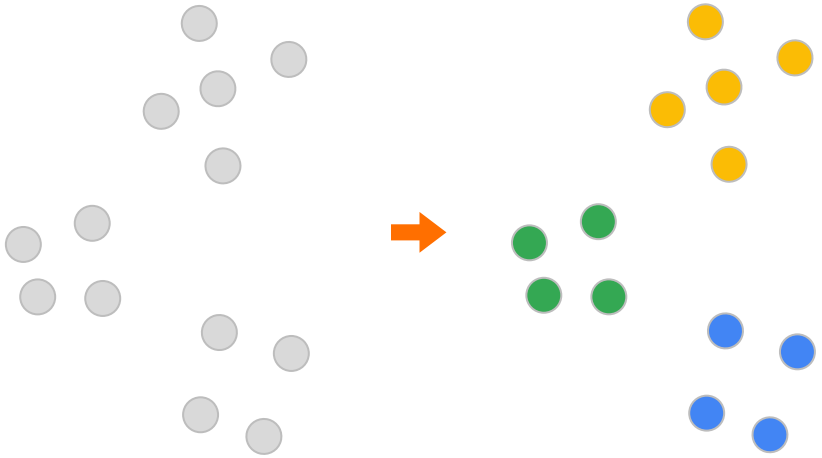
TensorFlow

Learning to Cluster using Neural Clustering Process

Yueqi Wang

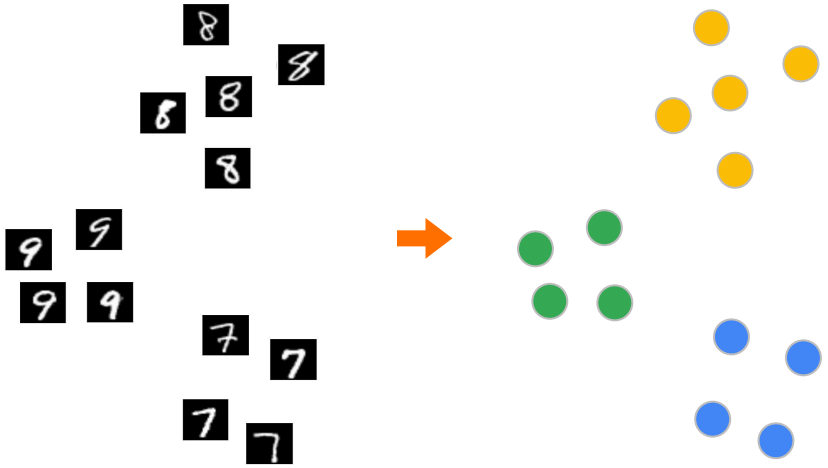
Clustering



Input Data

Clusters

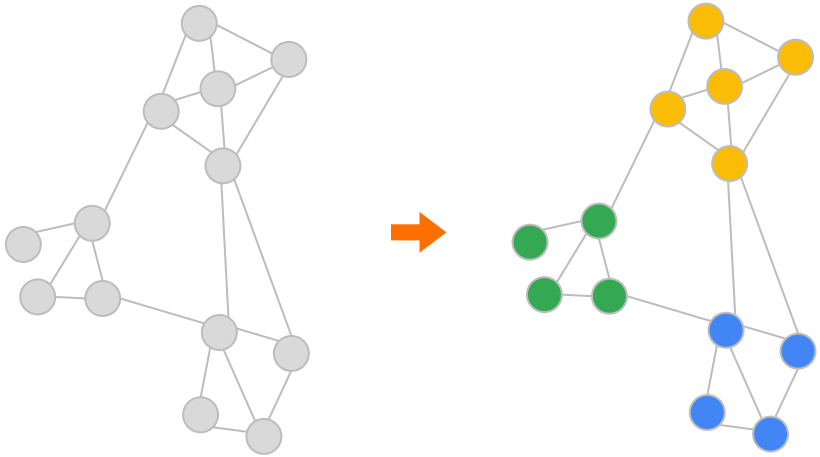
Image Clustering



Images

Clusters

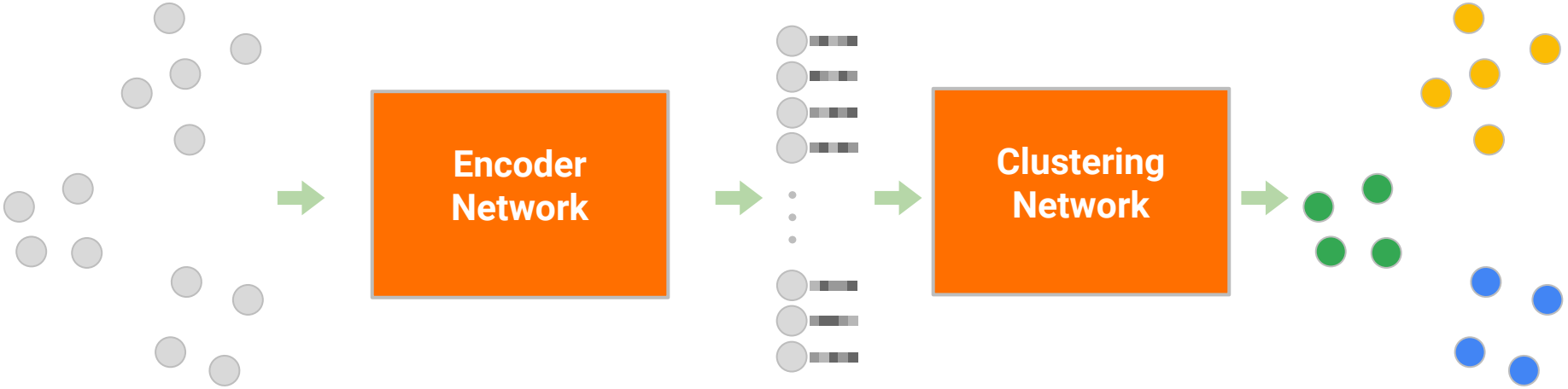
Graph Clustering



Graph Nodes

Clusters

Framework for Neural Clustering

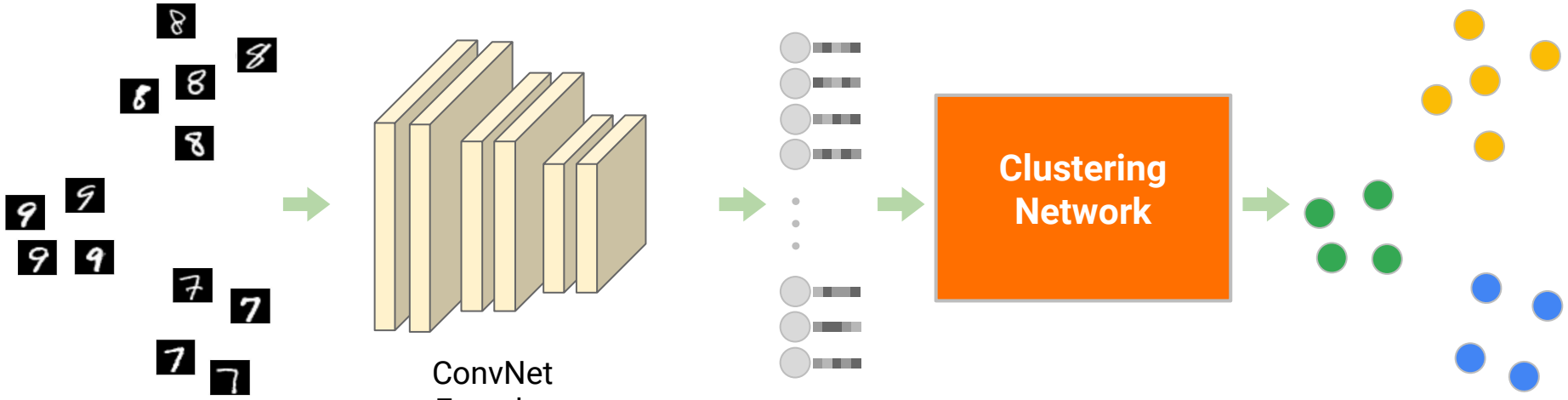


Input Data

Embeddings

Clusters/Communities

Image Clustering



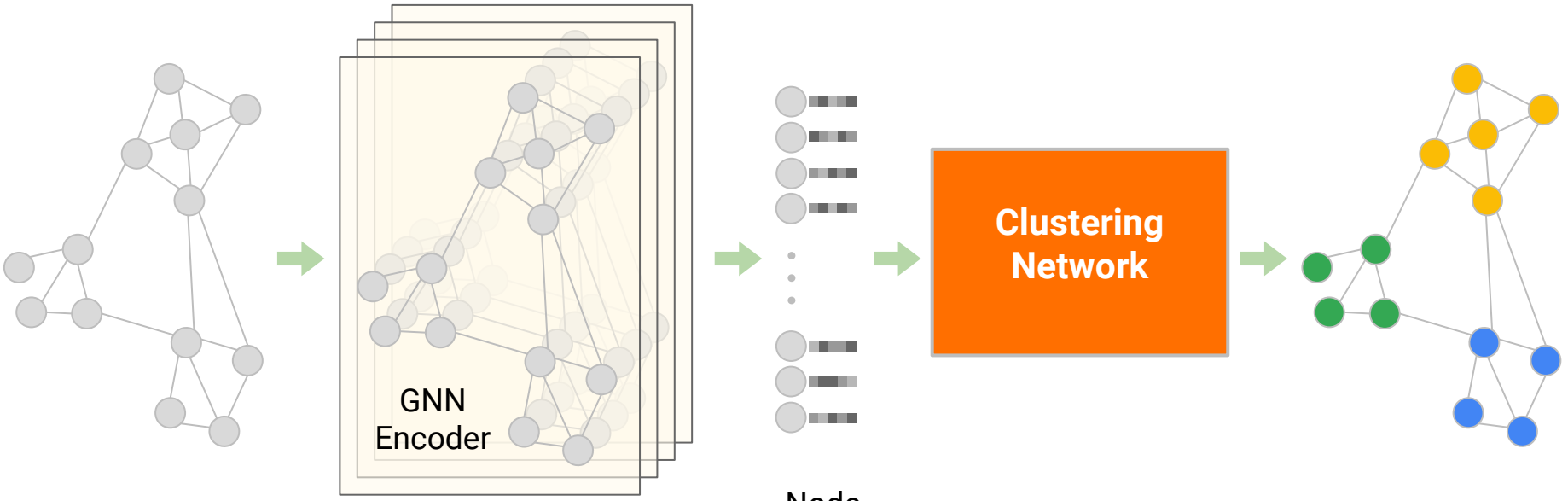
Input Images

ConvNet Encoder

Image Embeddings

Clusters

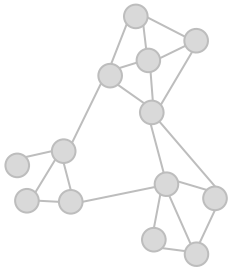
Graph Clustering



How to Cluster the Embeddings

No labeled data

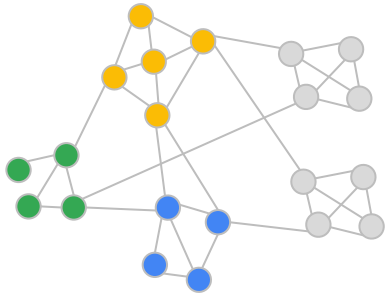
Unsupervised clustering



Not always agree with human labels

Have labeled data

Train a classifier?



unseen labels

Cannot generalize to unseen labels/clusters

Goal: A model that learns the clustering objective from labeled data.

Learning to Cluster using Neural Networks

- Amortized/Supervised Clustering
  - Fast inference on many statistically similar datasets after training
- Encoding prior knowledge into a trained model
  - Specify complex clustering objectives by labeled training data
- End-To-End model to cluster many types of data
  - ConvNet encoder for images, GNN encoder for graphs, etc

Neural Clustering Process (NCP)

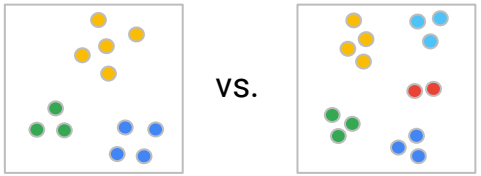
- Handles arbitrary numbers of clusters at test time
- Probabilistic clustering with well-defined posterior

Challenges in Neural Clustering

Challenges:

Unknown number of clusters at test time

Permutation symmetry between cluster labels



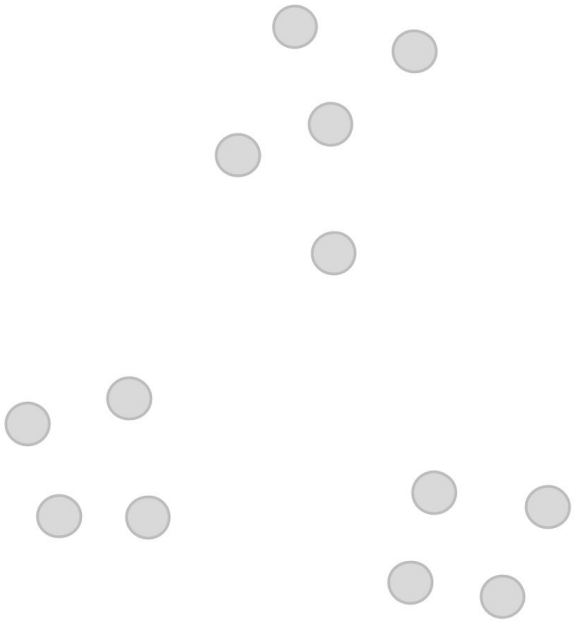
3 clusters 5 clusters

NCP:

Sequential assignment of cluster labels

Permutation-invariant neural representation

Neural Clustering Process (NCP)



- Pick any unlabeled point
- Predict
  - which existing cluster to join
  - or create a new cluster
- Iterate

Pakman, et al. ICML 2020

Neural Clustering Process (NCP)

Pointwise expansion of the posterior

$$p(c_{1:N}|\mathbf{x}) = p(c_1|\mathbf{x})\, p(c_2|c_1, \mathbf{x}) \ldots p(c_N|c_{1:N-1}, \mathbf{x})$$

Each factor $p(c_i|c_{1:i-1}, \mathbf{x}) = \dfrac{p(c_i, c_{1:i-1}, \mathbf{x})}{\sum_{c_i'=1}^{K+1} p(c_i', c_{1:i-1}, \mathbf{x})}$ is learned using a neural network.

K+1 possibilities:

- Join one of the K existing clusters
- Create a new cluster

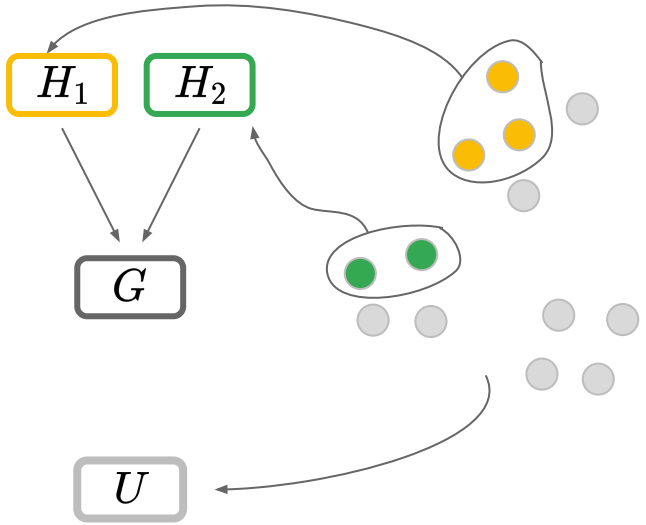
Dynamically grows the number of clusters at test time

Pakman, et al. ICML 2020

Neural Architecture

- Each cluster $H_k = \sum_{i:c_i=k} h(x_i)$
- All existing clusters $G = \sum_{k \in \{1 \ldots K\}} g(H_k)$
- All unassigned data $U = \sum_{i \in I_u} u(x_i)$

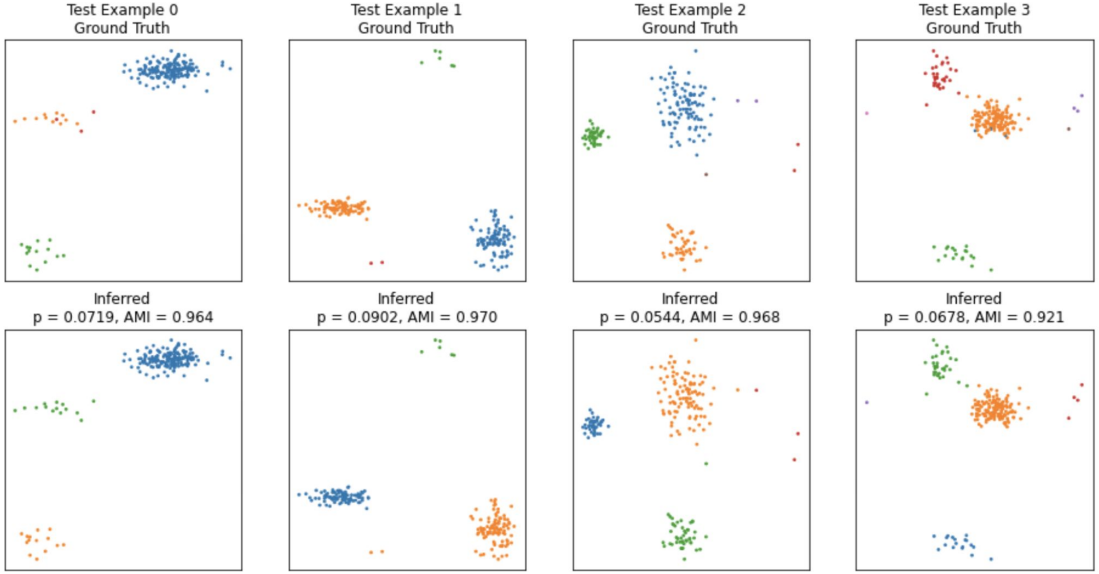
I_u -- indices of unassigned data



(G, U): current clustering configuration

$$p_\theta(c_i = k | c_{1:i-1}, \mathbf{x}) = \frac{e^{f(G_{c_i=k}, U)}}{\sum_{k'=1}^{K+1} e^{f\left(G_{c_i=k'}, U\right)}}$$

Example: Gaussian Mixtures



[Colab notebook]

Probabilistic Formulation for Amortized Clustering

Generative Model (e.g. Gaussian Mixtures)

$$c_1 \dots c_N \sim p(c_1 \dots c_N)$$ Cluster Labels (K distinct)

$$\mu_k \sim p(\mu) \qquad k = 1 \dots K$$ Cluster Parameters

$$x_i \sim p(x_i \mid \mu_{c_i}) \qquad i = 1 \dots N$$ Observations

Inference goal: given observations $\mathbf{x} = \{x_i\}$, produce cluster labels from $p(c_{1:N} \mid \mathbf{x})$

Alternative approaches: MCMC, variational inference

Amortized Inference for Probabilistic Clustering

Learn a neural network function that approximates $p(c_{1:N} \mid \mathbf{x})$