialize θ and ϕ with θ_1, ϕ_i
for round t in $1, \dots T$ **do**
  Sample $\mathcal{S}$ clients from the population
  Initialize $\nabla_\theta^t = \mathbf{0}, \nabla_\phi^t = \mathbf{0}$
  for s in $\mathcal{S}$ **do**
    $\theta_s, \phi_s \leftarrow$ CLIENT(s, θ_t, ϕ_t)
    $\nabla_\theta^t += \frac{\theta_t - \theta_s}{|\mathcal{S}|}$
    $\nabla_\phi^t += \frac{\phi_t - \phi_s}{|\mathcal{S}|}$
  end for
  $\theta^{t+1}, \phi^{t+1} \leftarrow$ ADAM$(\nabla_\theta^t, \nabla_\phi^t)$
end for

Algorithm 2 The client side algorithm for our federated SimCLR / Spectral CL / SimSiam with optional user-verification and semi-supervision. L_{ul} corresponds to the unsupervised loss component of SimCLR / Spectral CL / SimSiam. β is a coefficient that determines the weight of the UV loss, with a default value of 1.

Get θ, ϕ from the server
$\theta_s, \phi_s \leftarrow \theta, \phi$
for epoch e in $1, \ldots, E$ **do**
  for batch $b \in B$ **do**
    ▷ Unlabelled and labelled datapoints of the batch b
    $x_{ul}, (x_l, y_l) \leftarrow b$
    ▷ Get the two views through augmentations
    $[x^1_{ul}, x^1_l], [x^2_{ul}, x^2_l] = \text{AUG}([x_{ul}, x_l]), \text{AUG}([x_{ul}, x_l])$
    ▷ Representations of the two views from the encoder f with parameters θ_s
    $[z^1_{ul}, z^1_l], [z^2_{ul}, z^2_l] \leftarrow f([x^1_{ul}, x^1_l]; \theta_s), f([x^2_{ul}, x^2_l]; \theta_s)$
    ▷ Unsupervised loss with, depending on β, an additional UV loss
    $\mathcal{L}_s = \mathcal{L}_{ul}(z^1_{ul}, z^2_{ul}; \phi_s) + \beta \mathcal{L}_{uv}(s, z^1_{ul}, z^2_{ul}; \phi_s)$
    ▷ Supervised loss on the labelled data
    for label $i \in \{0, \ldots, |Y| - 1\}$ **do**
      ▷ Unsupervised loss between datapoints of the same class
      $\mathcal{L}_s += \mathcal{L}_{ul}(z^1_l[y_l == i], z^2_l[y_l == i]; \phi_s)$
    end for
    ▷ Standard supervised loss
    $\mathcal{L}_s += \mathcal{L}_y(y_l, z^1_l, z^2_l; \phi_s)$
    ▷ Local gradient updates on the loss
    $\theta_s, \phi_s \leftarrow \text{SGD}(\nabla_{\theta_s, \phi_s} L_s)$
  end for
end for
return θ_s, ϕ_s

D MISSING PROOFS

Proposition 1. *Let $s \in \mathbb{N}$ denote the user ID, $\mathbf{x} \in \mathbb{R}^{D_x}$ the input and $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{D_z}$ the latent representations of the two views of $\mathbf{x}$ given by the encoder with parameters θ. Given a critic function $f : \mathbb{R}^{D_z} \times \mathbb{R}^{D_z} \to \mathbb{R}$, we have that*

$$\mathrm{I}_\theta(\mathbf{z}_1; \mathbf{z}_2 | s) \geq \mathbb{E}_{p(s) p_\theta(\mathbf{z}_1, \mathbf{z}_2 | s)_{1:K}} \left[\frac{1}{K} \sum_{k=1}^{K} \log \frac{\exp(f(\mathbf{z}_{1k}, \mathbf{z}_{2k}))}{\frac{1}{K} \sum_{j=1}^{K} \exp(f(\mathbf{z}_{1j}, \mathbf{z}_{2k}))} \right]. \tag{14}$$

Proof. The proof follows Poole et al. (2019). We can show that

$$\mathrm{I}_\theta(\mathbf{z}_1;\mathbf{z}_2|s) = \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1,1},\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,2:K}|s)}\left[\log \frac{p_\theta(\mathbf{z}_{1,1}|\mathbf{z}_2,s)p_\theta(\mathbf{z}_{1,2:K}|s)}{p_\theta(\mathbf{z}_{1,2:K}|s)p_\theta(\mathbf{z}_{1,1}|s)}\right] \tag{15}$$

$$= \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1,1:K},\mathbf{z}_2|s)}\left[\log \frac{p_\theta(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}{p_\theta(\mathbf{z}_{1,1:K}|s)}\right] \tag{16}$$

$$= \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1,1:K},\mathbf{z}_2|s)}\left[\log \frac{p_\theta(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}{q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)p_\theta(\mathbf{z}_{1,1:K}|s)}\right] \tag{17}$$

$$= \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1,1:K},\mathbf{z}_2|s)}\left[\log \frac{q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}{p_\theta(\mathbf{z}_{1,1:K}|s)}\right] + \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}\left[\log \frac{p_\theta(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}{q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}\right] \tag{18}$$

$$= \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1,1:K},\mathbf{z}_2|s)}\left[\log \frac{q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}{p_\theta(\mathbf{z}_{1,1:K}|s)}\right] + \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)}\left[D_{\mathrm{KL}}(p_\theta(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)||q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s))\right] \tag{19}$$

$$\geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1,1:K},\mathbf{z}_2|s)}\left[\log \frac{q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)}{p_\theta(\mathbf{z}_{1,1:K}|s)}\right], \tag{20}$$

and then by parametrizing $q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s)$ in terms of a critic function f,

$$q(\mathbf{z}_{1,1:K}|\mathbf{z}_2,s) = \frac{p_\theta(\mathbf{z}_{1,1:K}|s)\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))}{\mathbb{E}_{p_\theta(\mathbf{z}_{1,1:K}|s)}[\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))]}, \tag{21}$$

we have that

$$\mathrm{I}_\theta(\mathbf{z}_1;\mathbf{z}_2|s) \geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1,1:K},\mathbf{z}_2|s)}\left[\log \frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))}{\mathbb{E}_{p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))\right]}\right]. \tag{22}$$

Since the denominator depends on the aggregate score $\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))$ over $p_\theta(\mathbf{z}_{1,1:K}|s)$, which is similarly intractable, we can introduce one more lower bound that will allow us to work with minibatches of data Poole et al. (2019). Due to the positivity of the exponent, we have that for any $a > 0$

$$\log \mathbb{E}_{p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))\right] \leq \frac{\mathbb{E}_{p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))\right]}{a} + \log a - 1. \tag{23}$$

Using this bound with $\alpha = \exp(1)$, we have that

$$\mathrm{I}_\theta(\mathbf{z}_1;\mathbf{z}_2|s) \geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_{1:K},\mathbf{z}_2|s)}\left[\log\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))\right] - \exp(-1)\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1:K}))\right]. \tag{24}$$

We can now set $f(\mathbf{z}_2,\mathbf{z}_{1,1:K})$ as Poole et al. (2019)

$$f(\mathbf{z}_2,\mathbf{z}_{1,1:K}) \to 1 + f(\mathbf{z}_2,\mathbf{z}_{1,1}) - \log a(\mathbf{z}_2,\mathbf{z}_{1,1:K}). \tag{25}$$

In this way, we end up with

$$\mathrm{I}_\theta(\mathbf{z}_1;\mathbf{z}_2|s) \geq 1 + \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2,\mathbf{z}_{1,1:K}|s)}\left[\log \frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right] - \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right]. \tag{26}$$

We can now average the bound over K replicates and reindex $\mathbf{z}_1$ as

$$\begin{aligned}
\mathrm{I}_\theta(\mathbf{z}_1;\mathbf{z}_2|s) \geq 1 + \frac{1}{K}\sum_{k=1}^{K}\Bigg(&\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2,\mathbf{z}_{1,1:K}|s)}\left[\log\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right] \\
&- \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right]\Bigg) \qquad (27)\\
= 1 + \frac{1}{K}\sum_{k=1}^{K}&\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2,\mathbf{z}_{1,1:K}|s)}\left[\log\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right] \\
&- \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,1}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right] \qquad (28)\\
= 1 + \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2,\mathbf{z}_{1,1:K}|s)}&\left[\frac{1}{K}\sum_{k=1}^{K}\log\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,k}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right] \\
&- \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,k}))}{a(\mathbf{z}_2,\mathbf{z}_{1,1:K})}\right] \qquad (29)
\end{aligned}$$

and for the specific choice of $a(\mathbf{z}_2,\mathbf{z}_{1,1:K}) = \frac{1}{K}\sum_{k=1}^{K}\exp(f(\mathbf{z}_2,\mathbf{z}_{1,k}))$, we have that terms cancel, i.e.,

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\frac{\exp(f(\mathbf{z}_2,\mathbf{z}_{1,k}))}{\frac{1}{K}\sum_{k=1}^{K}\exp(f(\mathbf{z}_2,\mathbf{z}_{1,k}))}\right] \\
&= \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)p_\theta(\mathbf{z}_{1,1:K}|s)}\left[\frac{\frac{1}{K}\sum_{k=1}^{K}\exp(f(\mathbf{z}_2,\mathbf{z}_{1,k}))}{\frac{1}{K}\sum_{k=1}^{K}\exp(f(\mathbf{z}_2,\mathbf{z}_{1,k}))}\right] = 1. \qquad (30)
\end{aligned}$$

In this way, we end up with the well known InfoNCE loss Oord et al. (2018), where now we contrast between datapoints that share the same class

$$\mathrm{I}_\theta(\mathbf{z}_1;\mathbf{z}_2|s) \geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1,\mathbf{z}_2|s)_{1:K}}\left[\frac{1}{K}\sum_{k=1}^{K}\log\frac{\exp(f(\mathbf{z}_{1k},\mathbf{z}_{2k}))}{\frac{1}{K}\sum_{j=1}^{K}\exp(f(\mathbf{z}_{1j},\mathbf{z}_{2k}))}\right]. \qquad (31)$$

□

Lemma 2.1. *Let $s \in \mathbb{N}$ denote the client ID, $\mathbf{x} \in \mathbb{R}^{D_x}$ the input and $\mathbf{z}_1 \in \mathbb{R}^{D_z}$ the latent representation of a view of $\mathbf{x}$ given by the encoder with parameters θ. Let ϕ denote the parameters of a client classifier $r_\phi(s|\mathbf{z}_1)$ that predicts the client ID from this specific representation and let $\mathrm{H}(s)$ be the entropy of the client distribution $p(s)$. We have that*

$$\mathrm{I}_\theta(\mathbf{z}_1;s) \geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1|s)}\left[\log r_\phi(s|\mathbf{z}_1)\right] + \mathrm{H}(s) \qquad (32)$$

Proof.

$$\begin{aligned}
\mathrm{I}_\theta(\mathbf{z}_1;s) &= \mathbb{E}_{p_\theta(s,\mathbf{z}_1)}\left[\log\frac{p_\theta(s,\mathbf{z}_1)}{p(s)p_\theta(\mathbf{z}_1)}\right] = \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1|s)}\left[\log\frac{p_\theta(s|\mathbf{z}_1)}{p(s)}\right] \qquad (33)\\
&= \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1|s)}\left[\log\frac{r_\phi(s|\mathbf{z}_1)}{p(s)}\right] + \mathbb{E}_{p(s)}\left[D_{\mathrm{KL}}(p_\theta(s|\mathbf{z}_1)||r_\phi(s|\mathbf{z}_1))\right] \qquad (34)\\
&\geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1|s)}\left[\log\frac{r_\phi(s|\mathbf{z}_1)}{p(s)}\right] = \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1|s)}\left[\log r_\phi(s|\mathbf{z}_1)\right] + \mathrm{H}(s). \qquad (35)
\end{aligned}$$

□

Lemma 2.2. *Let $s \in \mathbb{N}$ denote the user ID, $\mathbf{x} \in \mathbb{R}^{D_x}$ the input and $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{D_z}$ the latent representations of the views of $\mathbf{x}$ given by the encoder with parameters θ. Let ϕ denote the parameters of a client classifier $r_\phi(s|\mathbf{z}_2)$ that predicts the client ID from the representations. We have that*

$$\mathrm{I}_\theta(\mathbf{z}_1;s|\mathbf{z}_2) \leq -\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)}\left[\log r_\phi(s|\mathbf{z}_2)\right] \qquad (36)$$

Proof.

$$\mathrm{I}_\theta(\mathbf{z}_1; s|\mathbf{z}_2) = \mathrm{H}_\theta(s|\mathbf{z}_2) - \mathrm{H}_\theta(s|\mathbf{z}_2, \mathbf{z}_1) \tag{37}$$

$$\leq \mathrm{H}_\theta(s|\mathbf{z}_2) = \mathrm{H}(s) - \mathrm{I}_\theta(\mathbf{z}_2; s) \leq -\mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)}\left[\log r_\phi(s|\mathbf{z}_2)\right] \tag{38}$$

where $\mathrm{H}(s)$ is the entropy of $p(s)$, $\mathrm{H}_\theta(s|\mathbf{z}_2)$, $\mathrm{H}_\theta(s|\mathbf{z}_2, \mathbf{z}_1)$ are the conditional entropies of s given $\mathbf{z}_2$ and $\mathbf{z}_2, \mathbf{z}_1$ and the last inequality is due to the lower bound of lemma 2.1. We also used the fact that the entropy of a discrete distribution is non-negative. □

Proposition 2. *Consider the label skew data-generating process for federated SimCLR from Figure 1 with $s \in \mathbb{N}$ denoting the user ID with $\mathrm{H}(s)$ being the entropy of $p(s)$, $\mathbf{x} \in \mathbb{R}^{D_x}$ the input, $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{D_z}$ the latent representations of the two views of $\mathbf{x}$ given by the encoder with parameters θ. Let y be the label and let $r_\phi(s|\mathbf{z}_i)$ be a model with parameters ϕ that predicts the user ID from the latent representation $\mathbf{z}_i$. In this case, we have that*

$$\mathrm{I}_\theta(\mathbf{z}_1; y) + \mathrm{I}_\theta(\mathbf{z}_2; y) \geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1, \mathbf{z}_2|s)}\left[\log r_\phi(s|\mathbf{z}_1) + \log r_\phi(s|\mathbf{z}_2)\right] + 2\mathrm{H}(s). \tag{39}$$

Proof. The claim is a consequence of the data processing inequality. We start by noting that

$$\mathrm{I}_\theta(\mathbf{z}_1; y) + \mathrm{I}_\theta(\mathbf{z}_1; s|y) = \mathrm{I}_\theta(\mathbf{z}_1; y, s) = \mathrm{I}_\theta(\mathbf{z}_1; s) + \mathrm{I}_\theta(\mathbf{z}_1; y|s) \tag{40}$$

and since in this graphical model we have that $s \perp\!\!\!\perp \mathbf{z}_1|y$, so $\mathrm{I}_\theta(s; \mathbf{z}_1|y) = 0$, we end up with

$$\mathrm{I}_\theta(\mathbf{z}_1; y) = \mathrm{I}_\theta(\mathbf{z}_1; s) + \mathrm{I}_\theta(\mathbf{z}_1; y|s) \tag{41}$$

$$\geq \mathrm{I}_\theta(\mathbf{z}_1; s) \geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_1|s)}\left[\log r_\phi(s|\mathbf{z}_1)\right] + \mathrm{H}(s), \tag{42}$$

where we use the positivity of mutual information and our lemma 2.1. In a similar manner we can also show that

$$\mathrm{I}_\theta(\mathbf{z}_2; y) \geq \mathbb{E}_{p(s)p_\theta(\mathbf{z}_2|s)}\left[\log r_\phi(s|\mathbf{z}_2)\right] + \mathrm{H}(s). \tag{43}$$

By adding up eq. (42) and eq. (43) we arrive at the claim. □