Algorithm 1 CNSL Training Framework

Require: G_s, G_t, f_s, L, x_s, y_t
Ensure: Trained $q_\phi(\cdot)$, $p_\theta(\cdot)$, and $p_\psi(\cdot)$

1: **for** *epoch* in 1 to *num_epochs* **do**
2: **for** each batch in *train_set* **do**
3: $z_s = q_{\phi_1}(x_s, G_s)$ ▷ Dynamic Encoder
4: $z_{fs} = q_{\phi_2}(x_s, x_{fs}, G_s)$ ▷ Static Encoder
5: $\hat{x}_s = p_\theta(z_s, z_{fs})$ ▷ Decoder
6: $\hat{y}_s = p_{\psi_1}(\hat{x}_s, G_s)$ ▷ Source Network Diffusion
7: $\hat{x}_t \leftarrow \hat{y}_s$ ▷ $L = \{(v_s, v_t) | v_s \in V_s, v_t \in V_t\}$
8: $\hat{y}_t = p_{\psi_2}(\hat{x}_t, G_t)$ ▷ Target Network Diffusion
9: Calculate $\mathcal{L}_{\text{train}}$ ▷ Equation (3)
10: Backpropagate loss
11: Update model parameters
12: **end for**
13: **end for**

For training, we want to use observed x_s and y_t to learn the approximate posterior $q_\phi(Z|x_s, \mathcal{G})$, the decoding function $p_\theta(x_s|Z)$, and the cross-network diffusion prediction function $p_\psi(y_t|x_s, \mathcal{G})$. Specifically, we separately obtain two latent variables z_s and z_{fs} in Line 2-3. Both z_s and z_{fs} are fed to reconstruct $\hat{x}_s$ in Line 5. After the seed set reconstruction, we conduct cross-network diffusion prediction as shown in Line 6-8. The backpropagation is calculated based on Eq. (3) that consists of seed nodes reconstruction error, diffusion estimation error, as well as constraints of KL divergence and influence monotonicity.