

# FedDAG: Clustered Federated Learning via Global Data and Gradient Integration for Heterogeneous Environments



**Anik Pramanik**

New Jersey Institute of Technology



**Murat Kantarcioglu**

Virginia Tech



**Vincent Oria**

New Jersey Institute of Technology

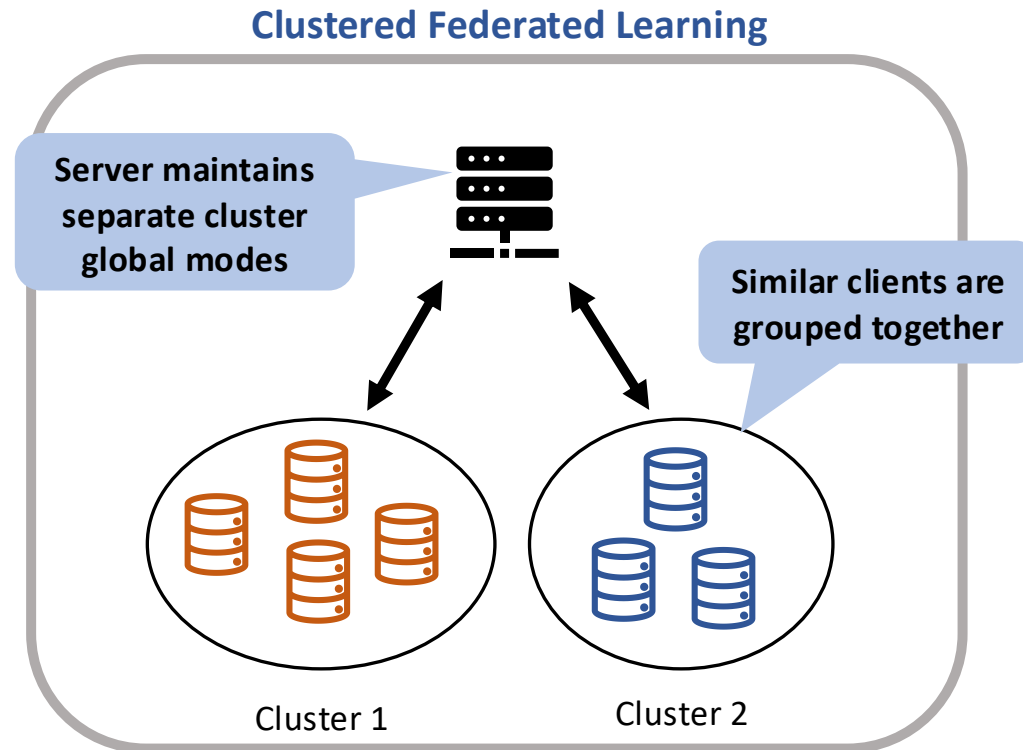


**Shantanu Sharma**

New Jersey Institute of Technology

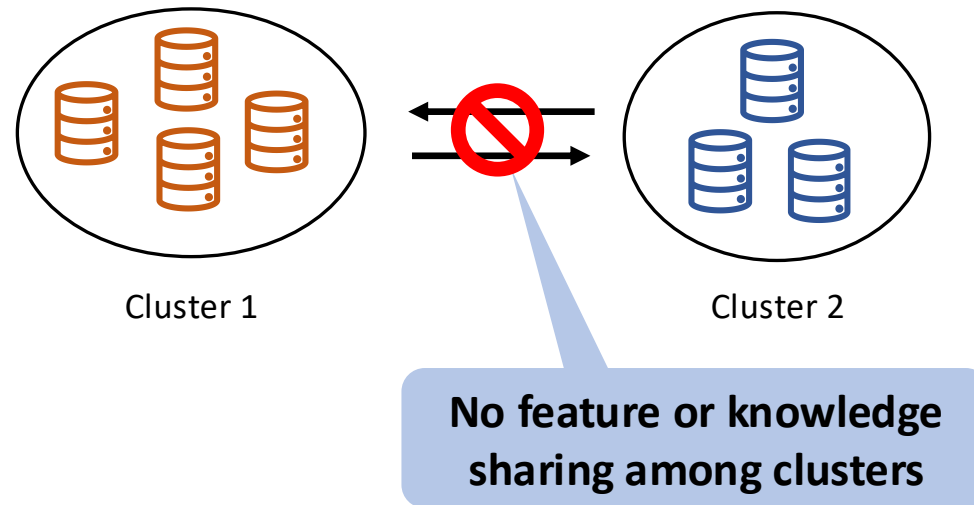
# Clustered Federated Learning on Non-IID Data

- Cluster FL is a prominent approach to tackle non-IID data in FL
- Clustered FL groups clients into clusters based on data distributions
- Each cluster maintains a separate cluster global model for FL training



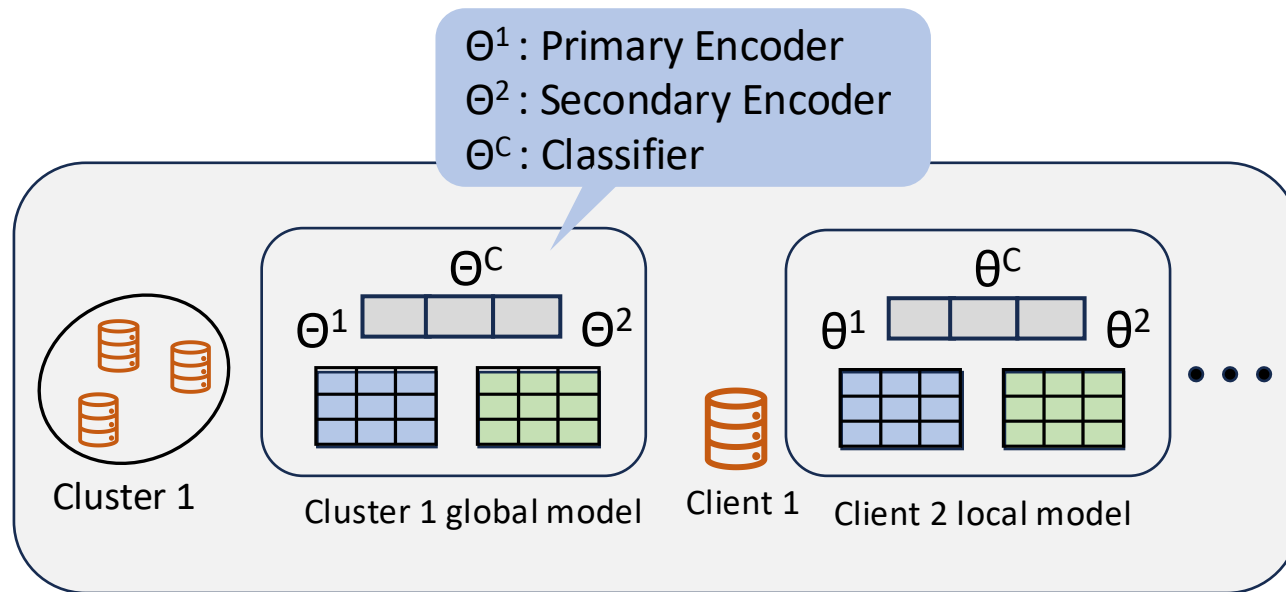
# Existing Approaches: Drawbacks of Cluster FL

- Restricts knowledge sharing and training to clients within the same cluster.
- Prohibits clients in different clusters from benefiting from low-level features.

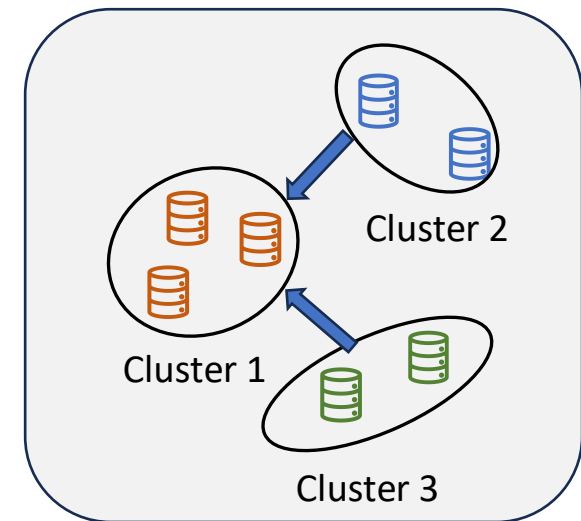


# FedDAG: Feature Sharing Among Clusters

- FedDAG enables feature sharing among clusters via a dual-encoder architecture
- FedDAG models contain a primary encoder, a secondary encoder, and a classifier.
- Primary encoder is trained on its own cluster's data, capturing cluster-specific features.
- Secondary encoder captures complementary representations from other clusters.
- FedDAG also creates a graph to indicate which cluster should learn from which other cluster



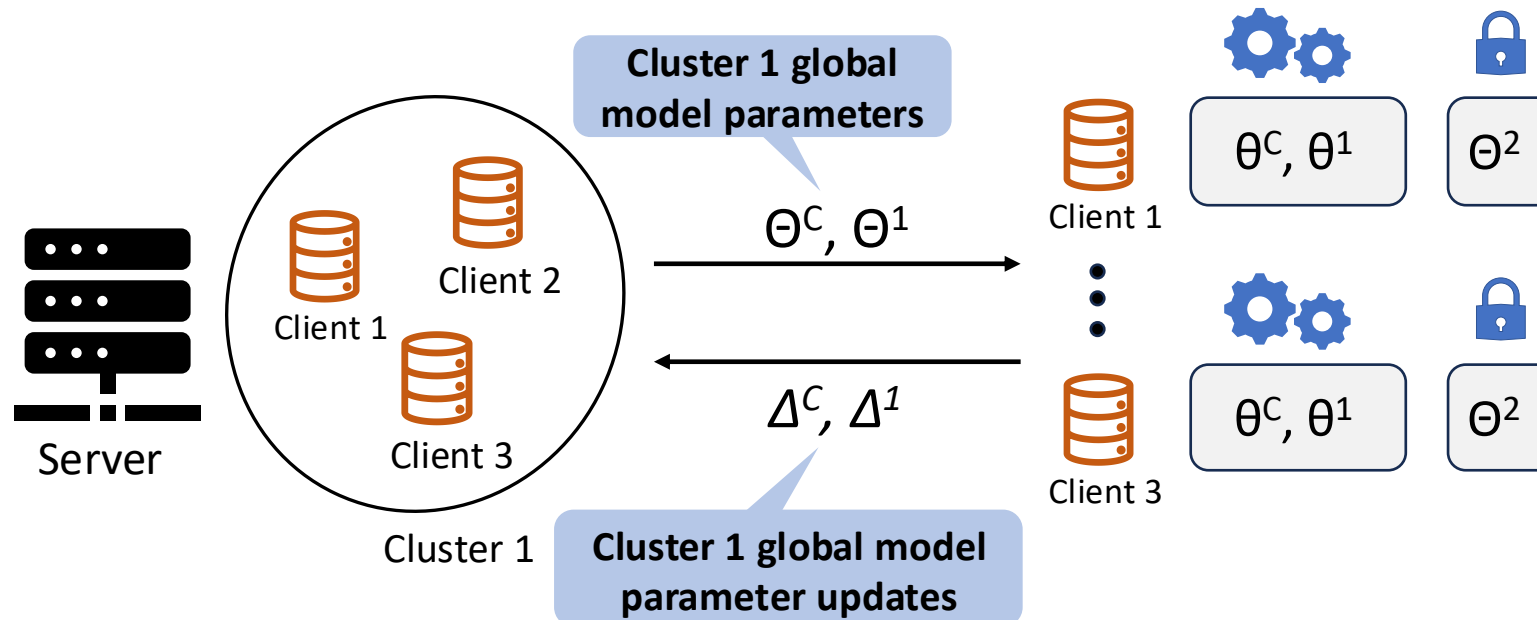
FedDAG Global and Local Model Architecture



CC-Graph

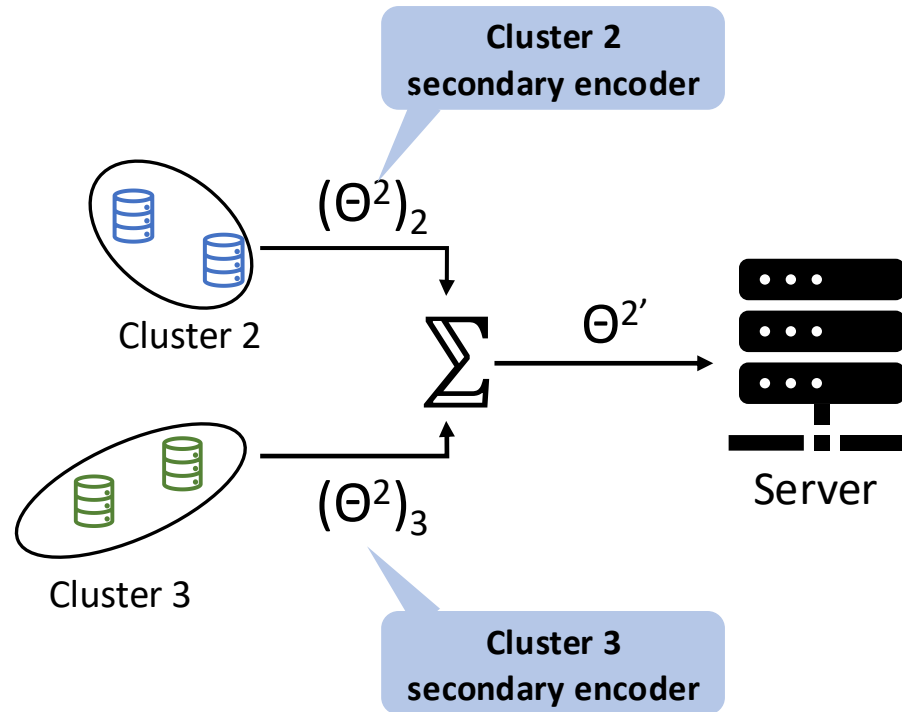
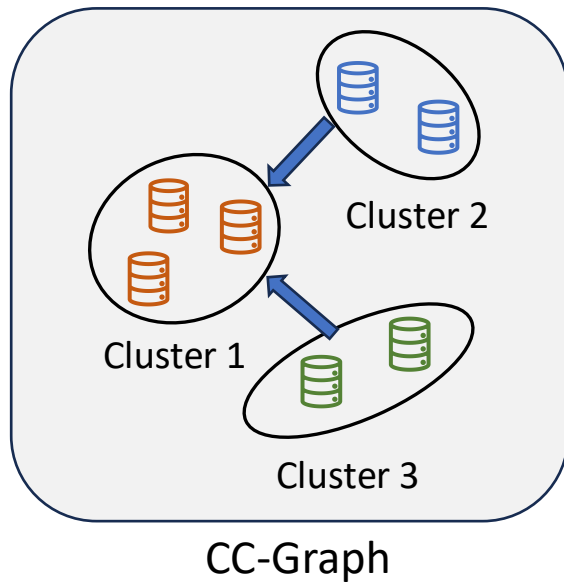
# FedDAG: Feature Sharing Among Clusters

- **Step 1:** FedDAG trains the cluster global model's primary encoder  $\Theta^1$  on its own data
  - Server sends the cluster 1 global model's parameters to its clients
  - Clients freeze the secondary encoder  $\Theta^2$ , and trains the classifier  $\Theta^C$  and the primary encoder  $\Theta^1$  on their own data
  - Clients send back updates  $(\Delta^C, \Delta^1)$  to the server to update cluster 1 global model parameters



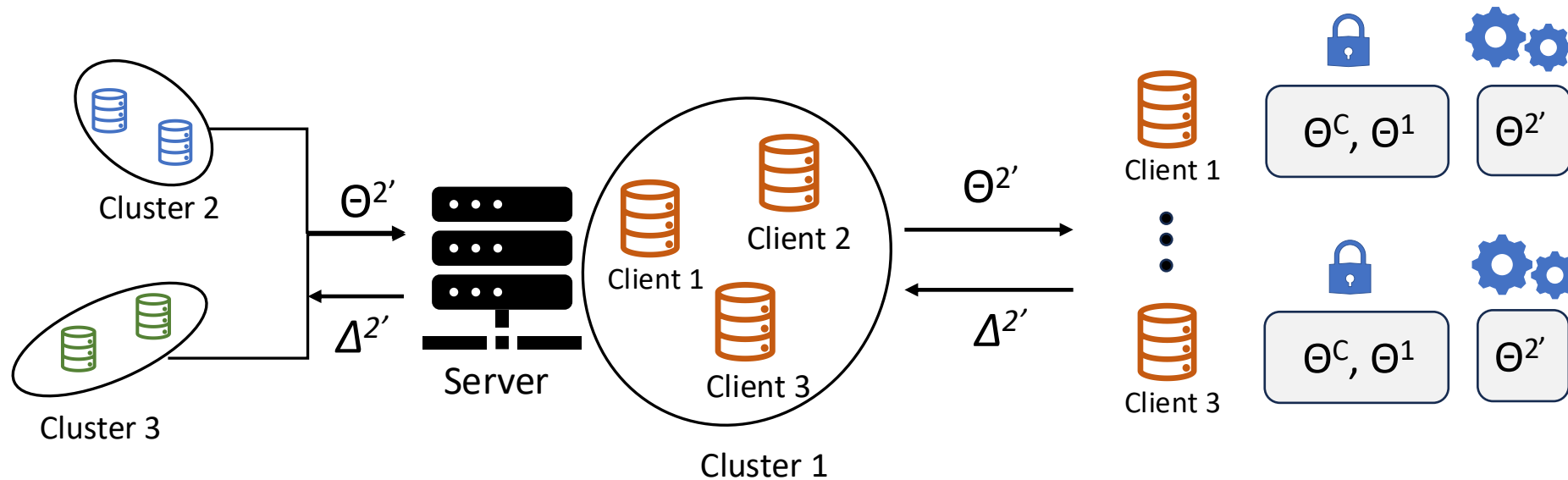
# FedDAG: Feature Sharing Among Clusters

- **Step 2:** FedDAG trains the secondary encoder by sending it to supplier clusters for training
  - Example: Cluster 2 and 3 want to learn from cluster 1.
  - Cluster 2 and 3 combine their secondary encoder  $(\Theta^2)_2$ ,  $(\Theta^2)_3$  and send them to the server
  - Server sends the combined secondary encoder  $\Theta^{2'}$  to the clients of cluster 1



# FedDAG: Feature Sharing Among Clusters

- **Step 3:** Supplier cluster trains the received secondary encoder on its own data and returns the update to the learner clusters
  - Cluster 1 clients initialize their secondary encoder using received  $\Theta^{2'}$  from clusters 2 and 3.
  - Clients freeze the classifier  $\Theta^C$  and the primary encoder  $\Theta^1$  and optimize  $\Theta^{2'}$  on their own data
  - After training, cluster 1 clients send back the encoder update  $\Delta^{2'}$  to the server
  - Server then sends the update  $\Delta^{2'}$  to the learner clusters to update their secondary encoder



Thank you