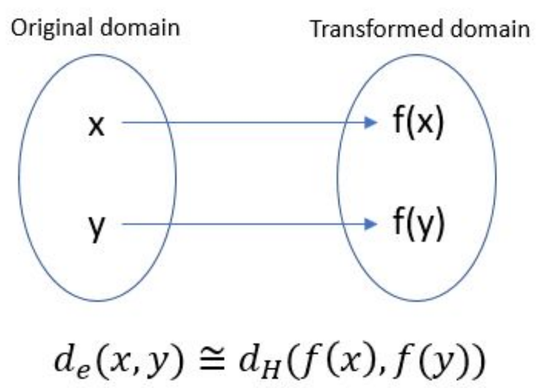
Edit Embedding via Reinforcement Learning

Shunfu Mao (EE) and Joshua Fan (CS)

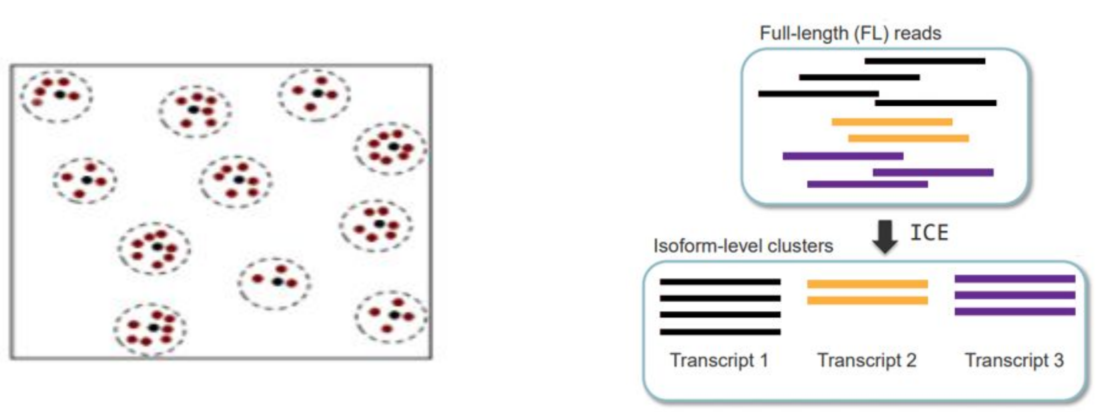
Problem Statement

- **Edit Distance Approximation**
  - Design a transformation f such that $d_e(x,y) \cong d_H(f(x), f(y))$



- x, y
  input strings
- f(x), f(y)
  transformed strings
- $d_e(x,y)$
  edit distance between x and y
- $d_H(x,y)$
  hamming distance between x and y

  - A theoretical computer science research topic
    - Main motivation is to reduce computational complexity
    - Requires ingenious mathematical design
    - CGK-embedding f_{CGK} [3]
  - Can we design f using neural network architectures?
    - Objective is to minimize transform deviation (max R(s,a) in RL framework)
    - Consider reinforcement learning for the non-differentiable objective function
- **Downstream Application of Sequence Clustering**
  - DNA Storage recovery [1] and Long RNA-Seq assembly [2]



Models (Cont.)

- **RL Algorithms (PGPE [8])**
  - Basic idea
    Sampling directly in parameter space for lower variance gradient estimates
  - Objective

$$J(\rho) = \int_{\Theta}\int_{H} p(h,\theta|\rho) r(h) dh d\theta$$

  - Metaparam (ρ) update and param (θ) sampling

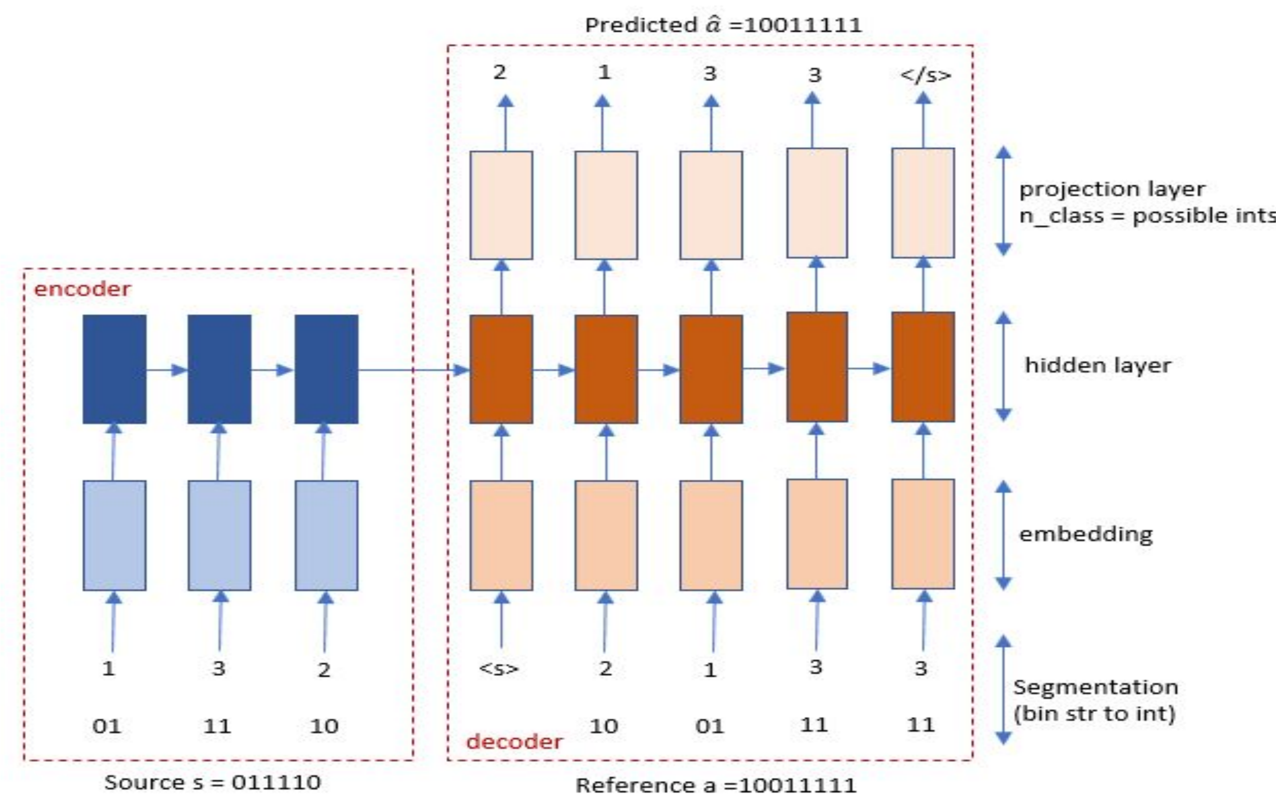
$$\nabla_{\mu_i} \log p(\theta|\rho) = \frac{\theta_i - \mu_i}{\sigma_i^2} \qquad \nabla_{\sigma_i} \log p(\theta|\rho) = \frac{(\theta_i-\mu_i)^2 - \sigma_i^2}{\sigma_i^3}$$

$$\nabla_\rho J(\rho) \approx \frac{1}{N}\Sigma_{n=1}^{N} \nabla_\rho \log p(\theta|\rho) r(h^n) \qquad \rho \leftarrow \rho + \alpha \nabla_\rho J(\rho)$$

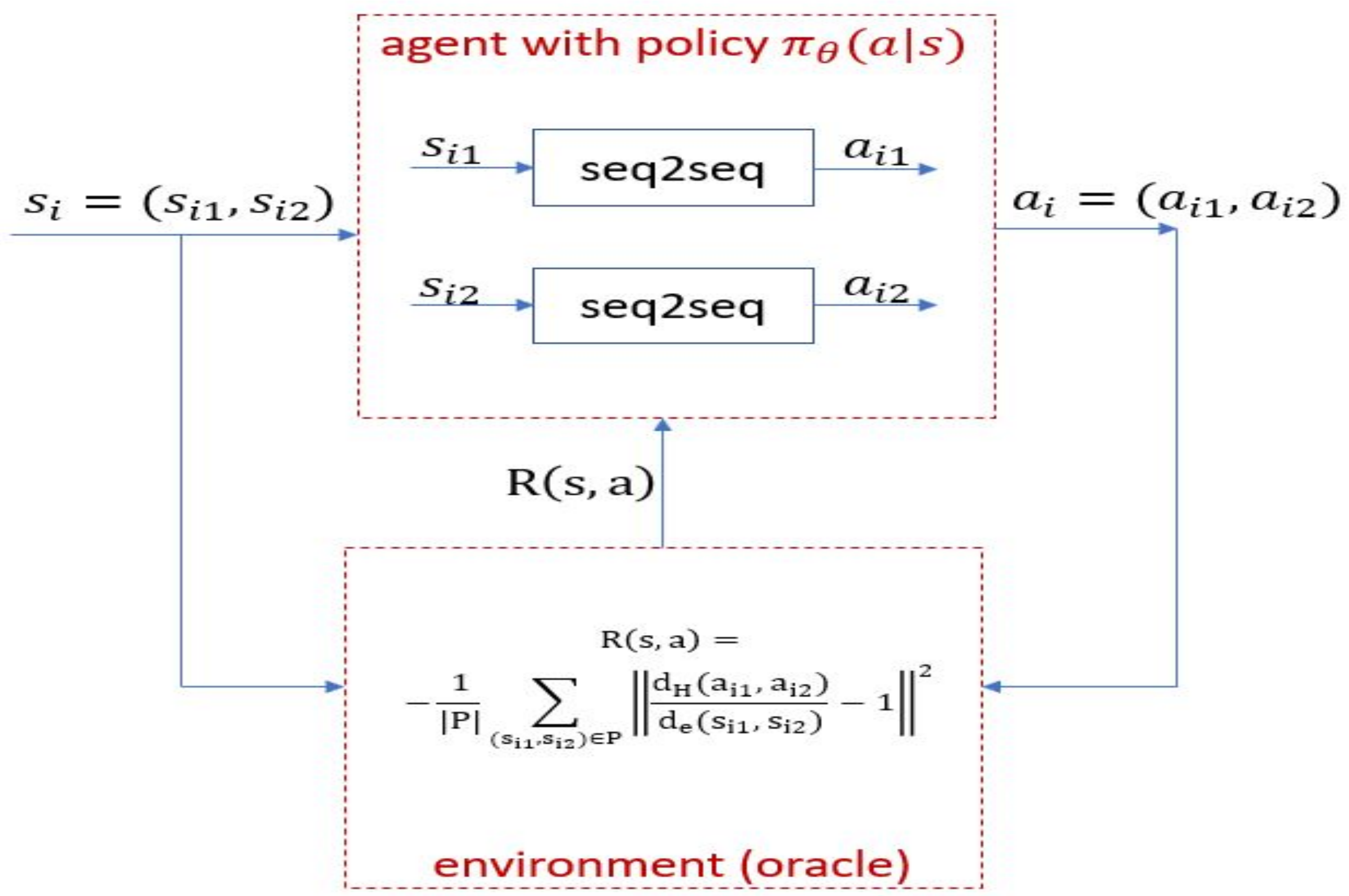
$$\rho_i = (u_i, \sigma_i), \theta_i \sim N(u_i, \sigma_i^2)$$

Models

- **Sequence-to-Sequence (Seq2Seq) Architecture [4]**
  - Learn an existing edit embedding (e.g CGK), as a pre-train step
  - BiLSTM + Attention used



- **Reinforcement Learning (RL) Framework**
  - Agent is a Siamese Network [5], with input s_i as state and output a_i as action at time i
  - Agent's policy is updated per $|P|$ time steps



- **RL Algorithms (REINFORCE [6])**
  - Basic idea
    Increase (decrease) weights for backprop with high (low) rewards
  - Objective

$$\eta(\theta) = \mathbb{E}_{a\sim\pi(a|s,\theta)}[R(s,a)]$$

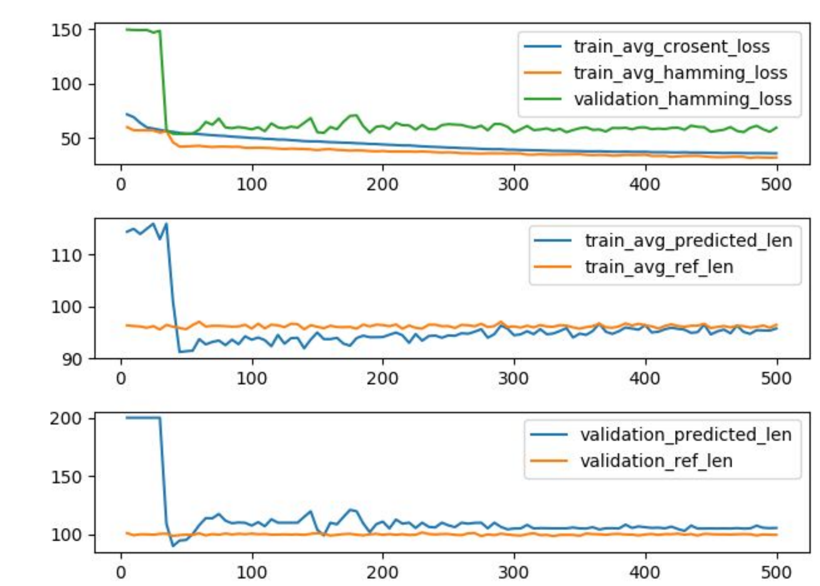
  - Gradients update

$$\nabla_\theta \eta(\theta) = \mathbb{E}_{a\sim\pi(a|s,\theta)}[R(s,a)\nabla_\theta \log \pi(a|s,\theta)]$$

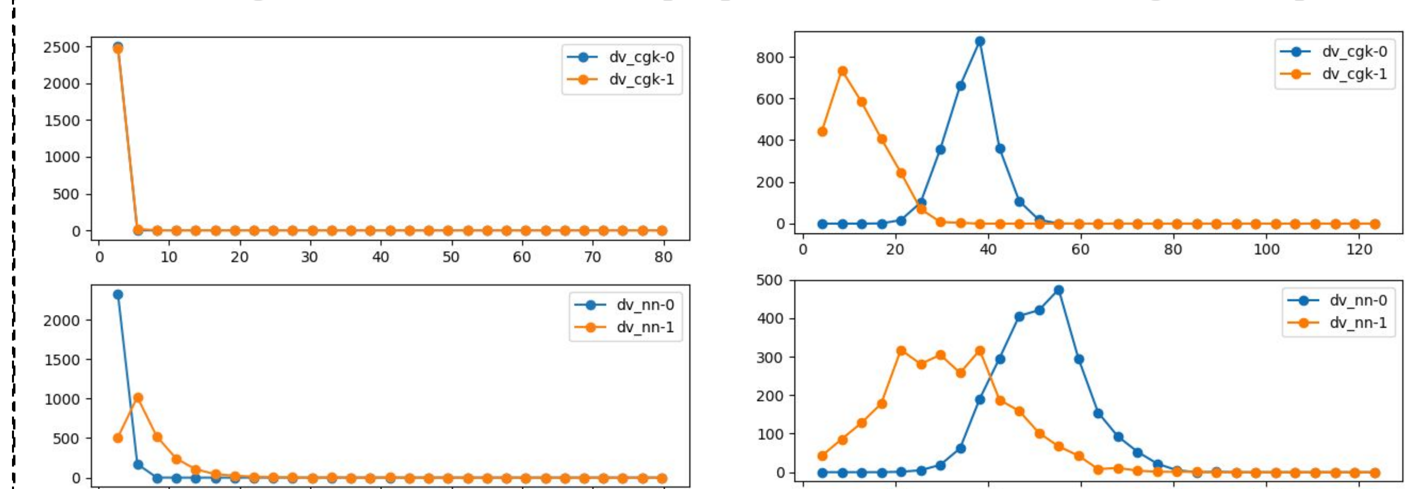
$$\theta \leftarrow \theta + \alpha \nabla_\theta \eta(\theta)$$

Evaluation

- **Experiments**
  - Dataset: synthetic binary data
  - Seq2Seq: $\{(s, a = f_{CGK}(s))\}$
  - RL: $\{(s_{i1}, s_{i2}, d_e(s_{i1}, s_{i2}))\}$
- **Seq2Seq**
  - 10K samples for training, 2K samples for validation
  - BiLSTM with Attention reduces cross-entropy loss further
  - Seq2Seq model does not seem to learn the CGK embedding well



- **RL with Seq2Seq pre-train, without policy update**
  - (s_{i1}, s_{i2}) can be independent/blue or similar/orange (s_{i2} is obtained via s_{i1} through an indel channel with substitution rate). Distance (d_H) is better than Deviation ($\frac{d_H}{d_e} - 1$) to distinguish similar/ dis-similar input pairs. CGK is better to distinguish the pairs.



- **RL with Seq2Seq pre-train, with policy update**
  - PGPE: larger σ_i or learning rate makes training not stable. Rewards seems not better.

Reference

[1] Clustering Billions of Reads for DNA Data Storage
[2] Widespread Polycistronic Transcripts in Fungi Revealed by Single-Molecule mRNA Sequencing
[3] Streaming algorithms for embedding and computing edit distance in the low distance regime.
[4] Learning phrase representations using rnn encoderdecoder for statistical machine translation
[5] Siamese recurrent architectures for learning sentence similarity
[6] Simple statistical gradient-following algorithms for connectionist reinforcement learning
[7] Selfcritical sequence training for image captioning
[8] Parameter-exploring Policy Gradients