Appendix of FGTL

Organization of the Appendix. In this Appendix, we first provide the parameter settings for the algorithm implementation. Then, we show two visualization experiments to complement the results we presented in the main paper.

A Implementation Parameter

Each dataset is randomly divided into training (10%), validation (10%), and testing set (80%). For all experiments, we run 3 times and report the average results. For the model architecture, we set the number of layers for GNN-based encoder to 2, the number of layers for MLP-based classifier to 1, and the hidden dimension to 256. For all methods, we set the local training epoch $E = 10$ and the communication round $R = 1500$. For local training, the learning rate is set to $1e-4$ and the weight decay is set to $5e-4$. In generalization phase of FGTL, the communication round is set to 500 and the path length of random walk is set to 10. In transfer phase of FGTL, confidence gate is set to 0.7. Algorithm 1 describes our proposed method, and the code is available on Github[1].

B Additional Experiment

B.1 Performance and Convergence Visualization

We visually verify the advantages brought by each module in FGTL. Figure 1a and 1b show the performance curves of variants of FGTL on the target client, while Figure 1c shows their convergence curves in generalization phase. We can intuitively observe that FGTL¬t not only gains a large performance improvement over FGTL¬g¬t, but also has a significant advantage in convergence speed. Thus, it indicates that global context embedding promotes the convergence of local contrastive learning. Besides, we also notice that both FGTL¬t and FGTL¬g¬t in Figure 1a and 1b show a slight decrease after reaching the peak and then stabilize. We speculate that this is caused by the global model overfitting the source domain labeled data, which is consistent with our motivation to propose consensus knowledge transfer. We observe that FGTL does not show a decline, but stabilizes and slightly improves in the latter half. Therefore, it indicates that consensus knowledge transfer can effectively solve the overfitting problem and enhance the performance of the global model on the target client.

[1] https://github.com/FedGTL/FGTL

Algorithm 1 The training framework of FGTL+

Training in generalization phase

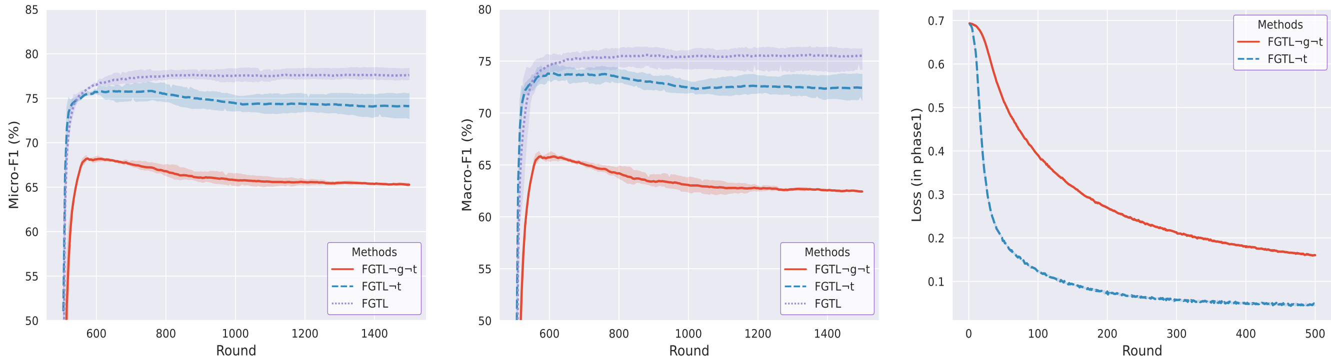
Input: Source domains $\{\mathcal{G}_S^k\}_{k=1}^K$; Target domain $\mathcal{G}_T$
Parameter: Communication round R; Local epoch E; Source clients number K
Output: Global GNN Encoder ω_g^R

Initialize global encoder ω_g^0.
for $r = 1, 2, \dots R$ **do**
  for $k = 1, 2, \dots K$ or t **do**
    Initialize encoder: ω_k or $\omega_t \leftarrow \omega_g^{r-1}$.
    $\hat{\mathcal{G}}_a, \hat{\mathcal{G}}_b \leftarrow$ Perturb($\mathcal{G}$).
    $\tilde{\mathcal{G}} \leftarrow$ Corrupt($\mathcal{G}$).
    $H^{(P)}, \tilde{H}^{(P)}, \hat{H}_a^{(P)}, \hat{H}_b^{(P)} \leftarrow$ Encode($\mathcal{G}; \tilde{\mathcal{G}}; \hat{\mathcal{G}}_a, \hat{\mathcal{G}}_b$).
    $\hat{s}_a^{(P)}, \hat{s}_b^{(P)} \leftarrow$ Readout($\hat{H}_a^{(P)}; \hat{H}_b^{(P)}$).
    Update encoder ω_k or ω_t locally for E times.
  end for
  $\omega_g^r \leftarrow$ Aggregate($\{\omega_k\}_{k=1}^K, \omega_t$).
end for
return ω_g^R

Training in transfer phase

Input: Source domains $\{\mathcal{G}_S^k\}_{k=1}^K$; Target domain $\mathcal{G}_T$
Parameter: Communication round R; Local epoch E; Source clients number K; Group number M
Output: Global Classifier ψ_g^R

$\{H_k^{(P)}\}_{k=1}^K, H_t^{(P)} \leftarrow$ Encode($\{\mathcal{G}_S^k\}_{k=1}^K$;$\mathcal{G}_T$) with ω_g^R.
Initialize global classifier ψ_g^0.
for $r = 1, 2, \dots R$ **do**
  for $k = 1, 2, \dots K$ **do**
    Initialize classifier: $\psi_k \leftarrow \psi_g^{r-1}$.
    Train ψ_k with $\{H_k^{(P)}, Y_k\}$ locally for E times.
  end for
  $\{\tilde{\psi}_m\}_{m=1}^M \leftarrow$ Group($\{\psi_k\}_{k=1}^K$).
  $\{P_m\}_{m=1}^M \leftarrow$ Predict($H_t^{(P)}$) with $\{\tilde{\psi}_m\}_{m=1}^M$.
  $\{\tilde{p}_i, n_i\}_{i=1}^{|V^t|} \leftarrow$ Knowledge Voting with $\{P_m\}_{m=1}^M$.
  $\{H_t^{(P)}, \tilde{P}, n\} \leftarrow$ LPA($\{\tilde{p}_i, n_i\}_{i=1}^{|V^t|}$).
  Distill $\{H_t^{(P)}, \tilde{P}, n\}$ to classifier ψ_{M+1}.
  $\psi_g^r \leftarrow$ Fuse($\{\psi_m\}_{m=1}^{M+1}$).
end for
return ψ_g^R

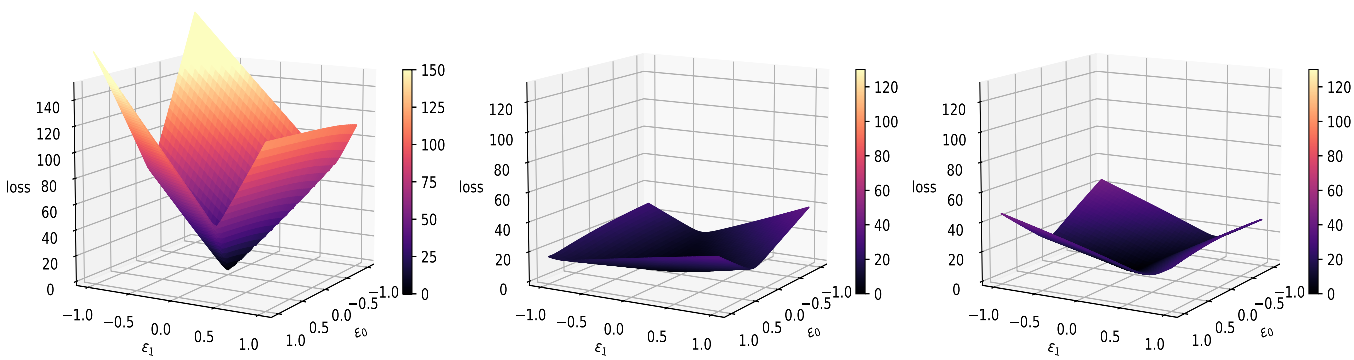


(a) The Micro-F1 curve of global model on the target client during training

(b) The Macro-F1 curve of global model on the target client during training

(c) The curve of average loss for all clients during the generalization phase of training

Figure 1: Comparison of performance and convergence curves among variants of FGTL. Target: DBLPv7, Source: ACMv9, Citationv1.



(a) FedAvg

(b) FGTL¬g¬t

(c) FGTL¬t

Figure 2: Visualization of the parametric loss landscape on target client with Hessian eigenvectors ϵ_0 and ϵ_1 for global model of each method. Target: Citationv1, Source: ACMv9, DBLPv7.

B.2 Generalization Visualization

To intuitively illustrate that local contrastive learning can improve the generalization of GNN-based encoder, we visually compare the loss landscapes of FedAvg, FGTL¬g¬t, and FGTL¬t on the target client in Figure 2. Recent works in network generalization [Jiang *et al.*, 2019; Keskar *et al.*, 2016] show that networks with lower the top Hessian eigenvalue and Hessian trace are generally less sensitive to small perturbations in the network weights, i.e., good generalization. It can smooth the loss space during training, thus facilitating convergence. We can observe in Figure 2 that FGTL¬g¬t and FGTL¬t are able to smooth the loss landscape, compared to FedAvg. Thus, it suggests that local contrastive learning promotes the generalization of the global model, thus alleviating the client heterogeneity problem.

References

[Jiang *et al.*, 2019] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

[Keskar *et al.*, 2016] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.