

CHATBOT DAISY

Dialog Analogous Intellectual System

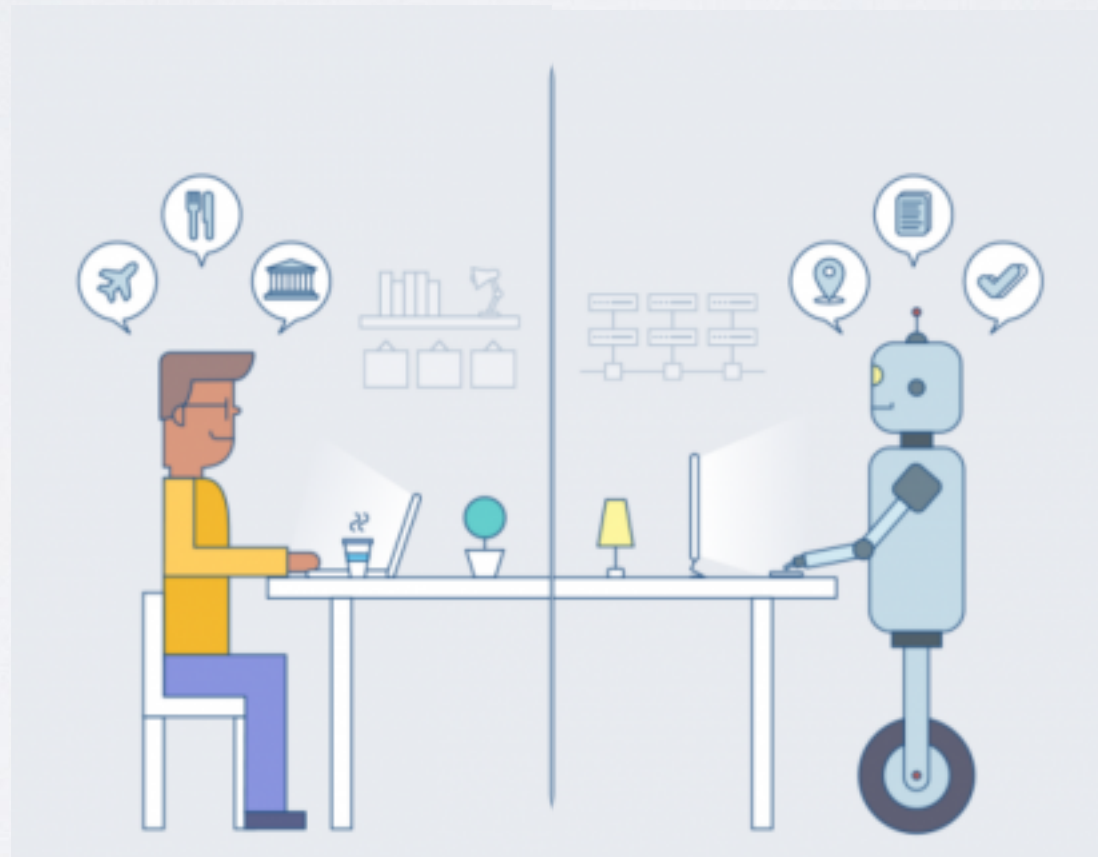
電機四 劉廷緯 B03901170

電機四 溫明浩 B03901179

2018.03.07

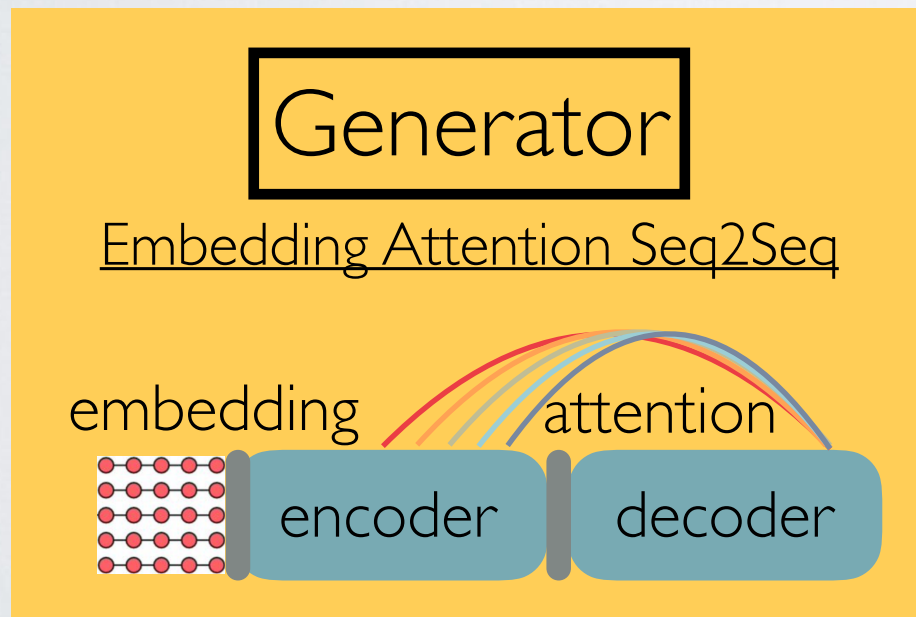
TASK

Improve ChatBot dialog generation quality:
to produce more intellectual and meaningful conversations



FRAMEWORK

conditional sequence generative adversarial
network trained with policy gradient

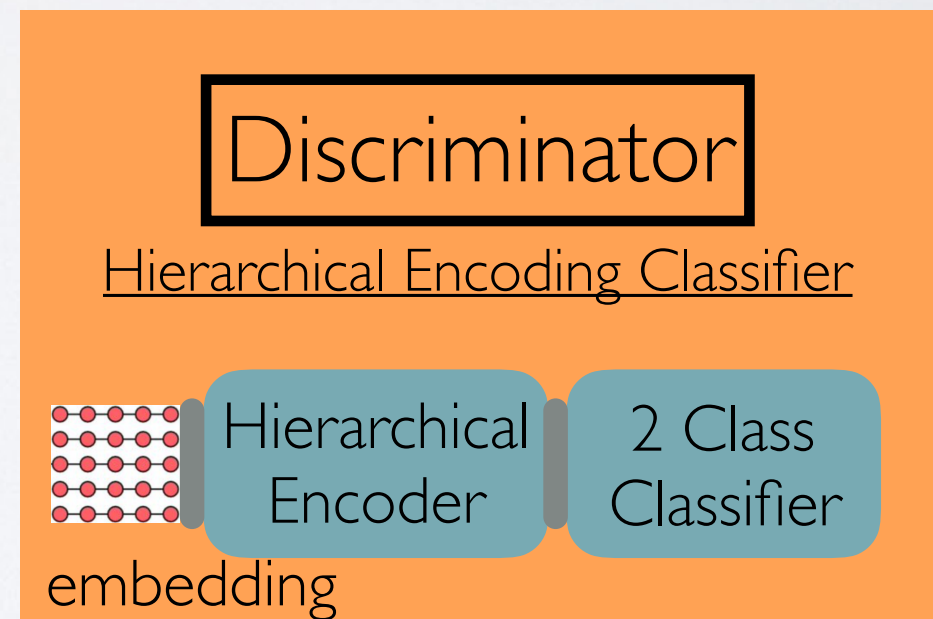


defines the policy that generates a
response y given dialogue history x

FRAMEWORK

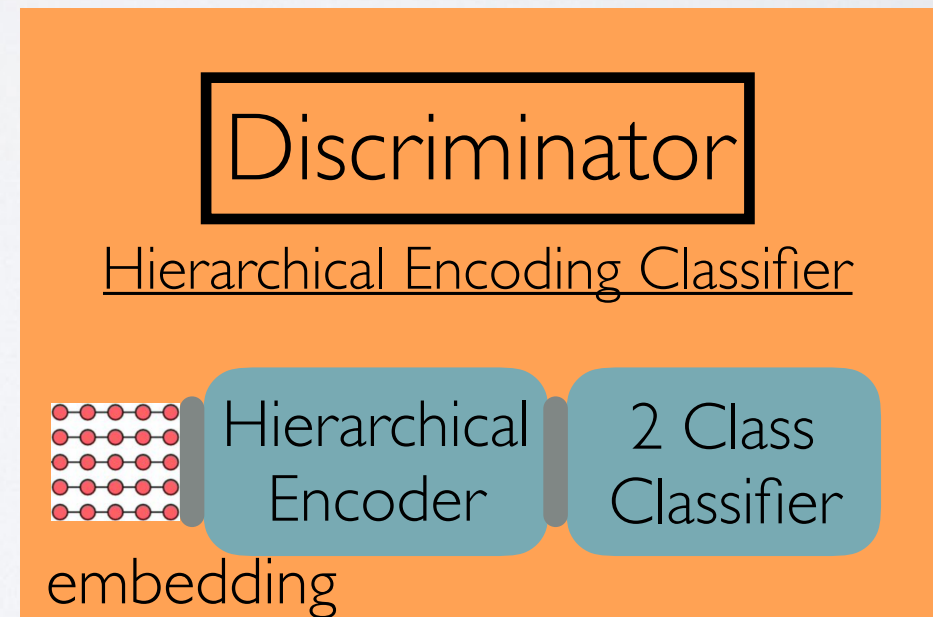
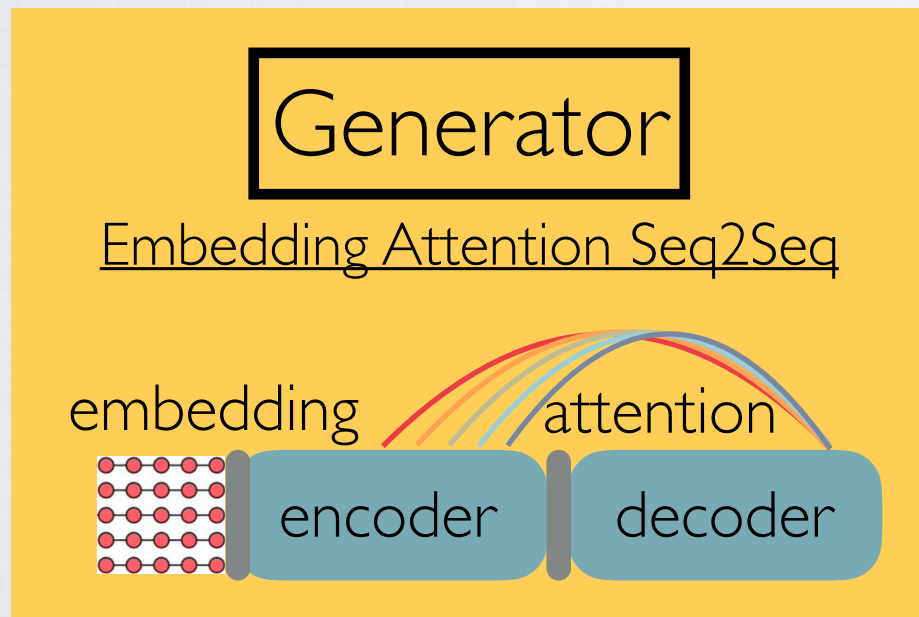
conditional sequence generative adversarial
network trained with policy gradient

Classifier that takes as input a sequence of dialogue utterances $\{x,y\}$ and outputs a label indicating whether the input is generated by humans or machines



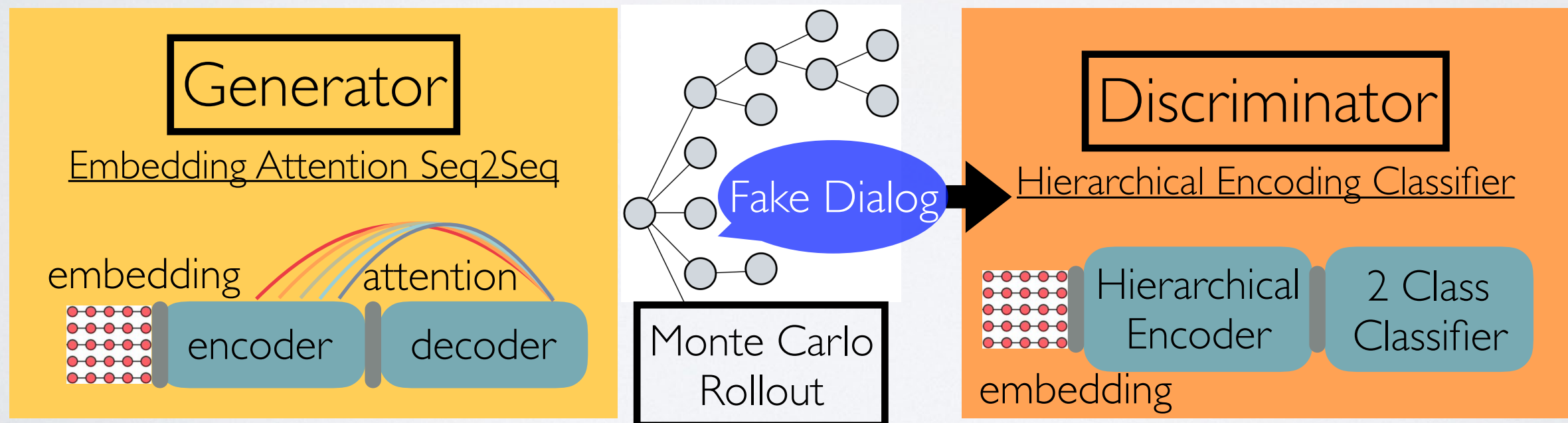
FRAMEWORK

conditional sequence generative adversarial
network trained with policy gradient



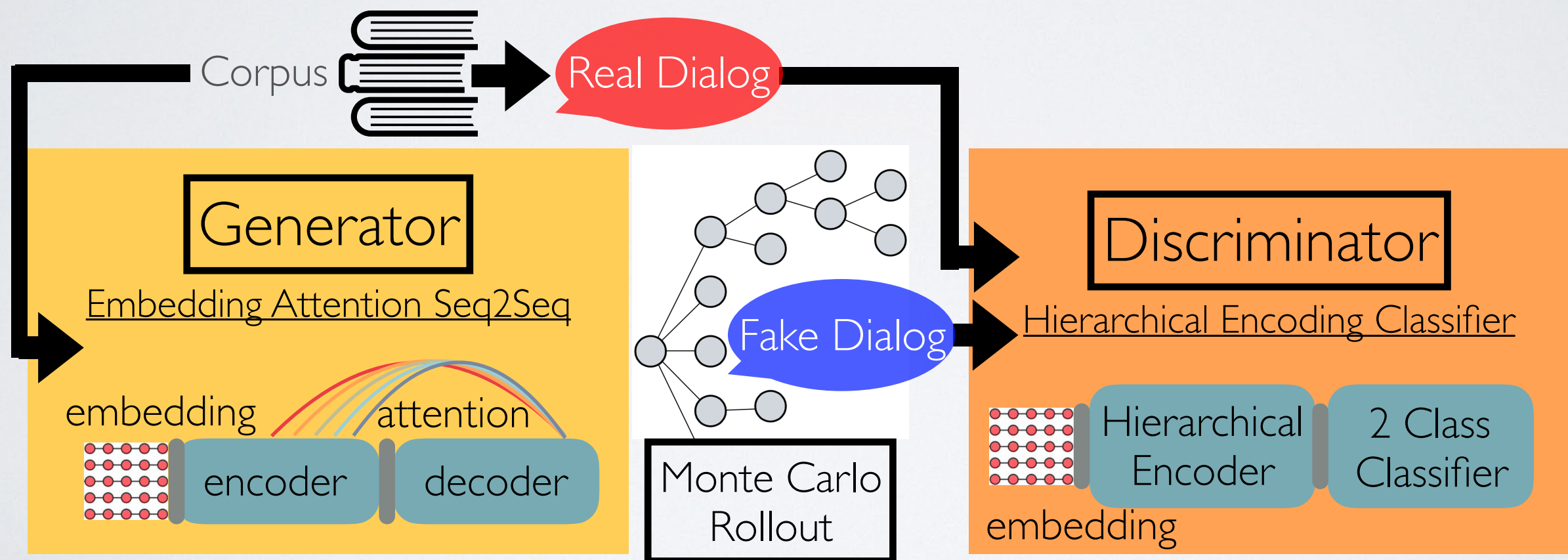
FRAMEWORK

conditional sequence generative adversarial
network trained with policy gradient



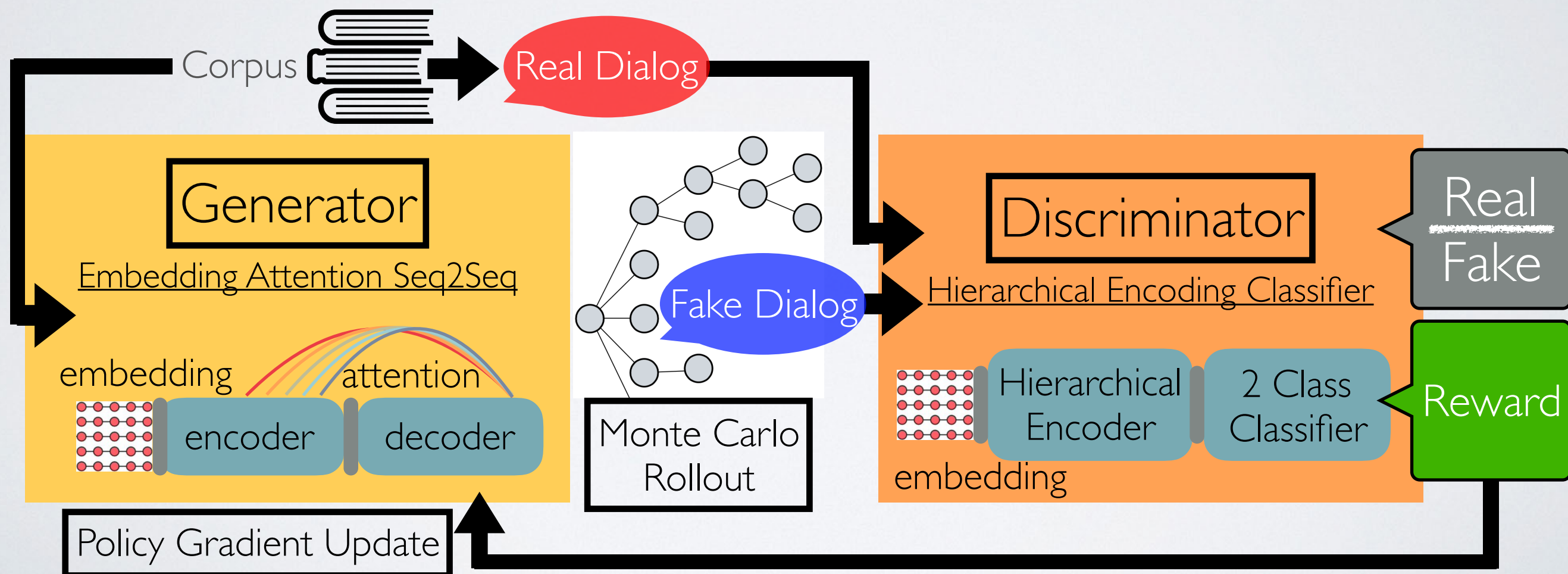
FRAMEWORK

conditional sequence generative adversarial
network trained with policy gradient



FRAMEWORK

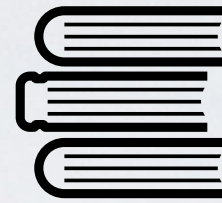
conditional sequence generative adversarial
network trained with policy gradient



DETAILS ABOUT OUR FRAMEWORK



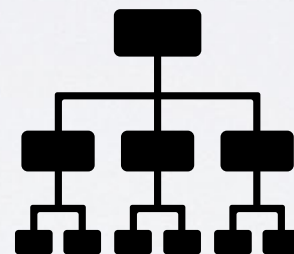
Seq2Seq with Attention



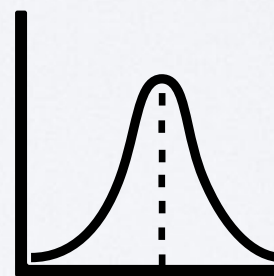
Training Corpus



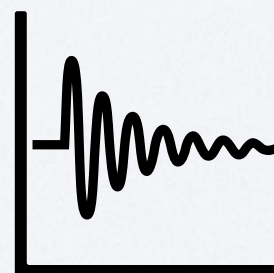
Policy Gradient



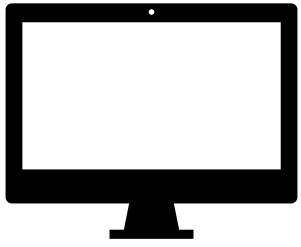
Monte Carlo Rollout



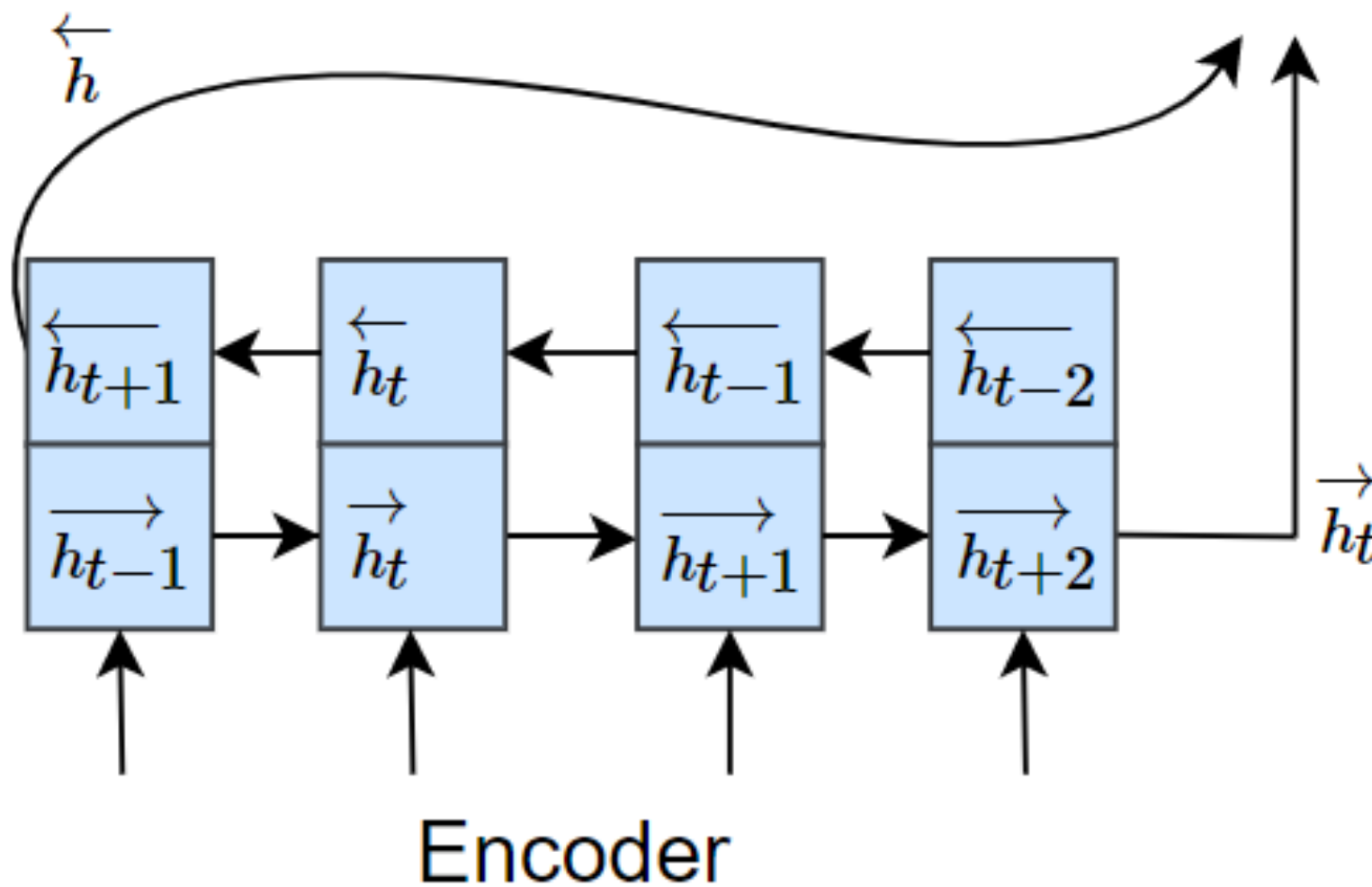
Sampled Softmax



Training Procedure



SEQ2SEQ + ATTENTION

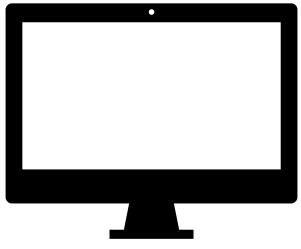


$$\overrightarrow{h_{e,t}} = GRU_{\rightarrow}(\overrightarrow{h_{e,t-1}}, Ex_t)$$

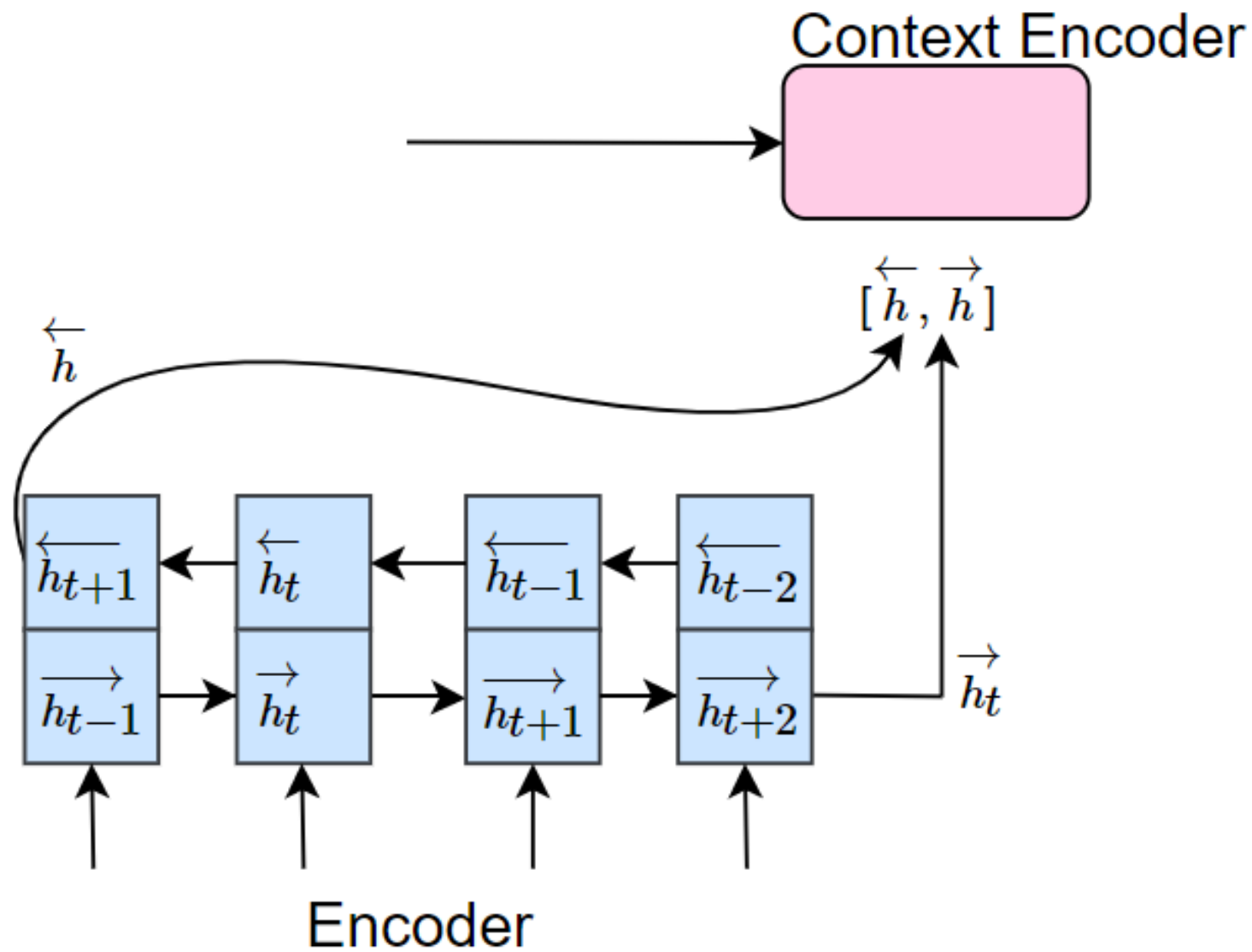
$$\overleftarrow{h_{e,t}} = GRU_{\leftarrow}(\overleftarrow{h_{e,t+1}}, Ex_t)$$

$$h_{e,t}^o = [\overleftarrow{h_{e,t}} , \overrightarrow{h_{e,t}}]$$

$$h_{c,t} = GRU(h_{c,t-1}, h_{e,t}^o)$$



SEQ2SEQ + ATTENTION

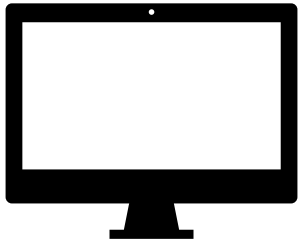


$$\overrightarrow{h_{e,t}} = GRU_{\rightarrow}(\overrightarrow{h_{e,t-1}}, Ex_t)$$

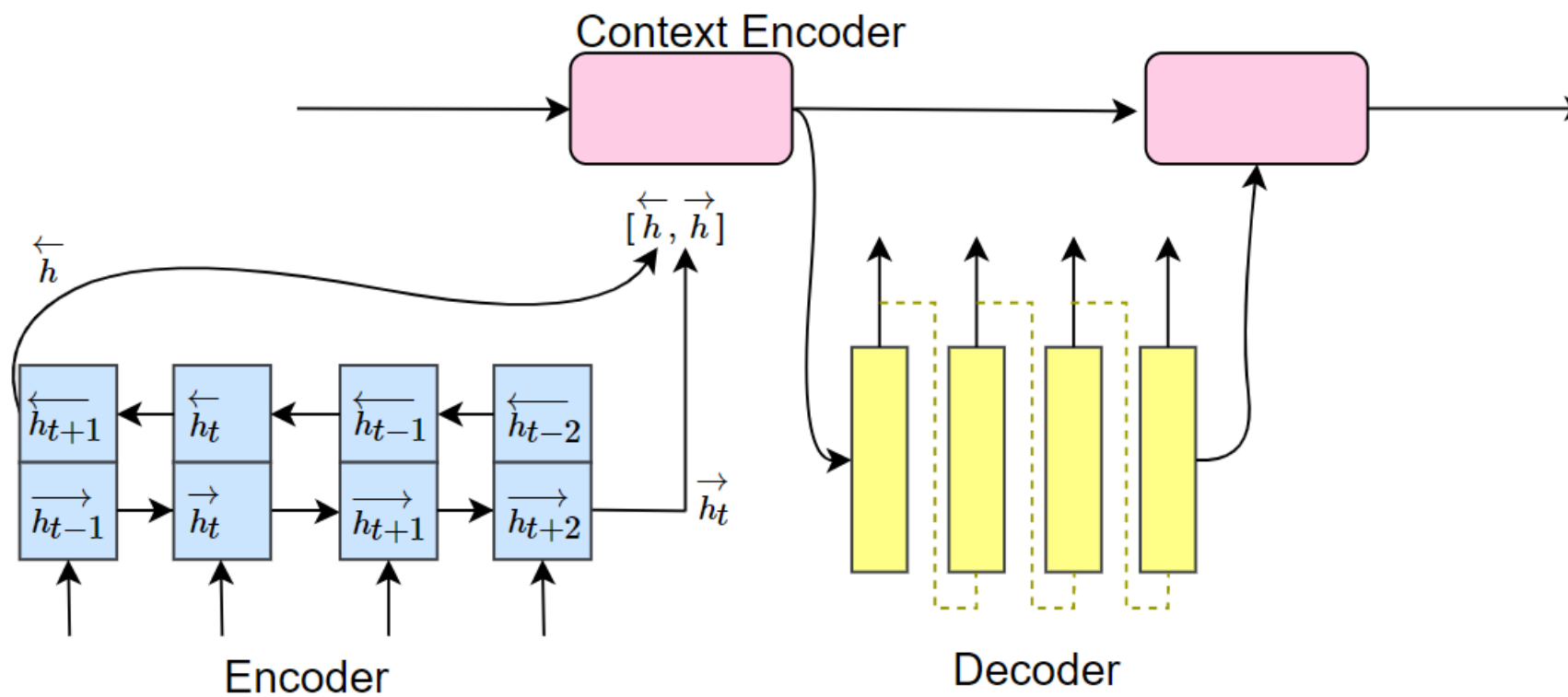
$$\overleftarrow{h_{e,t}} = GRU_{\leftarrow}(\overleftarrow{h_{e,t+1}}, Ex_t)$$

$$h_{e,t}^o = [\overleftarrow{h_{e,t}}, \overrightarrow{h_{e,t}}]$$

$$h_{c,t} = GRU(h_{c,t-1}, h_{e,t}^o)$$



SEQ2SEQ + ATTENTION



$$\overrightarrow{h_{e,t}} = GRU_{\rightarrow}(\overrightarrow{h_{e,t-1}}, Ex_t)$$

$$\overleftarrow{h_{e,t}} = GRU_{\leftarrow}(\overleftarrow{h_{e,t+1}}, Ex_t)$$

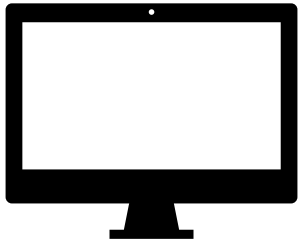
$$h_{e,t}^o = [\overleftarrow{h_{e,t}}, \overrightarrow{h_{e,t}}]$$

$$h_{c,t} = GRU(h_{c,t-1}, h_{e,t}^o)$$

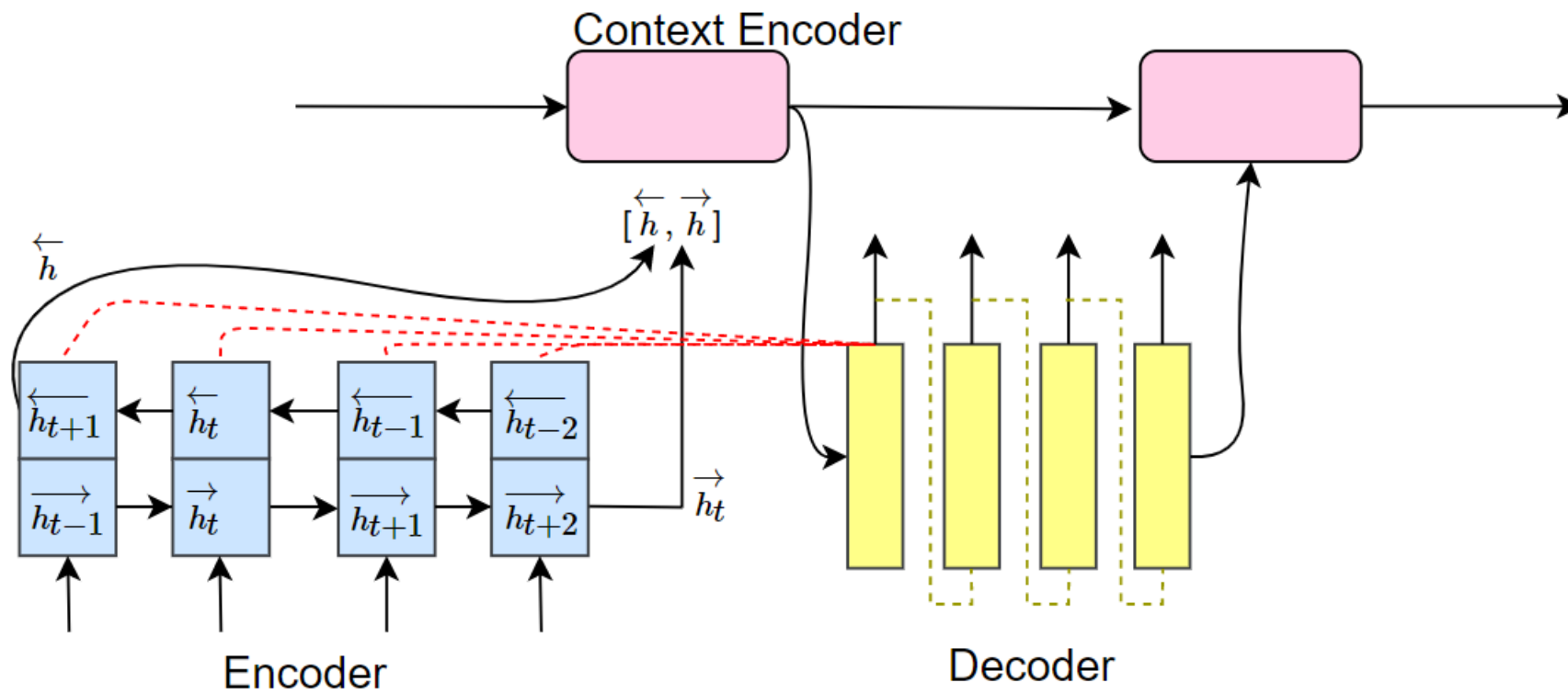
$$h_d = h_{c,t}$$

$$h_{d,t} = GRU(\overrightarrow{h_{d,t-1}}, Ey_t)$$

$$h_{c,t} = GRU(h_{c,t-1}, h_{d,t}^o)$$



SEQ2SEQ + ATTENTION



$$\overrightarrow{h_{e,t}} = GRU_{\rightarrow}(\overrightarrow{h_{e,t-1}}, Ex_t)$$

$$\overleftarrow{h_{e,t}} = GRU_{\leftarrow}(\overleftarrow{h_{e,t+1}}, Ex_t)$$

$$h_{e,t}^o = [\overleftarrow{h_{e,t}}, \overrightarrow{h_{e,t}}]$$

$$h_{c,t} = GRU(h_{c,t-1}, h_{e,t}^o)$$

$$h_d = h_{c,t}$$

$$h_{d,t} = GRU(\overrightarrow{h_{d,t-1}}, Ey_t)$$

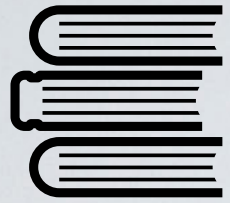
$$h_{c,t} = GRU(h_{c,t-1}, h_{d,t}^o)$$

$$w_{0...m} = softmax(linear([O_{e,0...m}]) \cdot h_t)$$

$$C = \sum_{i=0}^m O_{e,i} w_i$$

$$h^s = \tanh(linear([C, O_d]))$$

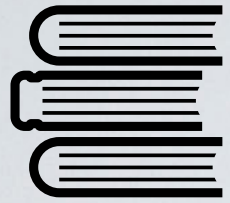
$$p_0, p_1, \dots, p_v = softmax(linear(h_s))$$



TRAINING CORPUS

Cornell Movie-Dialogs Corpus dataset

- 220,579 conversational
- 10,292 pairs of movie characters
- 9,035 characters from 617 movies
- Total of 304,713 utterances



TRAINING CORPUS

Cornell Movie-Dialogs Corpus dataset

- 220,579 conversational
- 10,292 pairs of movie characters
- 9,035 characters from 617 movies
- Total of 304,713 utterances

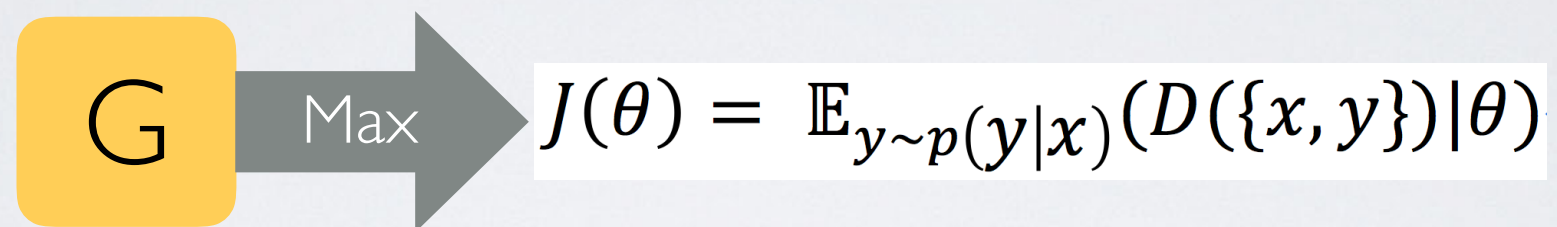


Select dialogs that has len \leq 19:
138127 dialog training pairs



POLICY GRADIENT

Reward is used to encourage the generator to generate higher quality dialogs



A diagram illustrating the policy gradient objective. On the left, a yellow square contains the letter 'G'. A grey arrow points from this square to the right, with the word 'Max' written inside it. To the right of the arrow, the equation $J(\theta) = \mathbb{E}_{y \sim p(y|x)}(D(\{x, y\})|\theta)$ is displayed within a white rectangular box.


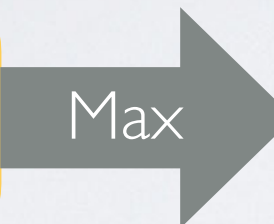
$$J(\theta) = \mathbb{E}_{y \sim p(y|x)}(D(\{x, y\})|\theta)$$

maximize the expected reward of generated sequence



POLICY GRADIENT

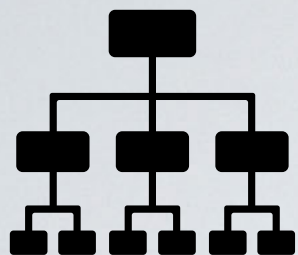
Reward is used to encourage the generator to generate higher quality dialogs

  $J(\theta) = \mathbb{E}_{y \sim p(y|x)} (D(\{x, y\}) | \theta)$

maximize the expected reward of generated sequence

Approximate

$$\begin{aligned} \nabla J(\theta) &\approx [D(\{x, y\}) - b] \nabla \log \pi(y|x) \\ &= [D(\{x, y\}) - b] \nabla \sum_t \log p(y_t | x, y_{1:t-1}) \end{aligned}$$



MC ROLLOUT



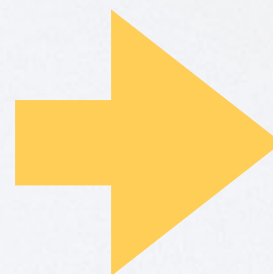
Where are you from ?



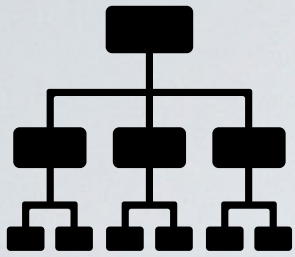
I don't know.



I am from Taiwan.

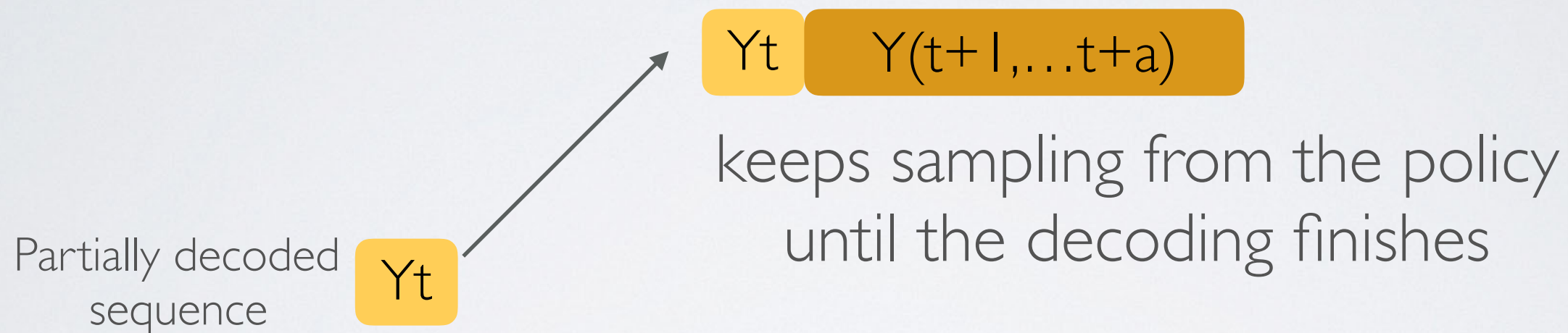


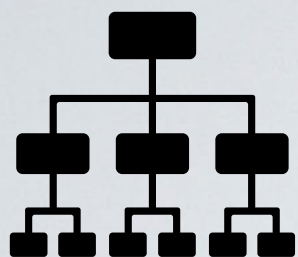
the first token I
should receive a
neutral reward



MC ROLLOUT

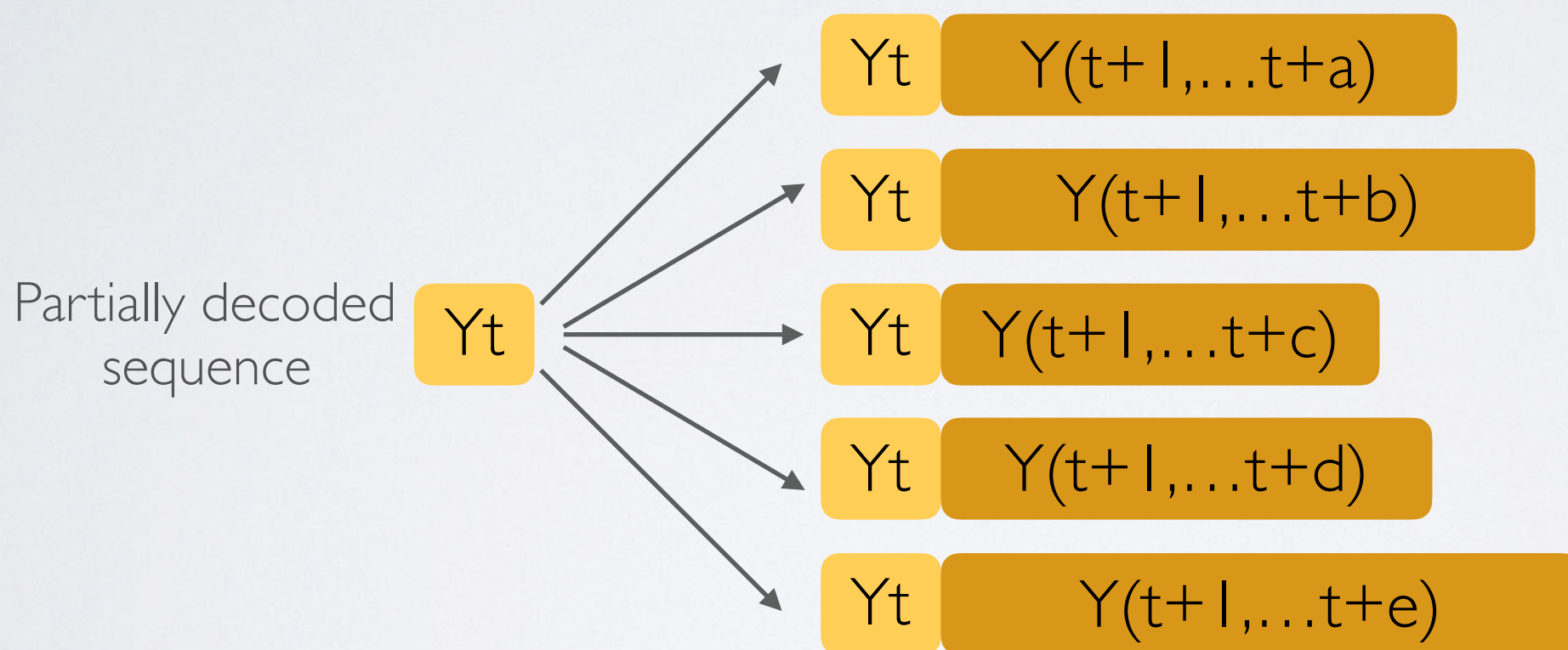
compute intermediate reward by using the Monte Carlo (MC) search



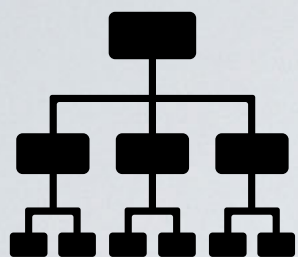


MC ROLLOUT

compute intermediate reward by using the Monte Carlo (MC) search

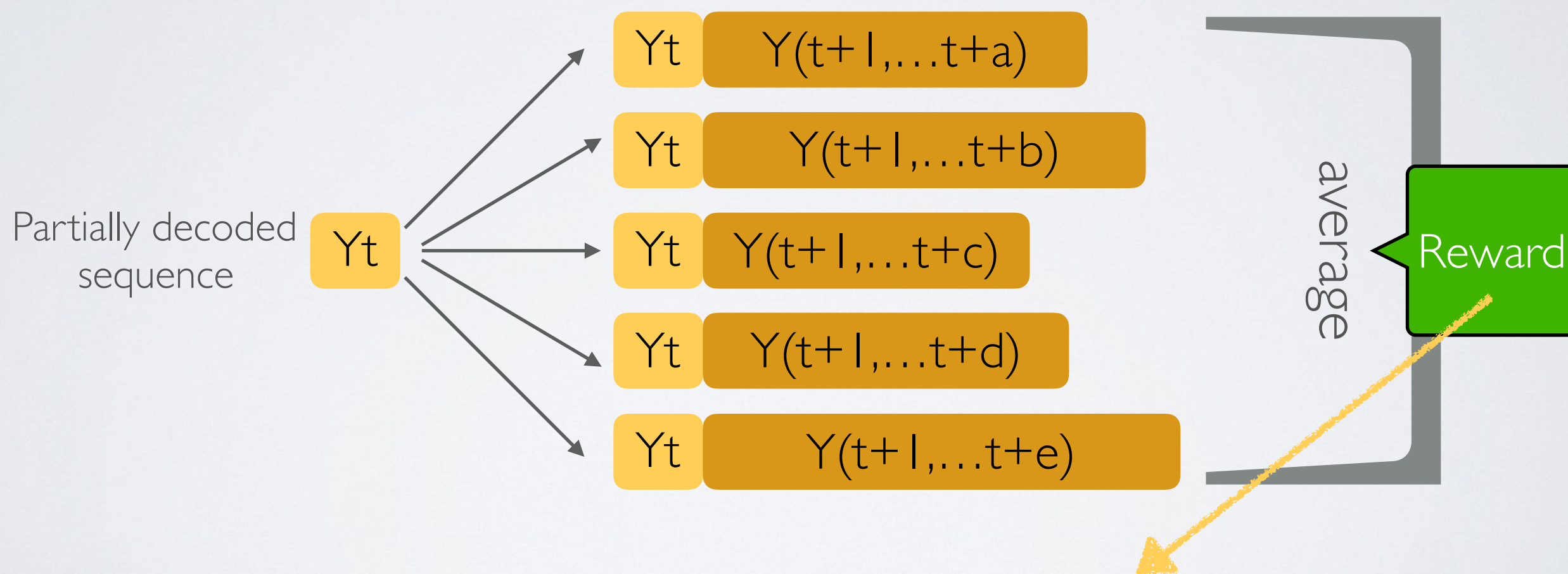


This process is repeated N times ($N=5$), resulting in N fully generated sequences that starts with Y_t .

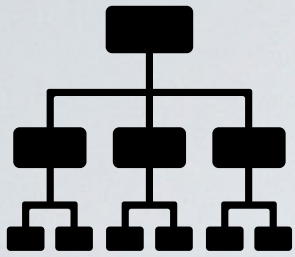


MC ROLLOUT

compute intermediate reward by using the Monte Carlo (MC) search



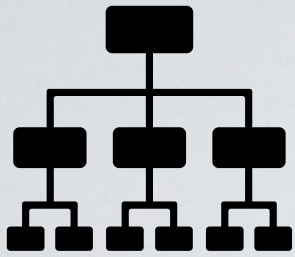
represents the intermediate reward at time step t



MC ROLLOUT

compute intermediate reward by using the Monte Carlo (MC) search

$$\begin{aligned}\nabla J(\theta) &\approx [D(\{x, y\}) - b] \nabla \log \pi(y|x) \\ &= [\underline{D(\{x, y\})} - b] \nabla \sum_t \log p(y_t|x, y_{1:t-1})\end{aligned}$$



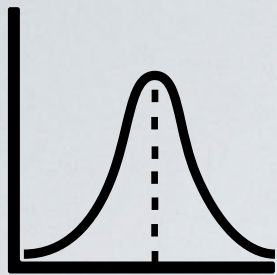
MC ROLLOUT

compute intermediate reward by using the Monte Carlo (MC) search

$$\begin{aligned}\nabla J(\theta) &\approx [D(\{x, y\}) - b] \nabla \log \pi(y|x) \\ &= [\underline{D(\{x, y\})} - b] \nabla \sum_t \log p(y_t|x, y_{1:t-1})\end{aligned}$$

$$\nabla J(\theta) \approx \sum_t [\underline{D(\{x, Y_t\})} - b] \nabla \log p(y_t|x, Y_{1:t-1})$$

$$D(\{x, Y_t\}) = \frac{1}{N} \sum_{i=1}^N D(\{x, Y_t^i\})$$



SAMPLED SOFTMAX

Paper: 《 On Using Very Large Target Vocabulary for Neural Machine Translation 》

Regular Softmax

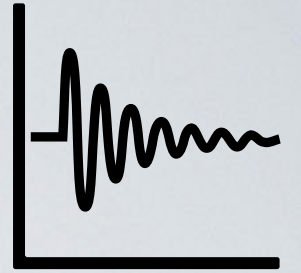
large vocabulary has a high cost at calculating regular softmax, because probability for every word in the vocabulary set has to be calculated.



Sampled Softmax

take in account subset V of your vocabulary to calculate loss.

TRAINING



Pre-train **G** on real data

For training_epochs **do:**

For D_steps **do:**

Real data ← **C**

Fake data ← **G**

Update **D** with real and fake data

End

For G_steps **do:**

Real data ← **C**

Fake data ← Monte Carlo Rollout **G**

Compute reward r for fake data using

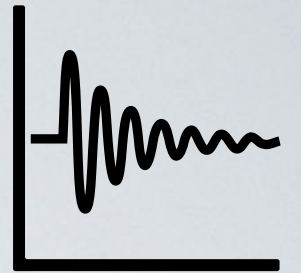
Policy Gradient: Update **G** on fake data with r

Teacher Forcing: Update **G** on real data

End

End

TRAINING



Pre-train **G** on real data

For training_epochs do:

For D_steps do:

Real data ← **C**

Fake data ← **G**

Update **D** with real and fake data

End

For G_steps do:

Real data ← **C**

Fake data ← Monte Carlo Rollout **G**

Compute reward r for fake data using

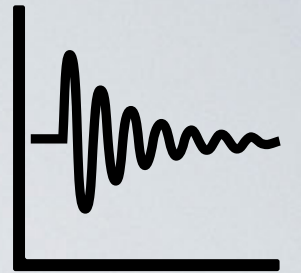
Policy Gradient: Update **G** on fake data with r

Teacher Forcing: Update **G** on real data

End

End

TRAINING



Pre-train **G** on real data

For training_epochs **do:**

For D_steps **do:**

Real data ← **C**

Fake data ← **G**

Update **D** with real and fake data

End

For G_steps **do:**

Real data ← **C**

Fake data ← Monte Carlo Rollout **G**

Compute reward r for fake data using

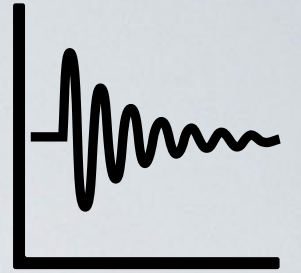
Policy Gradient: Update **G** on fake data with r

Teacher Forcing: Update **G** on real data

End

End

TRAINING



Pre-train **G** on real data

For training_epochs **do**:

For D_steps **do**:

 Real data ← **C**

 Fake data ← **G**

 Update **D** with real and fake data

End

For G_steps **do**:

 Real data ← **C**

 Fake data ← Monte Carlo Rollout **G**

 Compute reward r for fake data using

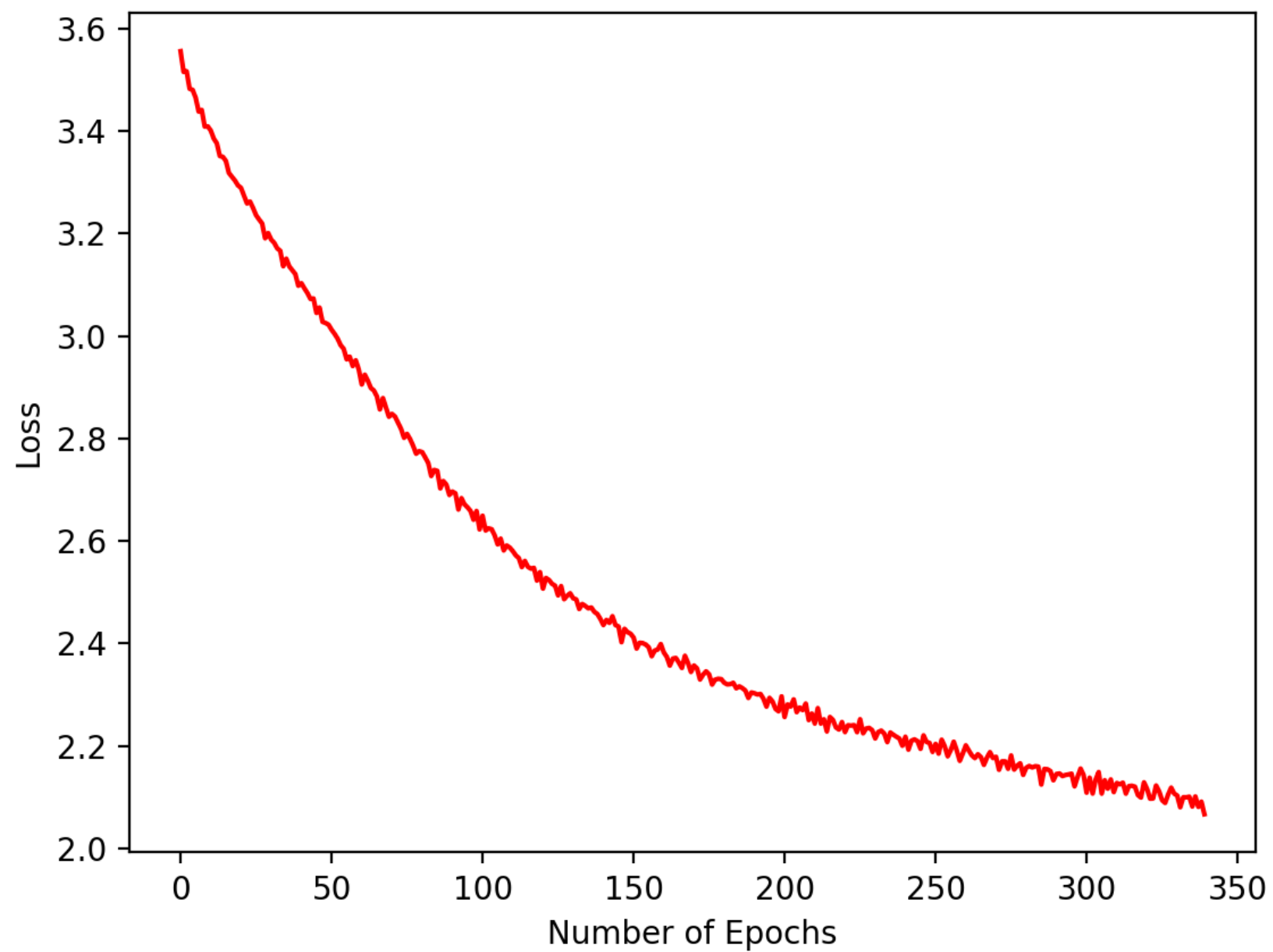
Policy Gradient: Update **G** on fake data with r

Teacher Forcing: Update **G** on real data


End

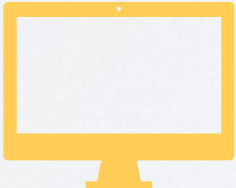
End

PRE-TRAIN CURVE



BLEU SCORE

Seq2Seq  0.395

GAN  0.352

CONVERSATION DEMO

User



you don't care if i die

Seq2Seq



oh ...!!!!!!!

GAN



oh ... yeah ... don't !.

User



you always been this selfish ?

Seq2Seq



oh , shit ! oh ... oh ... oh

GAN



oh my god ! oh yeah , shit !

CONVERSATION DEMO

User



did you change your hair ?

Seq2Seq



some off ...

GAN



oh , thanks !

User



where've you been ?

Seq2Seq



oh me too you are great .

GAN



oh ... you a maybe a little old.

CONVERSATION DEMO

User



you hate me don't you ?

Seq2Seq



what are you is is there a all night.

GAN



say you're a

User



are you following me ?

Seq2Seq

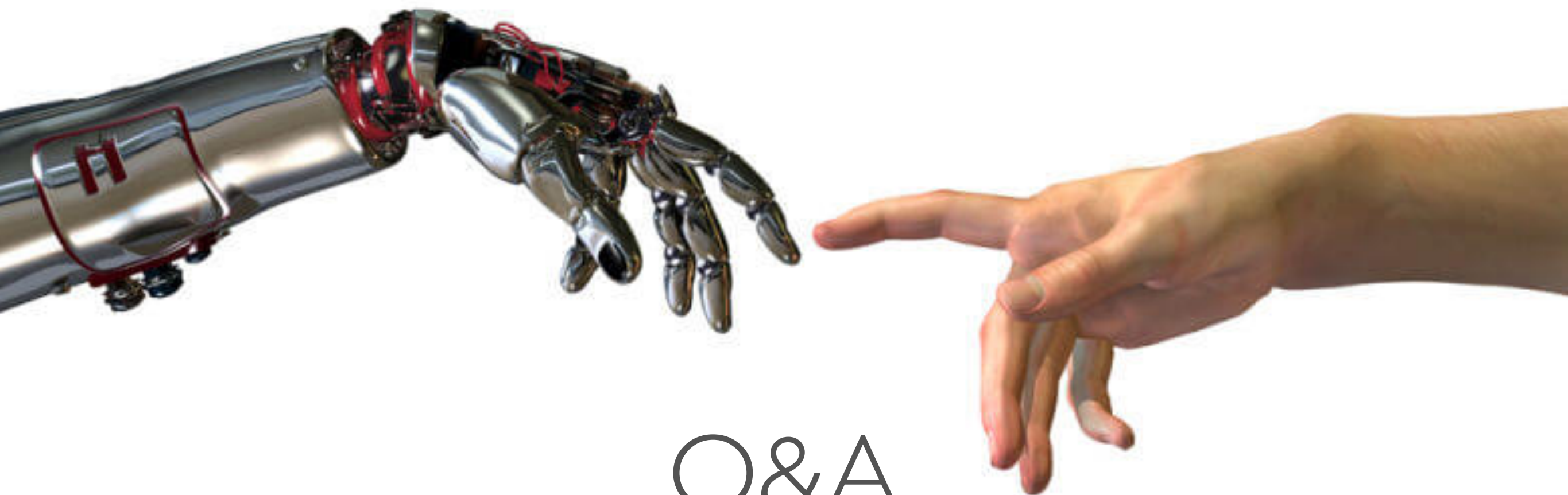


maybe you didn't ?

GAN



oh my you are you.



Q&A

Thank you for your attention!