We propose and study FlowGMM, a new classification model based on normalizing flows that can be naturally applied to semi-supervised learning. The idea of FlowGMM is to map each data class to a component in the Gaussian mixture using an invertible transformation. For semi-supervised learning:

- Labeled data from class \(i\) is modeled as transformation of the \(i\)-th Gaussian
- Unlabeled data is modeled as transformation of the mixture

**FlowGMM**

Define a normalizing flow with a class-conditional latent distribution

\[
p(x|y) = p_Z(f(x)|y) \cdot \frac{\partial f}{\partial z}, \quad p_Z(z|y) = \mathcal{N}(z|\mu_y, \Sigma_y).
\]

We can evaluate likelihood for unlabeled data as

\[
p(x) = \frac{1}{c} \sum_{k=1}^{c} p(x|y=k) = p_Z(f(x)) \cdot \frac{\partial f}{\partial x}, \quad p_Z = \frac{1}{c} \sum_{k=1}^{c} \mathcal{N}(\mu_k, \Sigma_k).
\]

**Consistency Loss Term.** Encourages the model to map small perturbations of the same unlabeled inputs to the same components of the mixture:

\[
L_{\text{Cons}}(x', x'') = \mathcal{N}(f(x')|\mu_y, \Sigma_y),
\]

where \(x'\) and \(x''\) are two perturbations (e.g. random crops) of the same input \(x\), and \(y\) is the class label predicted for \(x\).

**Classification.** Decision rule for a test point \(x\):

\[
y = \arg \max_{i \in \{1,...,c\}} p(x|y) = \arg \max_{i \in \{1,...,c\}} \mathcal{N}(f(x)|\mu_i, \Sigma_i)
\]

**Empirical Results.** Even with a small number of labeled data points, FlowGMM puts the decision boundary to a low-density region in data-space.

**Image Classification.** We use a Multiscale RealNVP architecture.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST (test acc 97.7%)</th>
<th>SVHN</th>
<th>CIFAR-10 (test acc 89.7%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIGLM</td>
<td>97.36</td>
<td>8.58</td>
<td>83.12</td>
</tr>
<tr>
<td>FlowGMM Sup</td>
<td>99.63</td>
<td>95.81</td>
<td>88.44</td>
</tr>
<tr>
<td>FlowGMM Sup w Temp</td>
<td>99.33</td>
<td>95.74</td>
<td>-</td>
</tr>
<tr>
<td>SSLNF Sup</td>
<td>97.79</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SSLNF-cons</td>
<td>98.94</td>
<td>82.42</td>
<td>78.24</td>
</tr>
</tbody>
</table>

**Uncertainty.** FlowGMM produces overconfident predictions on in-domain data; this problem can be remedied by scaling the variance of mixture components after the training is finished.

**Out-of-Domain Detection.** We use the likelihood \(p(x)\) of FlowGMM to identify out-of-domain data.

**Latent Representation.** FlowGMM naturally encodes the *clustering principle*: the decision boundary between classes must lie in the low-density region.

![Figure 2: Bottom] Unlabeled (blue dots) and labeled data (colored triangles) and decision boundary (dashed line). Top: mapping of the data to the latent space.

![Figure 3: Left] Log-likelihoods on in- and out-of-domain data for our model trained on MNIST and Right: FashionMNIST.

- FlowGMM trained on MNIST can identify notMNIST and FashionMNIST data as out-of-domain
- On the other hand, MNIST examples are assigned higher likelihoods by our model trained on FashionMNIST than the training data itself