It's a cold Friday morning. You had to wake up early to attend a class at your university. It started at 9:15AM. There is an ongoing presentation about a research paper on Learning Dynamic Belief Graphs to Generalize on Text-Based Games. It is presented by Michal Bacigal. What are you going to do?
It's a cold Friday morning. You had to wake up early to attend a class at your university. It started at 9:15AM. There is an ongoing presentation about a research paper on Learning Dynamic Belief Graphs to Generalize on Text-Based Games. It is presented by Michal Bacigal. What are you going to do?

> start listening
(A REALLY BRIEF LOOK AT)
REINFORCEMENT LEARNING
(A REALLY BRIEF LOOK AT)

REINFORCEMENT LEARNING

- a way for intelligent agents to learn how to perform tasks
(A REALLY BRIEF LOOK AT)

REINFORCEMENT LEARNING

• a way for intelligent agents to learn how to perform tasks

• the agent observes the environment and alters its state through actions
REINFORCEMENT LEARNING

- a way for intelligent agents to learn how to perform tasks
- the agent observes the environment and alters its state through actions
- the 'quality of actions' is evaluated through 'rewards'
(A REALLY BRIEF LOOK AT)

REINFORCEMENT LEARNING

- a way for intelligent agents to learn how to perform tasks
- the agent observes the environment and alters its state through actions
- the 'quality of actions' is evaluated through 'rewards'
- examples: gaming, robotics, self-driving cars and more!
(A REALLY BRIEF LOOK AT) REINFORCEMENT LEARNING

• a way for intelligent agents to learn how to perform tasks
• the agent observes the environment and alters its state through actions
• the ‘quality of actions’ is evaluated through ‘rewards’
• examples: gaming, robotics, self-driving cars and more!
REINFORCEMENT LEARNING

- A way for intelligent agents to learn how to perform tasks
- The agent observes the environment and alters its state through actions
- The 'quality of actions' is evaluated through 'rewards'
- Examples: gaming, robotics, self-driving cars and more!
(A REALLY BRIEF LOOK AT)

REINFORCEMENT LEARNING

• a way for intelligent agents to learn how to perform tasks
• the agent observes the environment and alters its state through actions
• the 'quality of actions' is evaluated through 'rewards'
• examples: gaming, robotics, self-driving cars and more!
REINFORCEMENT LEARNING
(FOR TEXT-BASED GAMES)
REINFORCEMENT LEARNING
(for text-based games)

AGENT
REINFORCEMENT LEARNING
(FOR TEXT-BASED GAMES)

AGENT

ENVIRONMENT
You are in a room. There is a man and his dog. The man asks you to take his dog out. You are carrying a dog leash and a shotgun. What do you do?
You are in a room. There is a man and his dog. The man asks you to take his dog out. You are carrying a dog leash and a shotgun. What do you do?
You are in a room. There is a man and his dog. The man asks you to take his dog out. You are carrying a dog leash and a shotgun. What do you do?
MICROSOFT TEXTWORLD
You are hungry! Let's cook a delicious meal. Check the cookbook in the kitchen for the recipe. Once done, enjoy your meal!

-- Backyard --
You find yourself in a backyard.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn't TextWorld just the best? The patio chair is stylish. But there isn't a thing on it. You see a glimpse over in a corner, where you can see a BBQ.

There is a closed screen door leading south. There is a closed wooden door leading west. There is an exit to the east. Don't worry, there is no door.

>go west
You have to open the wooden door first.

>open wooden door
You open wooden door.

>go west

-- Shed --
Welcome to the shed. You can barely contain your excitement.

You can make out a closed toolbox here. You can see a workbench. The workbench is wooden. Looks like someone's already been here and taken everything off it, though. You swear loudly.

There is an open wooden door leading east.

>
MICROSOFT TEXTWORLD

You are hungry! Let's cook a delicious meal. Check the cookbook in the kitchen for the recipe. Once done, enjoy your meal!

← Backyard →
You find yourself in a backyard.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn't TextWorld just the best? The patio chair is stylish. But there isn't a thing on it. You see a Glenn over in a corner, where you can see a BBQ.

There is a closed screen door leading south. There is a closed wooden door leading west. There is an exit to the east. Don't worry, there is no door.

>go west
You have to open the wooden door first.

>open wooden door
You open wooden door.

>go west

← Shed →
Welcome to the shed. You can barely contain your excitement.

You can make out a closed toolbox here. You can see a workbench. The workbench is wooden. Looks like someone's already been here and taken everything off it, though. You swear loudly.

There is an open wooden door leading east.
MICROSOFT TEXTWORLD

You are hungry! Let’s cook a delicious meal. Check the cookbook in the kitchen for the recipe. Once done, enjoy your meal!

--- Backyard ---
You find yourself in a backyard.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn’t TextWorld just the best? The patio chair is styilish. But there isn’t a thing on it. You see a gleam over in a corner, where you can see a BBQ.

There is a closed screen door leading south. There is a closed wooden door leading west. There is an exit to the east. Don’t worry, there is no door.

> go west
You have to open the wooden door first.

> open wooden door
You open wooden door.

> go west

--- Shed ---
Welcome to the shed. You can barely contain your excitement.

You can make out a closed toolbox here. You can see a workbench. The workbench is wooden. Looks like someone’s already been here and taken everything off it, though. You swear loudly.

There is an open wooden door leading east.

• an open-source text-based game generator for training reinforcement agents, developed by Microsoft

• allows one to generate random worlds with various levels of complexity, or handcraft a custom world to test specific skills
You are hungry! Let’s cook a delicious meal. Check the cookbook in the kitchen for the recipe. Once done, enjoy your meal!

→ Backyard ←
You find yourself in a backyard.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn’t TextWorld just the best? The patio chair is stylin’. But there isn’t a thing on it. You see a gleam over in a corner, where you can see a BBQ.

There is a closed screen door leading south. There is a closed wooden door leading west. There is an exit to the east. Don’t worry, there is no door.

>go west
You have to open the wooden door first.

>open wooden door
You open wooden door.

>go west

→ shed ←
Welcome to the shed. You can barely contain your excitement.

You can make out a closed toolbox here. You can see a workbench. The workbench is wooden. Looks like someone’s already been here and taken everything off it, though. You swear loudly.

There is an open wooden door leading east.

• an open-source text-based game generator for training reinforcement agents, developed by Microsoft

• allows one to generate random worlds with various levels of complexity, or handcraft a custom world to test specific skills

• available to try at aka.ms/textworld
STATE-OF-THE-ART
USING TRANSFORMER DEEP RECURRENT Q-NETWORKS
STATE-OF-THE-ART USING TRANSFORMER DEEP RECURRENT Q-NETWORKS
THE UNDERLYING IDEA
THE UNDERLYING IDEA

- natural language processing is easy for humans, but not for computers
THE UNDERLYING IDEA

- natural language processing is easy for humans, but not for computers

- how do we ‘represent’ the environment in a way that helps agents perform better?
THE UNDERLYING IDEA

- natural language processing is easy for humans, but not for computers

- how do we ‘represent’ the environment in a way that helps agents perform better?

- low-level vs. high-level reasoning (computers vs. humans)
THE UNDERLYING IDEA

- natural language processing is easy for humans, but not for computers

- how do we 'represent' the environment in a way that helps agents perform better?

- low-level vs. high-level reasoning (computers vs. humans)

- if we could find a way to interpret at a higher level, could we increase the performance?
A HIGHER-LEVEL REASONING USING KNOWLEDGE GRAPHS
You find yourself in a backyard. You make out a patio table. You see a patio chair. There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the west.
A HIGHER-LEVEL REASONING USING KNOWLEDGE GRAPHS

You find yourself in a backyard. You make out a patio table. You see a patio chair. There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the west.
You find yourself in a backyard. You make out a patio table. You see a patio chair. There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the west.
You find yourself in a **backyard**. You make out a **patio table**. You see a **patio chair**. There is a closed **screen door** leading south. There is a closed **wooden door** leading east. There is an **exit** to the west.
You find yourself in a backyard. You make out a patio table. You see a patio chair. There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the west.
You find yourself in a backyard. You make out a patio table. You see a patio chair. There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the west.
You find yourself in a backyard. You make out a patio table. You see a patio chair. There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the west.
You find yourself in a backyard. You make out a patio table. You see a patio chair. There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the west.
MEMORY REPRESENTATION USING

KNOWLEDGE GRAPHS

player

at

backyard

at

patio chair

at

patio table

exit

west of

south of

east of

screen door

wooden door
MEMORY REPRESENTATION USING

KNOWLEDGE GRAPHS

patio chair

player

at

backyard

exit

screen door

east of

south of

west of

patio table

at

wooden door

entity A

relation

entity B
MEMORY REPRESENTATION USING

KNOWLEDGE GRAPHS

(entity A, relation, entity B)
MEMORY REPRESENTATION USING KNOWLEDGE GRAPHS

(entity A, relation, entity B)
GRAPH AIDED TRANSFORMER AGENT
GRAPH AIDED TRANSFORMER AGENT
GRAPH AIDED TRANSFORMER AGENT

GAME
GRAPH AIDED TRANSFORMER AGENT
You find yourself in a backyard.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn't NextWorld just the best? The patio chair is stylish. But there isn't a thing on it. You see a glaze over in a corner, where you can see a MRO.

There is a closed screen door leading south. There is a closed wooden door leading west. There is an exit to the east. Don't worry, there is no door.
You find yourself in a **backyard**.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn't NextWorld just the best? The patio chair is stylish. But there isn't a thing on it. You see a gazebo over in a corner, where you can see a **HUM**.

There is a closed **screen door leading south.** There is a closed wooden door leading west. There is an exit to the east. Don't worry, there is no door.
You find yourself in a **backyard**.

You make out a **patio table**. But the thing is empty. You see a **patio chair**. Wow, isn't 2DtextWorld just the best? The **patio chair** is stylish. But there isn't a thing on it. You see a **glimpse** over in a corner, where you can see a **HOG**.

There is a closed **screen door** leading south. There is a closed **wooden door** leading east. There is an exit to the east. Don't worry, there is no door.
You find yourself in a backyard.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn't TextWorld just the best? The patio chair is stylish. But there isn't a thing on it. You see a gisal over in a corner, where you can see a HUN.

There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the east. Don't worry, there is no door.
You find yourself in a **backyard**.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn't TextWorld just the best? The patio chair is stylish. But there isn't a thing on it. You see a glass over in a corner, where you can see a **MOO**.

There is a closed screen door leading south. There is a closed wooden door leading west. There is an exit to the east. Don't worry, there is no door.
You find yourself in a backyard.
You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn't TextWorld just the best? The patio chair is stylish. But there isn't a thing on it. You see a glimpse over in a corner, where you can see a shed.
There is a closed screen door leading south. There is a closed wooden door leading east. There is an exit to the east. Don't worry, there is no door.

Welcome to the shed. You can barely contain your excitement.

You can make out a closed toolbox here. You can see a workbench. The workbench is wooden. Looks like someone's already been here and taken everything off it, though. You swear loudly.

There is an open wooden door leading east.
You find yourself in a **backyard**.

You make out a patio table. But the thing is empty. You see a patio chair. Wow, isn’t &textWorld just the best? The *patio chair* is stylish. But there isn’t a thing on it. You see a glimpse over in a corner, where you can see a **shed**.

There is a closed **screen door** leading south. There is a closed wooden door leading east. There is an exit to the east. Don’t worry, there is no door.

---

Welcome to the **shed**. You can barely contain your excitement.

You can make out a closed **tool box** here. You can see a workbench. The **workbench** is wooden, looks like someone’s already been here and taken everything off it, though. You swear loudly.

There is an open wooden door leading east.
A DEEPER LOOK AT THE

GRAPH UPDATER
A DEEPER LOOK AT THE GRAPH UPDATER

[Diagram showing the Graph Updater process with nodes representing text, graph encoders, and operations like $f_d$, $f_{\Delta}$, and $\Delta g_t$]
A DEEPER LOOK AT THE GRAPH UPDATER

TEXT INPUT FROM THE PREVIOUS STEP

$O_t, A_{t-1}$

$O_t, A_{t-1} \rightarrow$ Text Encoder

Text Encoder

$O_t, A_{t-1}$

$h_{t-1}$

$f_d$

Graph Encoder

$G_{t-1}$

$h_{t-1}$

$f_d$

$G_t$

$\Delta g_t$
A DEEPER LOOK AT THE
GRAPH UPDATER

TEXT INPUT FROM THE PREVIOUS STEP

ACTION FROM THE PREVIOUS STEP

\[ O_t, A_{t-1} \rightarrow Text Encoder \]

\[ h_{t-1} \rightarrow f_d \rightarrow Graph Encoder \]

\[ f_{\Delta} \rightarrow \Delta g_t \]

\[ h_{t-1} \rightarrow h_t \rightarrow f_d \rightarrow G_t \]
A DEEPER LOOK AT THE GRAPH UPDATER

TEXT INPUT FROM THE PREVIOUS STEP
ACTION FROM THE PREVIOUS STEP
A TRANSFORMER CREATES WORD EMBEDDINGS

\[ O_t, A_{t-1} \]
\[ h_{t-1} \]
\[ G_{t-1} \]
A DEEPER LOOK AT THE GRAPH UPDATER

- TEXT INPUT FROM THE PREVIOUS STEP
- ACTION FROM THE PREVIOUS STEP
- A TRANSFORMER CREATES WORD EMBEDDINGS

Graph Updater Diagram:

- $O_t, A_{t-1} \rightarrow$ Text Encoder
- $h_{t-1} \rightarrow f_d \rightarrow$ Graph Encoder
- $\Delta g_t \rightarrow f_d \rightarrow G_t$

- Hidden RNN Layer from Previous Step
A DEEPER LOOK AT THE GRAPH UPDATER

TEXT INPUT FROM THE PREVIOUS STEP
ACTION FROM THE PREVIOUS STEP
A TRANSFORMER CREATES WORD EMBEDDINGS

\[ O_t, A_{t-1} \]
\[ h_{t-1} \]
\[ f_d \]
\[ G_{t-1} \]
\[ f_d \]
\[ h_t \]
\[ \Delta g_t \]
\[ f_\Delta \]

HIDDEN STATE
HIDDEN RNN
DECODER (MLP)
LAYER FROM
PREVIOUS STEP

GRAPH UPDATER
A DEEPER LOOK AT THE GRAPH UPDATE

TEXT INPUT FROM THE PREVIOUS STEP

ACTION FROM THE PREVIOUS STEP

A TRANSFORMER CREATES WORD EMBEDDINGS

GRAPH FROM THE PREVIOUS STEP

HIDDEN RNN LAYER FROM PREVIOUS STEP

HIDDEN STATE DECODER (MLP)

GRAPH UPDATER
A Deeper Look at the Graph Updater

Text Input from the previous step

Action from the previous step

A Transformer creates word embeddings

A Transformer encodes the graph

Hidden RNN layer from previous step

Hidden state decoder (MLP)

Graph from the previous step
A Deeper Look at the Graph Updater

- Text Input from the Previous Step
- Action from the Previous Step
- A Transformer Creates Word Embeddings
- Concatenator Aggregates Embeddings

Graph Updater:

- Hidden State MLP
- Hidden RNN Layer from Previous Step
- Graph from Previous Step
- A Transformer Encodes the Graph

Diagram:

- $O_t, A_{t-1}$
- $h_{t-1}$
- $G_{t-1}$
- $f_d$
A DEEPER LOOK AT THE
GRAPH UPDATER

TEXT INPUT FROM THE PREVIOUS STEP
ACTION FROM THE PREVIOUS STEP
A TRANSFORMER CREATES WORD EMBEDDINGS
CONCATENATOR AGGREGATES EMBEDDINGS

HIDDEN STATE MLP
HIDDEN RNN LAYER FROM PREVIOUS STEP
GRAPH FROM THE PREVIOUS STEP
AN UPDATE OF THE GRAPH

O_t, A_{t-1} → Text Encoder
f_d → Graph Encoder
h_{t-1} → f_d

Δg_t → f_d
G_{t-1} → HIDDEN STATE

h_t-1 → h_t → G_t
A DEEPER LOOK AT THE GRAPH UPDATER

- **Text Input from the Previous Step**
- **Action from the Previous Step**
- **A Transformer Creates Word Embeddings**
- **Concatenator Aggregates Embeddings**

### Graph Encoder

- Input: $O_t, A_{t-1}$
- Output: $f_d, G_{t-1}$

### Text Encoder

- Input: $O_t, A_{t-1}$
- Output: $f_d$

### Hidden State Decoder (MLP)

- Input: $h_{t-1}$
- Output: $f_d$

### Graph Function Operator

- $\Delta g_t = f_\Delta(h_{G_{t-1}}, h_{O_t}, h_{A_{t-1}})$
- $h_t = \text{RNN}(\Delta g_t, h_{t-1})$
- $G_t = f_d(h_t)$
A DEEPER LOOK AT THE
GRAPH UPDATER

TEXT INPUT
FROM THE
PREVIOUS STEP

ACTION FROM
THE PREVIOUS
STEP

A TRANSFORMER
CREATES WORD
EMBEDDINGS

CONCATENATOR
AGGREGATES
EMBEDDINGS

AN UPDATED
GRAPH

\[ O_t, A_{t-1} \rightarrow Text \ Encoder \]

\[ h_{t-1} \rightarrow Graph \ Encoder \]

\[ \Delta g_t = f_{\Delta}(h_{G_{t-1}}, h_{O_t}, h_{A_{t-1}}) \]

\[ h_t = RNN(\Delta g_t, h_{t-1}) \]

\[ G_t = f_{d}(h_t) \]
A DEEPER LOOK AT THE
ACTION SELECTOR
A DEEPER LOOK AT THE ACTION SELECTOR

Diagram:

- $O_t$ → Text Encoder
- $G_t$ → Graph Encoder
- $C_t$ → Text Encoder
- Representation Aggregator
- Scorer
- $A_t$
A DEEPER LOOK AT THE ACTION SELECTOR

TEXT FROM THE CURRENT STEP

ACTION SELECTOR

$O_t \rightarrow \text{Text Encoder} \rightarrow \text{Representation Aggregator} \rightarrow \text{Scorer} \rightarrow A_t$

$G_t \rightarrow \text{Graph Encoder} \rightarrow C_t \rightarrow \text{Text Encoder} \rightarrow \text{Scorer} \rightarrow A_t$
A DEEPER LOOK AT THE ACTION SELECTOR

TEXT FROM THE CURRENT STEP

A TRANSFORMER CREATES WORD EMBEDDINGS

ACTION SELECTOR

\[
O_t \quad \text{Text Encoder} \quad \text{Representation Aggregator} \quad \text{Scorer} \quad A_t
\]

\[
G_t \quad \text{Graph Encoder} \quad C_t \quad \text{Text Encoder}
\]
A DEEPER LOOK AT THE ACTION SELECTOR

A transformer creates word embeddings.

Text from the current step

Graph from the current step

$O_t$ → Text Encoder → Representation Aggregator → Scorer → $A_t$

$G_t$ → Graph Encoder → Representation Aggregator

$C_t$ → Text Encoder
A DEEPER LOOK AT THE ACTION SELECTOR

A TRANSFORMER ENCODES THE GRAPH FROM THE CURRENT STEP

A TRANSFORMER CREATES WORD EMBEDDINGS

TEXT FROM THE CURRENT STEP

O_t → Text Encoder

G_t → Graph Encoder

C_t → Text Encoder

Representation Aggregator

Scorer → A_t

ACTION SELECTOR
A DEEPER LOOK AT THE ACTION SELECTOR

- Text from the current step
- A Transformer creates word embeddings
- Concatenator aggregates embeddings

Graph from the current step
- A Transformer encodes the graph

- Text encoder
- Graph encoder
- Representation aggregator
- Scorer

Action selector

$O_t$ → Text Encoder → Representation Aggregator → Scorer → $A_t$

$G_t$ → Graph Encoder → Text Encoder → C_t
A DEEPER LOOK AT THE ACTION SELECTION

A TRANSFORMER creates word embeddings
A TRANSFORMER encodes the graph
CONCATENATOR aggregates embeddings

Graph from the current step
Text from the current step
Action candidates

A transformer creates word embeddings
A transformer encodes the graph
Concatenator aggregates embeddings

Action selector

Graph from the current step
Text encoder
Representaion aggregator
Scorer

$O_t$ → Text Encoder
$G_t$ → Graph Encoder
$C_t$ → Text Encoder
$A_t$
A DEEPER LOOK AT THE ACTION SELECTOR

- TEXT FROM THE CURRENT STEP
- A TRANSFORMER CREATES WORD EMBEDDINGS
- CONCATENATOR AGgregates EMBEDDINGS

GRAPH FROM THE CURRENT STEP

- A TRANSFORMER ENCODES THE GRAPH

ACTION CANDIDATES

ACTION QUALITY EVALUATOR

ACTION SELECTOR
A DEEPER LOOK AT THE ACTION SELECTOR

- TEXT FROM THE CURRENT STEP
- A TRANSFORMER CREATES WORD EMBEDDINGS
- CONCATENATOR AGGREGATES EMBEDDINGS
- ACTOR CANDIDATES
- ACTION QUALITY EVALUATOR
- RESULTING ACTION

**Graph from the Current Step**

- $O_t$ is the current text.
- $G_t$ is the graph from the current step.
- $C_t$ is the action candidates.

**Encoder and Representation Aggregator**

- Text Encoder
- Graph Encoder
- Representation Aggregator

**Scorer**

- Scorer outputs the resulting action $A_t$. 

**Diagram Details**

- A Transformer encodes the graph.
- The graph is represented as a concatenated embedding.
Pre-training the Graph Updater

Graph Updater (§3.2)

- $h_{t-1}$
- $A_{t-1}$
- $O_t$
- $A_t$
- $h_t$

Text Encoder

- $A_{t-1}$

Graph Encoder

- $G_t$

Representation Aggregator
PRE-TRAINING THE GRAPH UPDATER

Graph Updater (§3.2)

- $h_{t-1}$
- $A_{t-1}$
- $O_t$
- $A_t$

Text Encoder

- $h_t$
- $G_t$

Graph Encoder

Representation Aggregator

Observation Generator

$O_T$
PRE-TRAINING THE GRAPH UPDATER
PRE-TRAINING THE GRAPH UPDATER

Graph Updater (§3.2) → Text Encoder → Representation Aggregator

O_t
A_{t-1}

h_{t-1}
A_{t-1}

G_t

O_{T}

DISCRIMINATOR

T/F

O_{T}

O_{\hat{T}}

O_{\bar{T}}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}

O_{T}
INTERNAL REPRESENTATION OF BELIEF GRAPHS
INTERNAL REPRESENTATION OF BELIEF GRAPHS

GROUND TRUTH
INTERNAL REPRESENTATION OF BELIEF GRAPHS

GROUND TRUTH

GATA-COC
EXPERIMENTS
EXPERIMENTS

DIFFICULTY

4 + 1

LEVELS
EXPERIMENTS

DIFFICULTY
4+1

LEVELS

TRAIN/VALIDATION/TEST
SPLIT RATIO
100/20/20
EXPERIMENTS

DIFFICULTY
4+1
LEVELS

TRAIN/VALIDATION/TEST SPLIT RATIO
100/20/20

TOP PERFORMER ON THE VALIDATION SETS IS EVALUATED ON TEST DATA
EXPERIMENTS

DIFFICULTY
4+1

LEVELS

TRAIN/VALIDATION/TEST
SPLIT RATIO
100/20/20

TOP PERFORMER ON THE
VALIDATION SETS IS
EVALUATED ON TEST DATA

MODELS

Tr-DQN
Tr-DRQN
Tr-DRQN+
GATA
GATA-GTP
GATA-GTF
DIFFICULTY
4+1 LEVELS

TRAIN/VALIDATION/TEST SPLIT RATIO
100/20/20

TOP PERFORMER ON THE VALIDATION SETS IS EVALUATED ON TEST DATA

RESULTS
Tr-DRQN(+) outperforms Tr-DQN by roughly 10 pct. on smaller and 4 pct. on larger datasets.

GATA steadily outperforms text-based baselines, with additional improvements if text observations are available.

GATA-GTP outperforms text-based baselines, but has poorer performance than GATA.

GATA-GTF outperforms both text-based baselines and GATA, including GATA-GTP.

MODELS
Tr-DQN
Tr-DRQN
Tr-DRQN+
GATA
GATA-GTP
GATA-GTF
DIFFICULTY
4+1 LEVELS

MODELS
Tr-DQN
Tr-DRQN
Tr-DRQN+
GATA
GATA-GTP
GATA-GTF

TRAIN/VALIDATION/TEST
SPLIT RATIO
100/20/20

TOP PERFORMER ON THE
VALIDATION SETS IS
EVALUATED ON TEST DATA

RESULTS

k episodes

GATA w/ OG
GATA w/ COC
GATA-GTP
GATA-GTF
Tr-DQN
Tr-DRQN
Tr-DRQN+
Learning Dynamic Belief Graphs to Generalize on Text-Based Games

Ashutosh Adhikari†* Xingdi Yuan♥* Marc-Alexandre Côté♥*
Mikuláš Zelinka‡ Marc-Antoine Rondeau ♥
Romain Laroche ♥ Pascal Poupart †‖ Jian Tang ♠♠
Adam Trischler ♥ William L. Hamilton ♦

†University of Waterloo ♥ Microsoft Research, Montréal ‖ Charles University
♣ Mila ♦ McGill University ♠ HEC Montréal †‖ Vector Institute
eric.yuan@microsoft.com