# Dense Rewards and Continual Reinforcement Learning for Task-oriented Dialogue Policies

**Christian Geishauser (now at appliedAI Initiative)**

**Dialog Systems and Machine Learning**

**Heinrich Heine University Düsseldorf**

# Outline

- Introduction to task-oriented dialogue and dialogue policies
- Challenges in reinforcement learning for dialogue policies
  1. Sparse reward problem
  2. Absence of continually learning dialogue policies
  3. Absence of realistic continual learning environments
- Conclusion and future works

# Dialogue is Fundamental

- Dialogue is the fundamental way of communcation between humans
- Dialogue topics are infinitely diverse
- We focus on dialogue between a **user** and a **dialogue system**
- The work centers around **task-oriented** dialogue

# Task-oriented Dialogue

- Task-oriented dialogue systems help users to achieve a specific task/goal during interaction
  - Make a hairdresser appointment
  - Find attractions/activities for a trip
  - Find restaurants or hotels to book in town
  - Buy train/bus tickets
  - Set an alarm
  - Organize your calendar
  - Transfer money
  - Get weather information
  - …


- The objective of the dialogue system is to fulfil the user goal

# Ontology



- Task-oriented dialogue systems operate within certain boundaries defined by an ontology

# Ontology

Heinrich Heine
Universität Düsseldorf

- Task-oriented dialogue systems operate within certain boundaries defined by an ontology
  - Domains: restaurant, hotel, train, …

| Domain | Hotel |
|---|---|

# Ontology



- Task-oriented dialogue systems operate within certain boundaries defined by an ontology
  - Domains: restaurant, hotel, train, …
  - Slots within a domain:
    - Hotel: price, area, day, number of people, …

# Ontology



- Task-oriented dialogue systems operate within certain boundaries defined by an ontology
  - Domains: restaurant, hotel, train, …
  - Slots within a domain:
    - Hotel: price, area, day, number of people, …
  - Values for a domain-slot pair:
    - Hotel-area: north, east, west, centre
    - Hotel-price: cheap, moderate, expensive, …

| Domain | Hotel | | |
|---|---|---|---|
| Slot | price | area | day |
| Value | cheap | north | Monday |
| | moderate | east | Tuesday |
| | … | … | … |
| | expensive | south | Sunday |

# Ontology



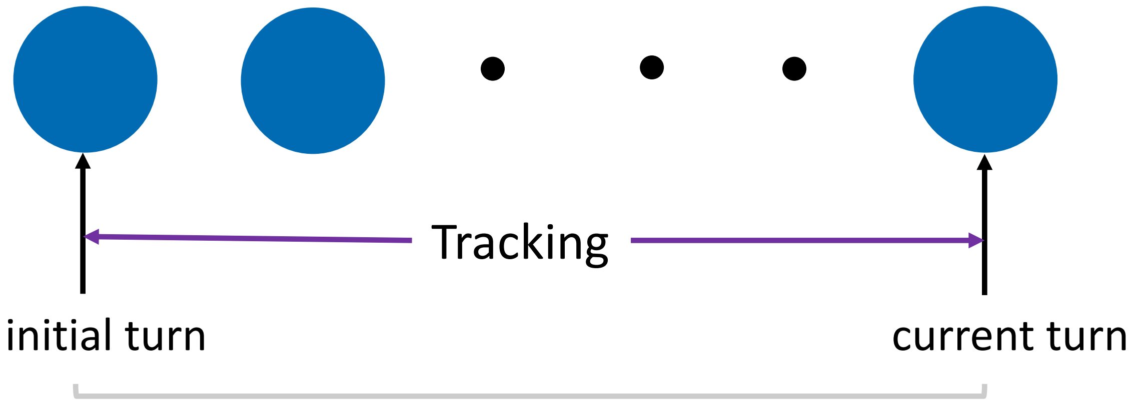
- Task-oriented dialogue systems operate within certain boundaries defined by an ontology
  - Domains: restaurant, hotel, train, …
  - Slots within a domain:
    - Hotel: price, area, day, number of people, …
  - Values for a domain-slot pair:
    - Hotel-area: north, east, west, centre
    - Hotel-price: cheap, moderate, expensive, …
  - Semantic actions:
    - Defined through domain-intent-slot triplets
    - hotel-inform-address, hotel-request-price, …

| Domain | Hotel | | |
|---|---|---|---|
| Slot | price | area | day |
| Value | cheap | north | Monday |
| | moderate | east | Tuesday |
| | … | … | … |
| | expensive | south | Sunday |

- Ontology defines what the system can **understand** (e.g. through domain-slot-value triplets)
- Ontology defines what the system can **say** (semantic actions)

# Tracking and Acting

- Task-oriented dialogue systems require two abilities
  - Maintaining the current state of the dialogue (tracking)

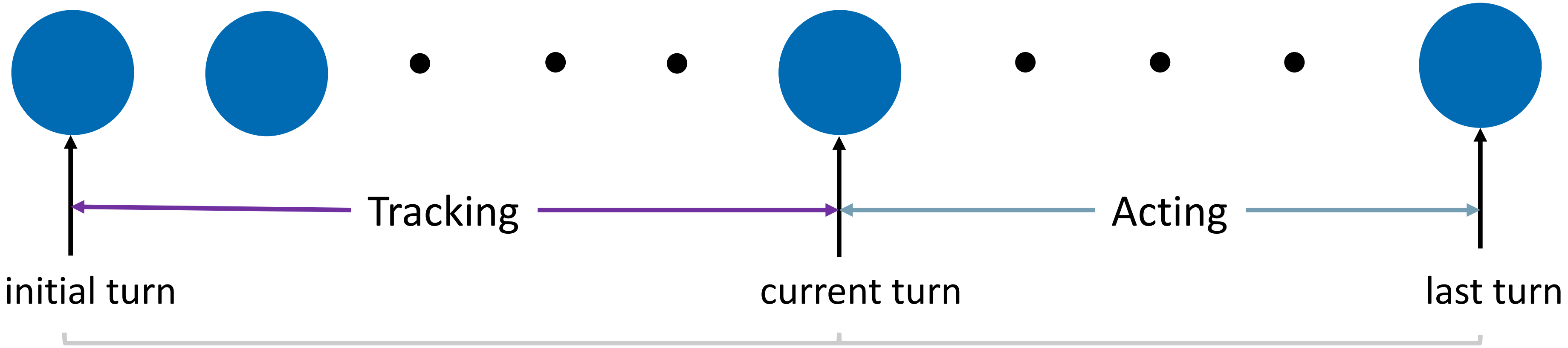



Belief tracker: maintains probability distribution
over values for every domain-slot pair

# Tracking and Acting



- Task-oriented dialogue systems require two abilities
  - Maintaining the current state of the dialogue (tracking)
  - Taking actions that lead to user goal success (acting)




Belief tracker: maintains probability distribution over values for every domain-slot pair

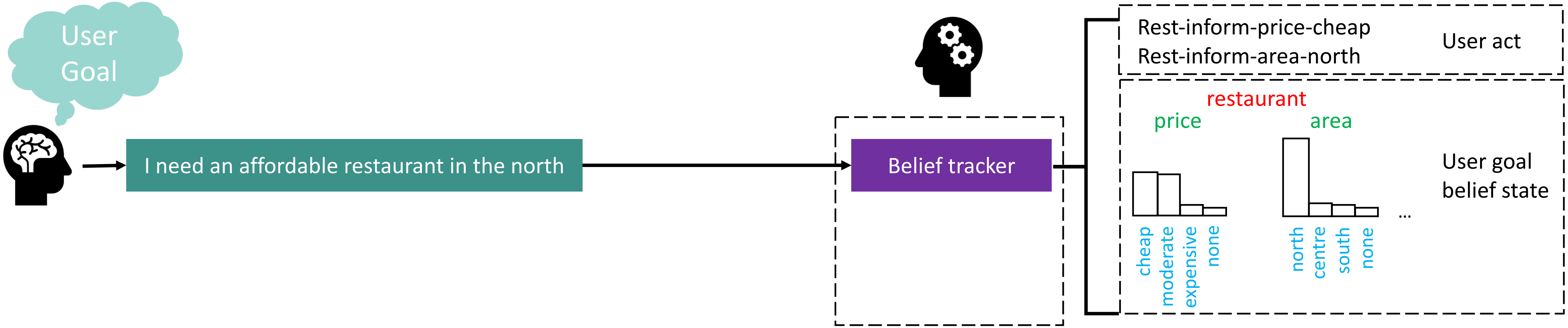
Dialogue policy: take actions in order to steer conversation to task success

# Dialogue as Reinforcement Learning Problem

Heinrich Heine
Universität Düsseldorf

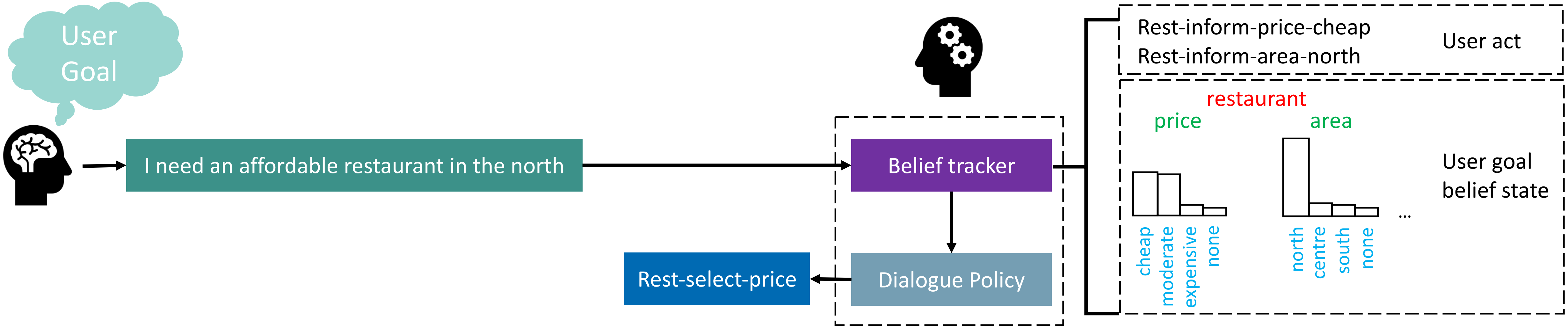

I need an affordable restaurant in the north

# Dialogue as Reinforcement Learning Problem



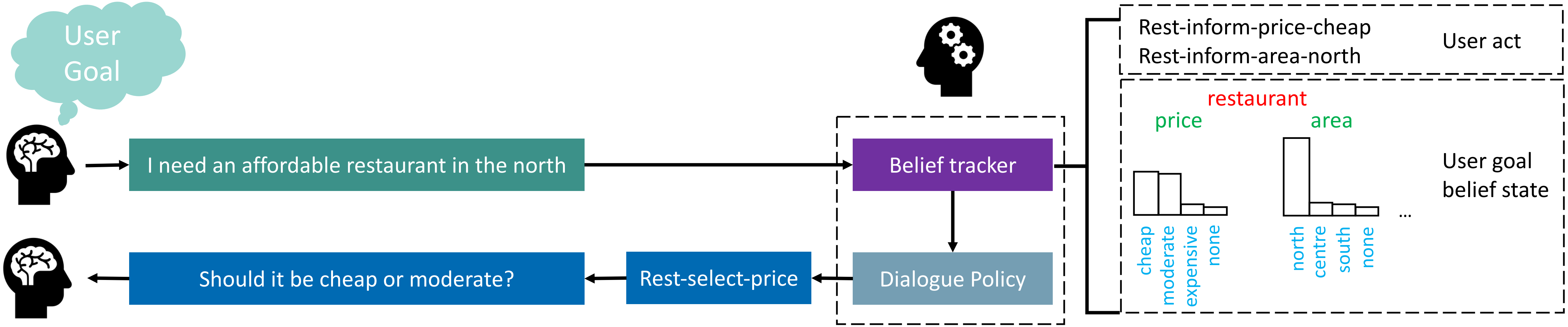
# Dialogue as Reinforcement Learning Problem

hhu
Heinrich Heine
Universität Düsseldorf

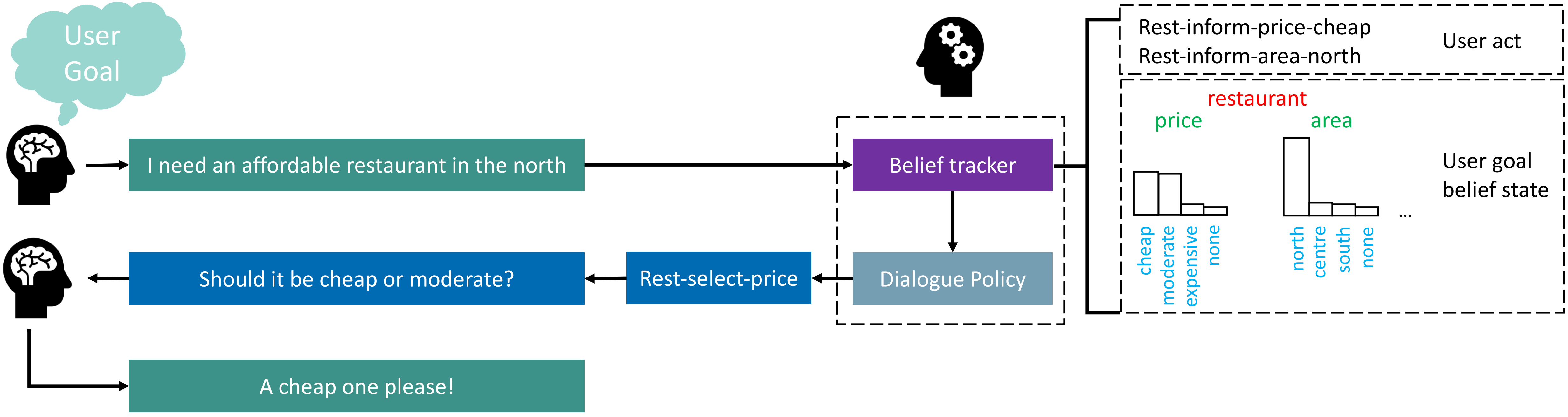


# Dialogue as Reinforcement Learning Problem

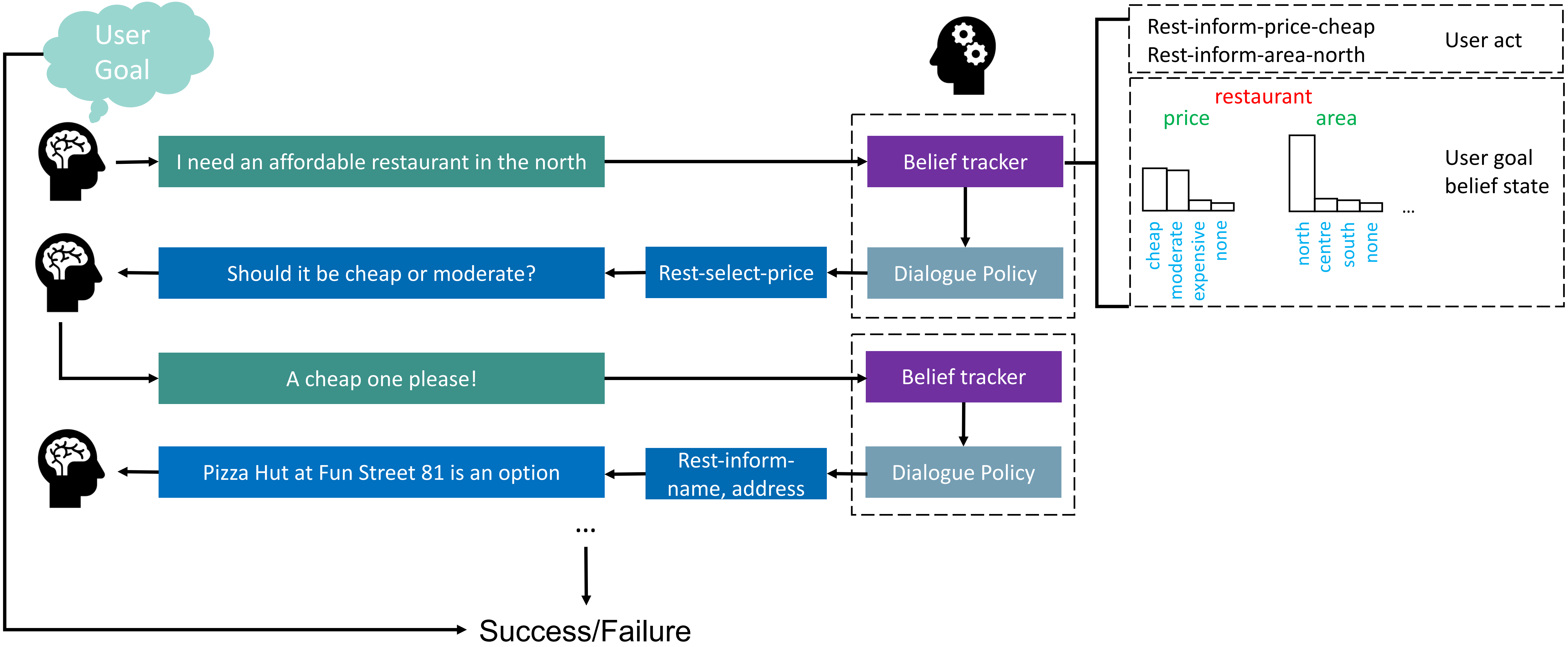
Heinrich Heine Universität Düsseldorf



# Dialogue as Reinforcement Learning Problem



# Dialogue as Reinforcement Learning Problem



# Dialogue as Reinforcement Learning Problem

- The dialogue policy has to solve a sequential decision making problem
  - Find actions that lead to fulfilment of the user goal (dialogue success)

# Dialogue as Reinforcement Learning Problem

- The dialogue policy has to solve a sequential decision making problem
  - Find actions that lead to fulfilment of the user goal (dialogue success)
  - We can optimize the dialogue policy using reinforcement learning (RL)

reward $r_t$, observation

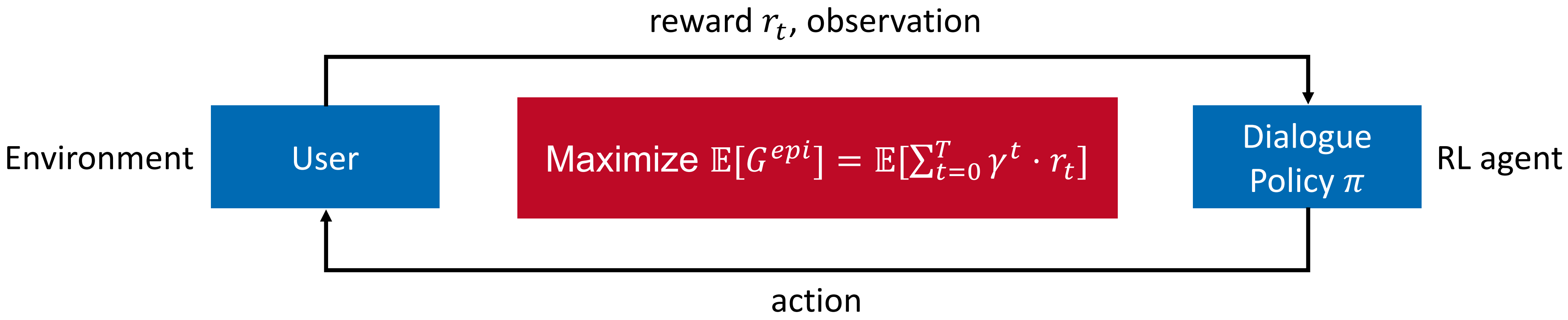
Environment | User | Maximize $\mathbb{E}[G^{epi}] = \mathbb{E}[\sum_{t=0}^{T} \gamma^t \cdot r_t]$ | Dialogue Policy $\pi$ | RL agent

action

# Dialogue as Reinforcement Learning Problem



- The dialogue policy has to solve a sequential decision making problem
  - Find actions that lead to fulfilment of the user goal (dialogue success)
  - We can optimize the dialogue policy using reinforcement learning (RL)

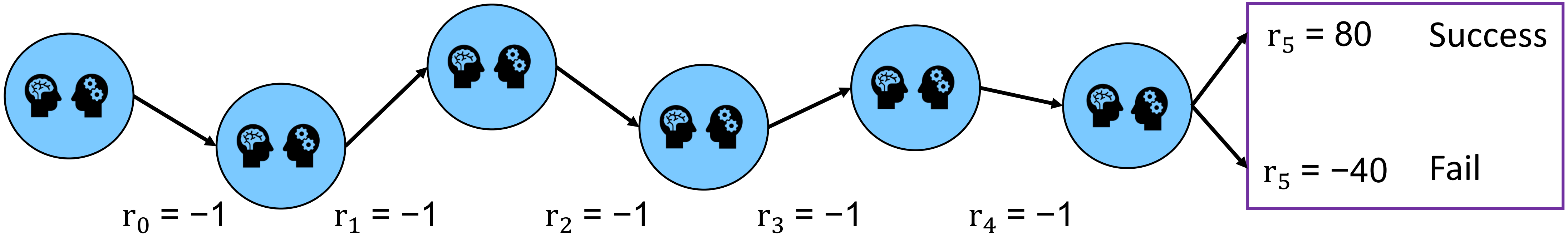



- High positive/negative reward at the end for dialogue success/failure
- Small negative reward (e.g. -1) in every turn for dialogue efficiency

# Problem with the Reward

- Informative reward is only received at the end of the conversation (success/failure)
  → Reward is sparse

- Requires many interactions to find an optimal solution
  - Low sample efficiency due to credit assignment problem



# Dense rewards for dialogue policy optimization

Heinrich Heine
Universität Düsseldorf

How can we obtain dense rewards for sample efficient learning?

# Observation

Heinrich Heine
Universität Düsseldorf

- Dialogue policy needs to know

**What is the user goal?**

User needs a cheap Italian restaurant and a reservation for Saturday, 6 p.m. Requires phone number and address.

**How to solve the goal?**

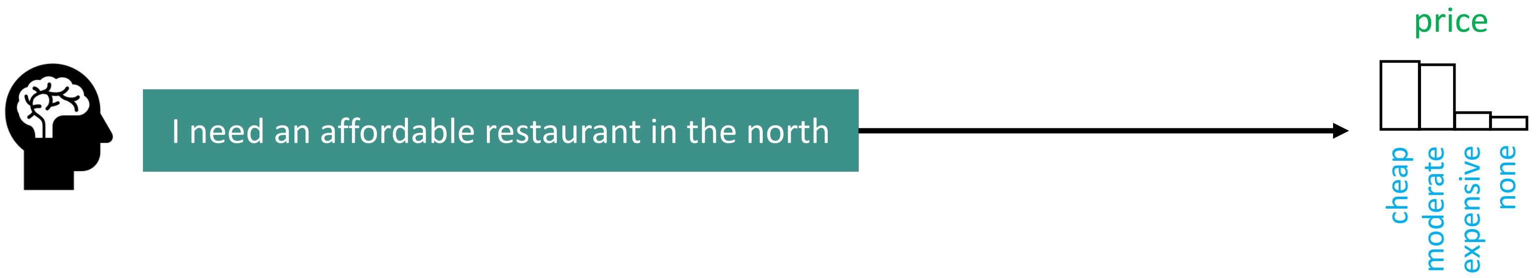
Book restaurant table and inform phone number and address.

- Task can be only solved once the user goal is known
  - Can we encourage behaviour that gathers information about the user goal?

# Proposal: Information Gain

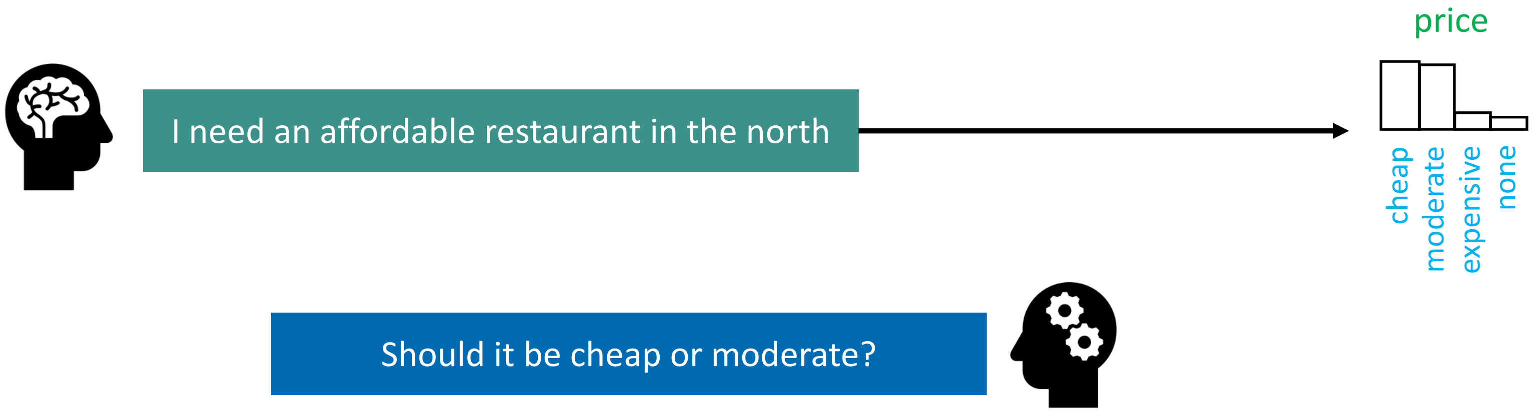
Heinrich Heine Universität Düsseldorf

- We propose **information gain** as reward for solving the sparse reward problem for dialogue
  - Information gain encourages actions that lead to information gathering about the user goal
  - Can be calculated in every turn of the conversation



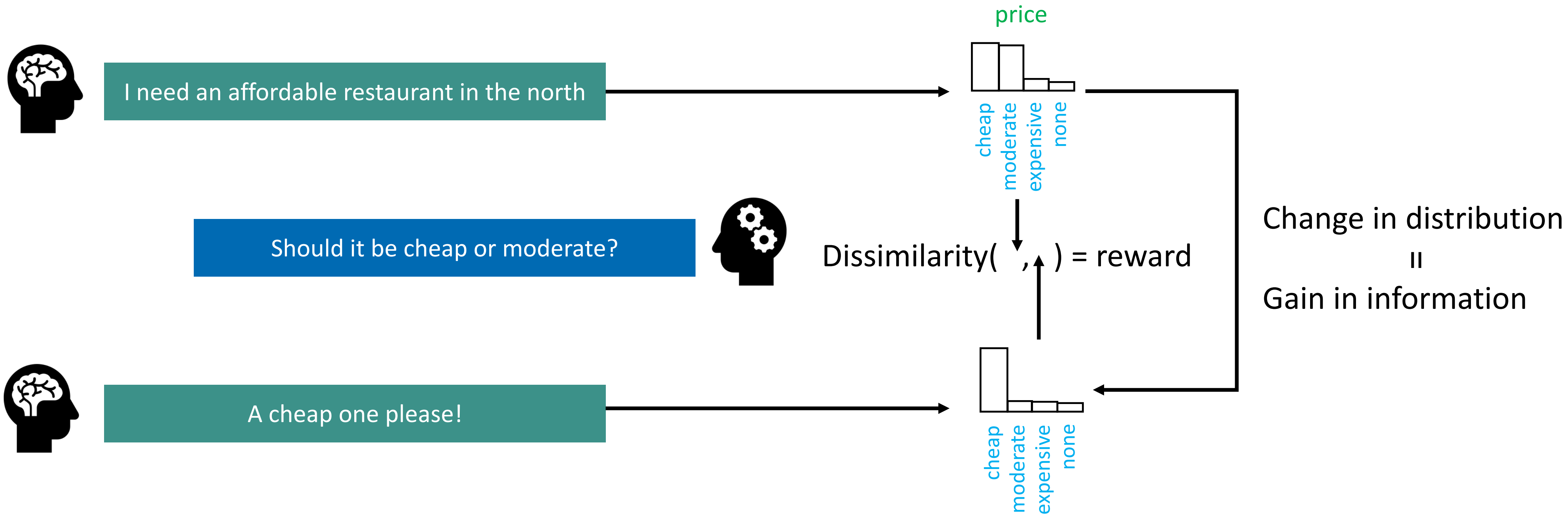
# Proposal: Information Gain

- We propose **information gain** as reward for solving the sparse reward problem for dialogue
  - Information gain encourages actions that lead to information gathering about the user goal
  - Can be calculated in every turn of the conversation



# Proposal: Information Gain

- We propose **information gain** as reward for solving the sparse reward problem for dialogue
  - Information gain encourages actions that lead to information gathering about the user goal
  - Can be calculated in every turn of the conversation



# Experimental Setup



- We leverage the feudal dialogue management architecture (Feudal RL) for experiments

| | Gather information | Solve the task |
|---|---|---|
| | $\pi_{info}$ | $\pi_{general}$ |
| Feudal RL | confirm, request, select, … | inform, book, recommend, … |

# Experimental Setup

- We leverage the feudal dialogue management architecture (Feudal RL) for experiments

| | Gather information | Solve the task |
|---|---|---|
| | $\pi_{info}$ | $\pi_{general}$ |
| Feudal RL | Success reward | Success reward |

# Experimental Setup



- We leverage the feudal dialogue management architecture (Feudal RL) for experiments

| | Gather information | Solve the task |
|---|---|---|
| | $\pi_{info}$ | $\pi_{general}$ |
| Feudal RL | Success reward | Success reward |
| Feudal RL + information gain (ours) | Information gain | Success reward |

- We also compare against STRAC (previous state-of-the-art)

# Experimental Setup

Heinrich Heine Universität Düsseldorf

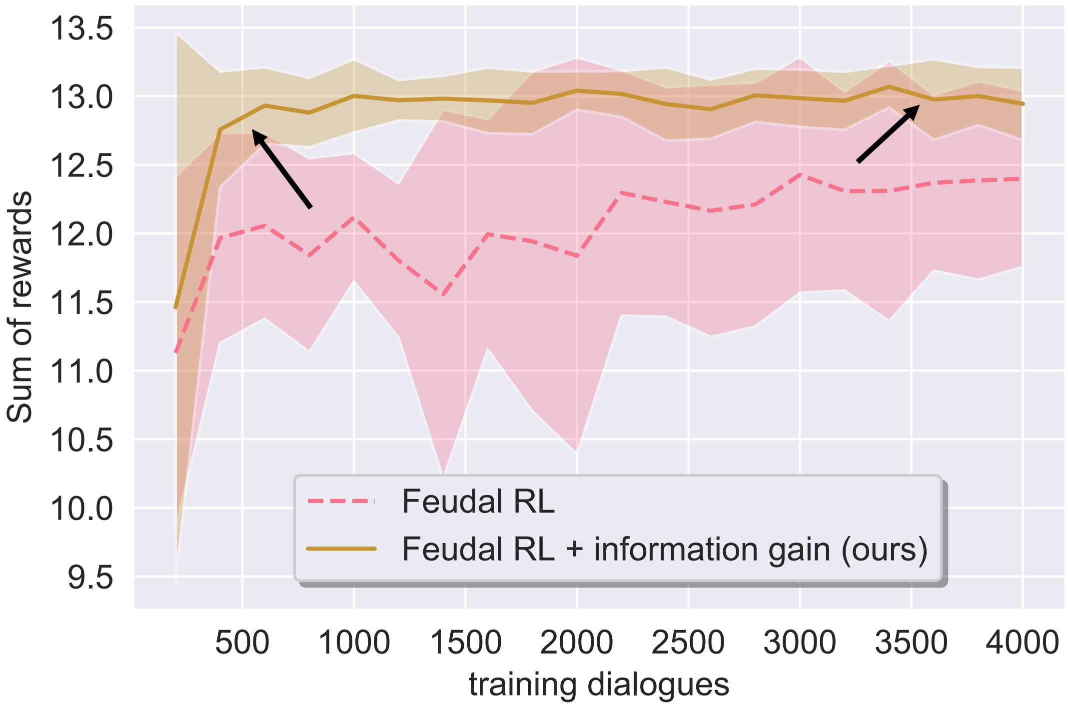
- We test on three different domains
  - Cambridge restaurant (CR)
  - San Francisco restaurant (SFR)
  - Laptops (Lap)

- And different user simulators

# Results



- Feudal RL with information gain improves sample efficiency and final performance



# Results



- Feudal RL with information gain obtains new state-of-the-art performance in terms of sample efficiency and final performance

| Approach | Training dialogues | Success rate | Sum of rewards |
|---|---|---|---|
| STRAC (Chen et. al. 2020) | 400 | 0.83 | 7.6 |
| Feudal RL + info gain (ours) | 400 | **0.89** | **9.5** |
| STRAC (Chen et. al. 2020) | 4000 | 0.93 | 10.7 |
| Feudal RL + info gain (ours) | 4000 | **0.94** | **11.0** |

# Results



- Feudal RL with information gain obtains new state-of-the-art performance in terms of sample efficiency and final performance

| Approach | Training dialogues | Success rate | Sum of rewards |
|---|---|---|---|
| STRAC (Chen et. al. 2020) | 400 | 0.83 | 7.6 |
| Feudal RL + info gain (ours) | 400 | **0.89** | **9.5** |
| STRAC (Chen et. al. 2020) | 4000 | 0.93 | 10.7 |
| Feudal RL + info gain (ours) | 4000 | **0.94** | **11.0** |

# Results



- Feudal RL with information gain obtains new state-of-the-art performance in terms of sample efficiency and final performance

| Approach | Training dialogues | Success rate | Sum of rewards |
|---|---|---|---|
| STRAC (Chen et. al. 2020) | 400 | 0.83 | 7.6 |
| Feudal RL + info gain (ours) | 400 | **0.89** | **9.5** |
| STRAC (Chen et. al. 2020) | 4000 | 0.93 | 10.7 |
| Feudal RL + info gain (ours) | 4000 | **0.94** | **11.0** |

# Results

Heinrich Heine
Universität Düsseldorf

- Interactions with humans show superior performance
  - Feudal RL with information gain asks questions if necessary

| Approach | Success | AskIfNec | Overall |
|---|---|---|---|
| Feudal RL | 0.43 | 3.0 | 2.7 |
| Feudal RL + info gain | **0.71** | **3.8** | **3.7** |

# Results



- Interactions with humans show superior performance
  - Feudal RL with information gain asks questions if necessary

| Approach | Success | AskIfNec | Overall |
|---|---|---|---|
| Feudal RL | 0.43 | 3.0 | 2.7 |
| Feudal RL + info gain | **0.71** | **3.8** | **3.7** |

# Dense Rewards and Continual Reinforcement Learning



✓ Information gain as dense reward for increased sample efficiency

But…

# Dense Rewards and Continual Reinforcement Learning

Heinrich Heine
Universität Düsseldorf

✓ Information gain as dense reward for increased sample efficiency

## But…

- As commonly done, we tested the model on a fixed environment
  - i.e. fixed ontology with fixed amount of domains

Tasks

Hotel
Restaurant
Taxi

# Dense Rewards and Continual Reinforcement Learning

Heinrich Heine Universität Düsseldorf

✓ Information gain as dense reward for increased sample efficiency

But…

- As commonly done, we tested the model on a fixed environment
  - i.e. fixed ontology with fixed amount of domains

What happens if we want to add new domains to the ontology?

Tasks

Hotel
Restaurant
Taxi
**Train**
**Attraction**

# Dense Rewards and Continual Reinforcement Learning



✓ Information gain as dense reward for increased sample efficiency

## But…

- As commonly done, we tested the model on a fixed environment
  - i.e. fixed ontology with fixed amount of domains

What happens if we want to add new domains to the ontology?

Can the system still interact successfully?

Tasks

Hotel
Restaurant
Taxi
**Train**
**Attraction**

# Dense Rewards and Continual Reinforcement Learning

Heinrich Heine
Universität Düsseldorf

✓ Information gain as dense reward for increased sample efficiency

## But…

- As commonly done, we tested the model on a fixed environment
  - i.e. fixed ontology with fixed amount of domains

What happens if we want to add new domains to the ontology?

Tasks

Hotel
Restaurant
Taxi
**Train**
**Attraction**

Can the system still interact successfully?

Can the system continue learning, more like humans?

# Continual Learning

- The world is ever-changing
  - There is a vast amount of tasks a dialogue system can assist with
  - And more are upcoming: Covid vaccination appointments, …

- A dialogue system needs to continue learning to assist with more tasks over time

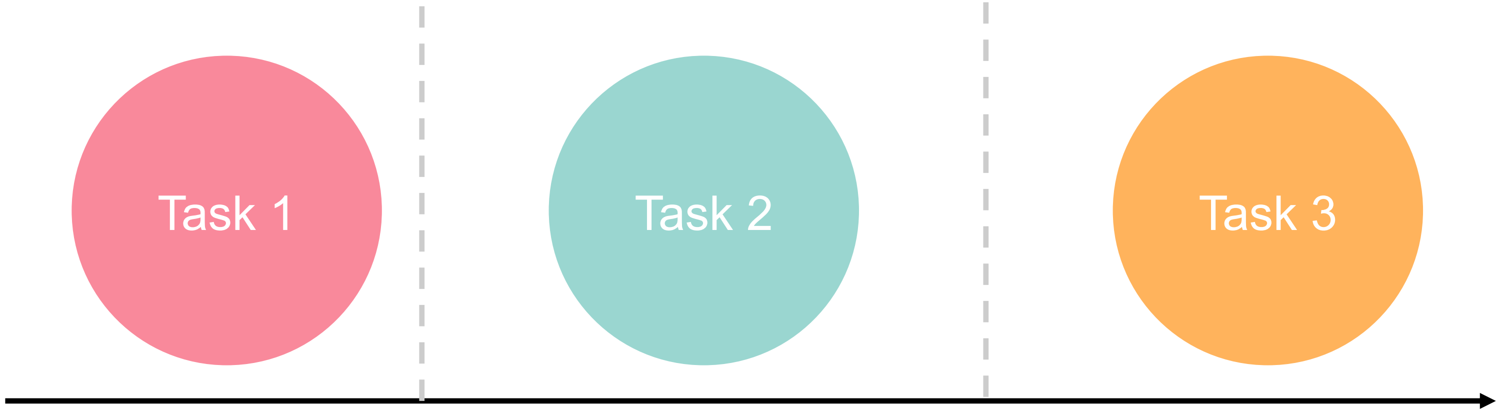
# Continual Learning

- Continual learning focuses on non-stationary, changing environments

# Continual Learning



- Continual learning focuses on non-stationary, changing environments
- Often divided into a set of tasks that need to be completed sequentially



# Continual Reinforcement Learning

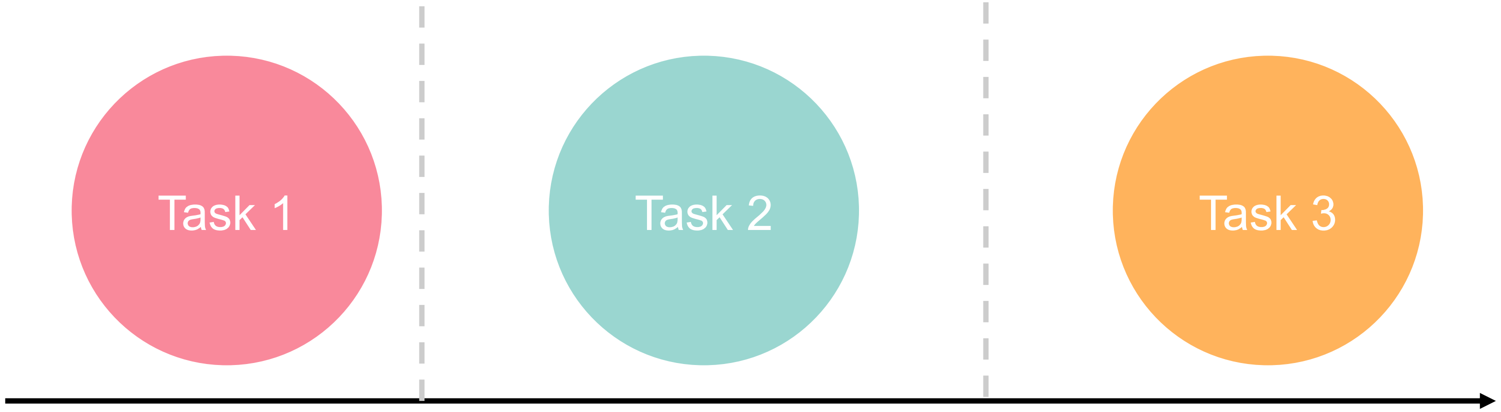
- Continual learning focuses on non-stationary, changing environments
- Often divided into a set of tasks that need to be completed sequentially


- $M_{CRL} = < S(t), A(t), R(t), P(t) >$ (Khetarpal et al. 2022)

# Continual Learning

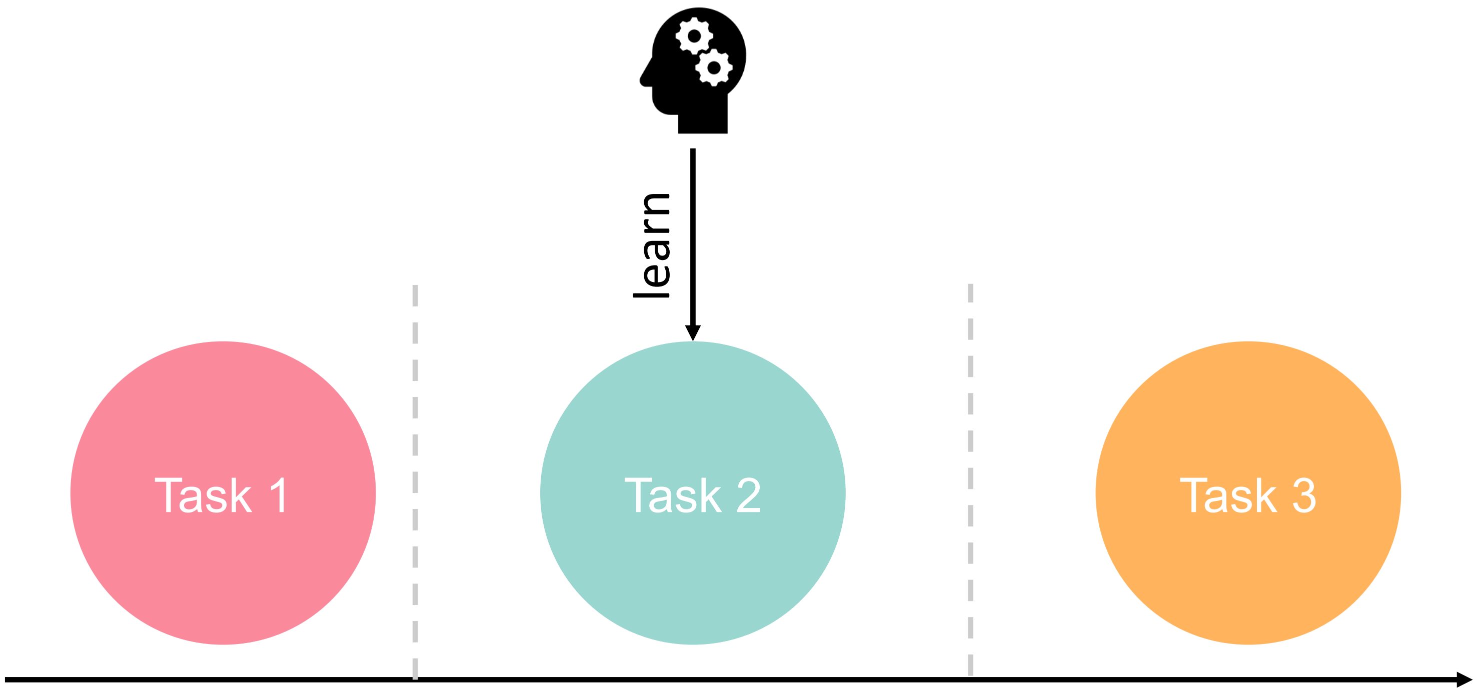
- Continual learning focuses on non-stationary, changing environments

- Often divided into a set of tasks that need to be completed sequentially



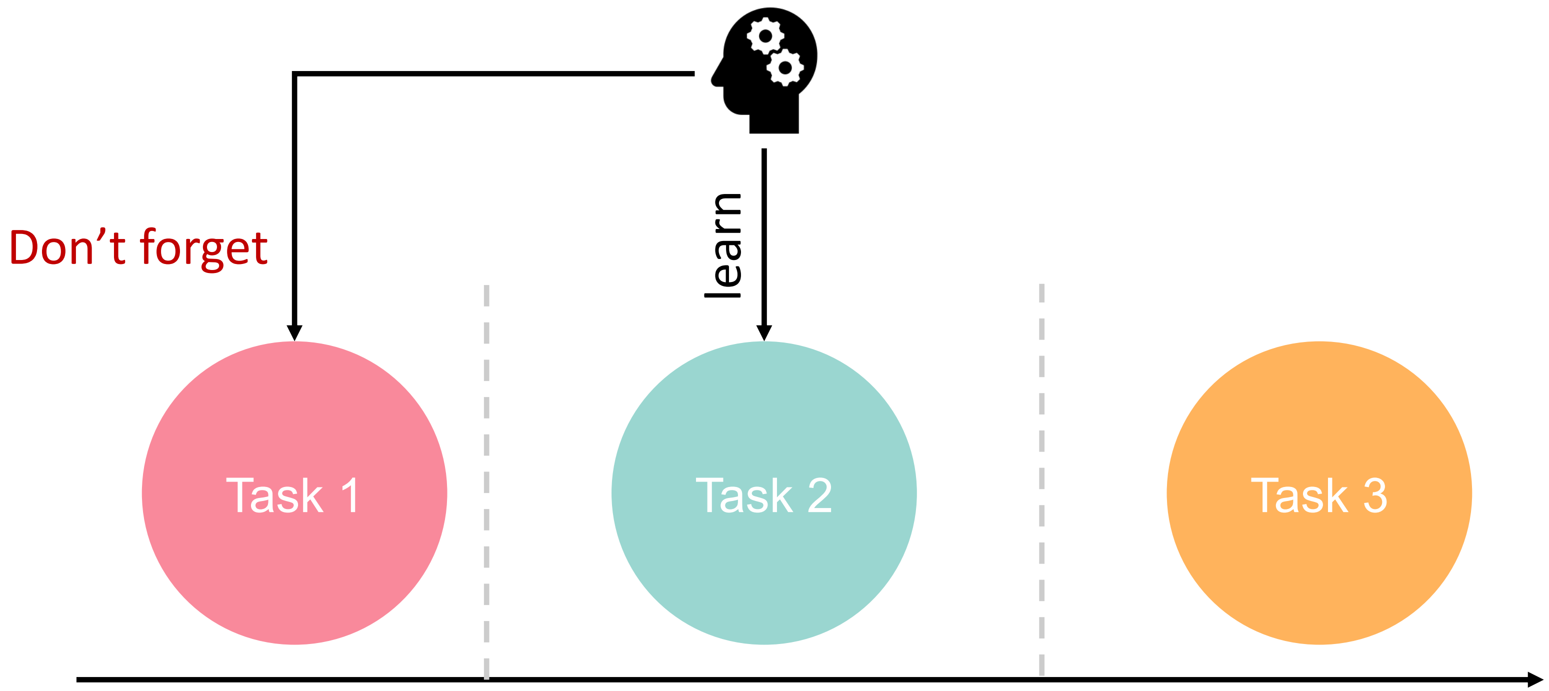

- Compared to multi-task learning, we do not see all tasks at once
- Compared to transfer learning, we still care about previously observed tasks
- Compared to curriculum learning, we have no influence on the task order

# Continual Learning Challenges



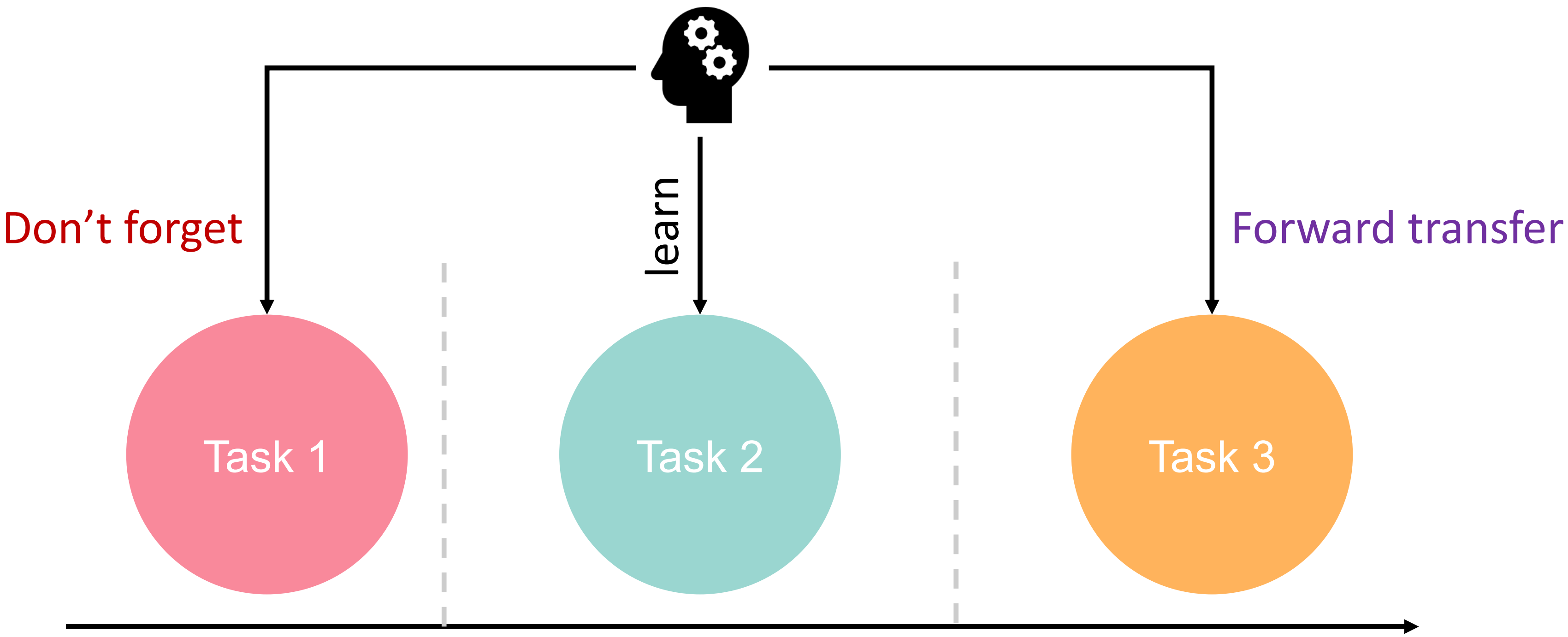
# Continual Learning Challenges

- Forgetting: performance on old tasks should not decrease when learning a new task



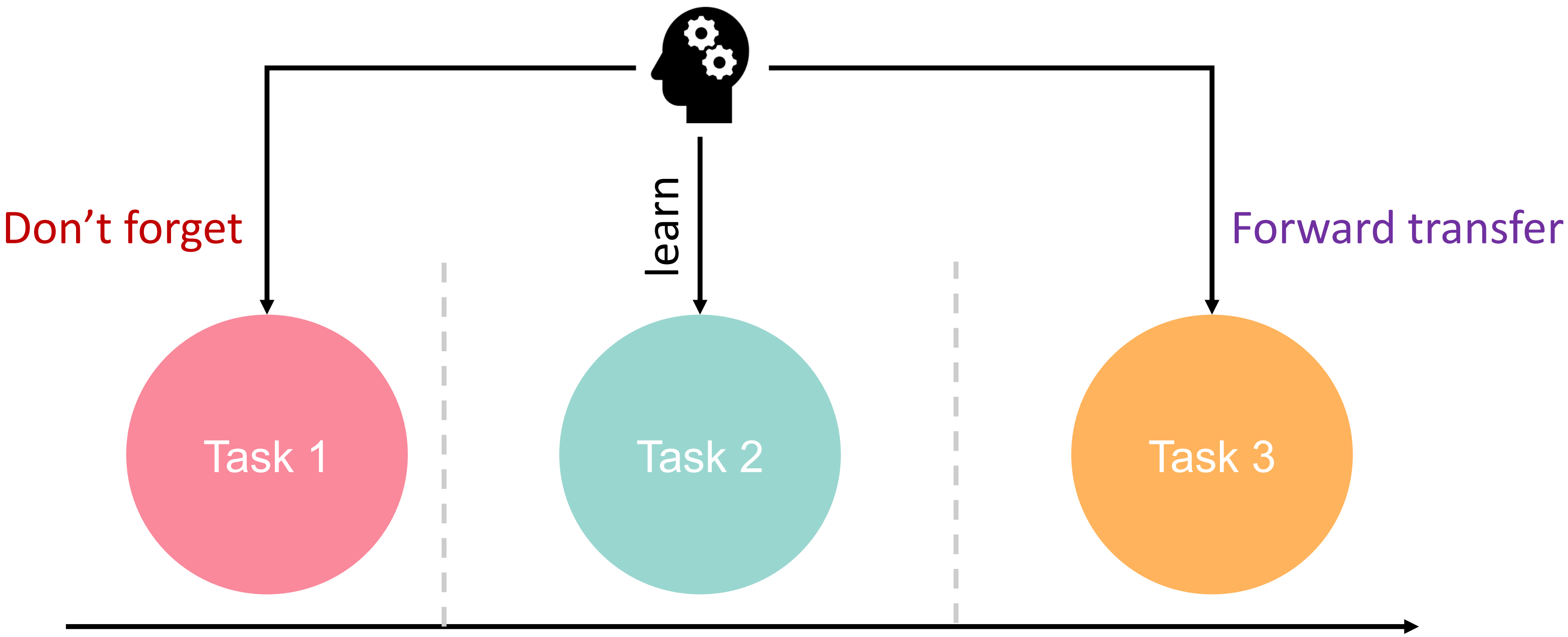
# Continual Learning Challenges

- Forgetting: performance on old tasks should not decrease when learning a new task
- Forward transfer: leverage past knowledge to improve performance on future tasks



# Continual Learning Challenges

- Forgetting: performance on old tasks should not decrease when learning a new task
- Forward transfer: leverage past knowledge to improve performance on future tasks
- Limited resources: learner has only limited model capacity and memory
- …

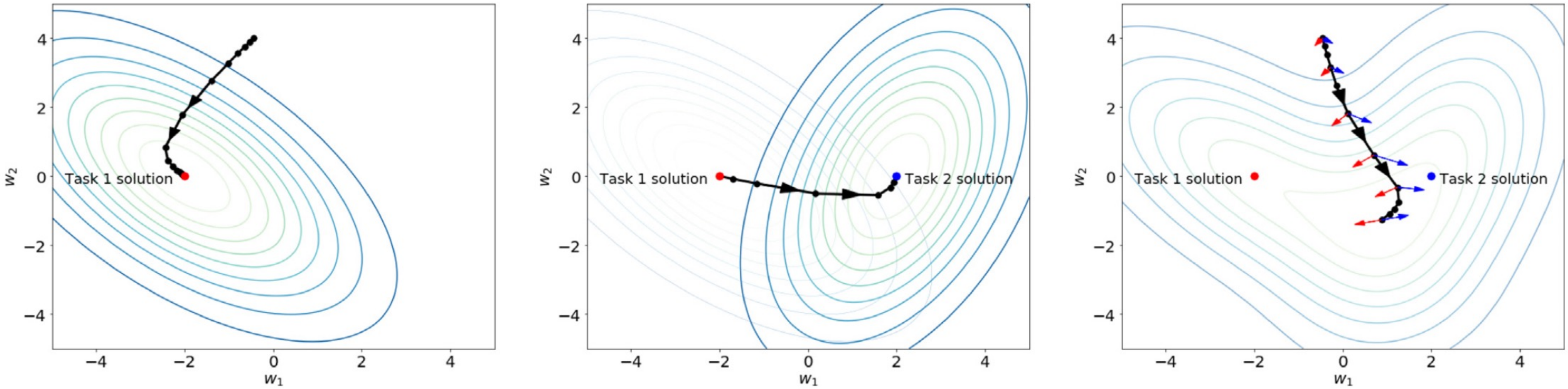


# Continual Learning Challenges

- Forgetting: performance on old tasks should not decrease when learning a new task
- Forward transfer: leverage past knowledge to improve performance on future tasks
- Limited resources: learner has only limited model capacity and memory
- …

- These challenges are competing with each other (stability-plasticity dilemma)

# A note on Catastrophic Forgetting




Embracing Change: Continual Learning in Deep Neural Networks (Hadsell et al. 2020)

# Continual Learning Examples

- Learning to crawl, walk, jump, run, …
- Learning different (programming) languages one after the other
- Learn different sports
- Classifying new image categories over time
- Learning different ATARI games one after the other
- Autonomous driving with changing tire frictions

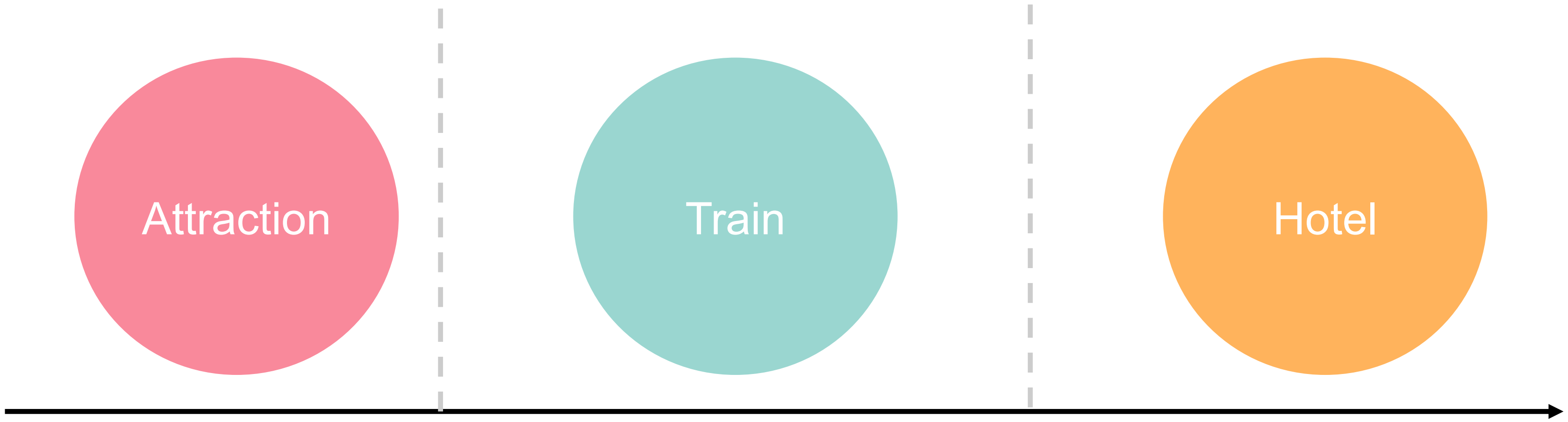
# LLM Training Stages

hhu
Heinrich Heine
Universität Düsseldorf


$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \underbrace{\beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]}_{\text{Prevent forgetting}}$$

# Continual Learning for Task-oriented Dialogue

- In task-oriented dialogue, tasks can be defined as different domains



# Continual Learning for Dialogue Policy

- Recall that an ontology defines what the system can understand and what actions it can take

**Information to comprehend**

| Information | Value |
|---|---|
| Hotel - price | none |
| Hotel - area | north |
| Hotel - request - address | ? |

**Possible actions**

| Actions |
|---|
| Hotel - request - price |
| Hotel - offerbook |
| Hotel - inform - address |

# Continual Learning for Dialogue Policy

Heinrich Heine
Universität Düsseldorf

- Recall that an ontology defines what the system can understand and what actions it can take

**Information to comprehend**

| | Information | Value |
|---|---|---|
| 0 | Hotel - price | none |
| 1 | Hotel - area | north |
| 1 | Hotel - request - address | ? |

**Possible actions**

| Actions | |
|---|---|
| Hotel - request - price | 1 |
| Hotel - offerbook | 0 |
| Hotel - inform - address | 1 |

# Continual Learning for Dialogue Policy

- Recall that an ontology defines what the system can understand and what actions it can take
- Ontology grows as more domains are seen

**Information to comprehend**

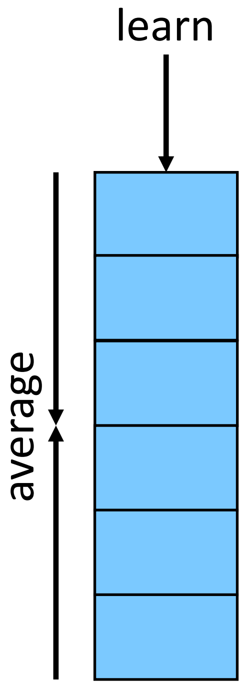
| | Information | Value |
|---|---|---|
| 0 | Hotel - price | none |
| 1 | Hotel - area | north |
| 1 | Hotel - request - address | ? |
| 0 | Train - destination | none |
| 0 | Train - arrival time | none |
| 0 | … | … |

**Possible actions**

| Actions | |
|---|---|
| Hotel - request - price | 1 |
| Hotel - offerbook | 0 |
| Hotel - inform - address | 1 |
| Train - book | 0 |
| Train - recommend - id | 0 |
| … | 0 |

# Continual Learning for Dialogue Policy

- Recall that an ontology defines what the system can understand and what actions it can take
- Ontology grows as more domains are seen

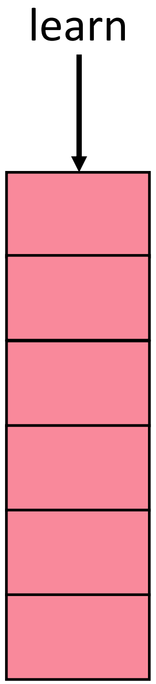



**Information to comprehend**

| Information | Value |
|---|---|
| Hotel - price | none |
| Hotel - area | north |
| Hotel - request - address | ? |
| Train - destination | none |
| Train - arrival time | none |
| … | … |

**Possible actions**

| Actions |
|---|
| Hotel - request - price |
| Hotel - offerbook |
| Hotel - inform - address |
| Train - book |
| Train - recommend - id |
| … |



# Dialogue Policy Challenges



- How can we incorporate **new information and actions** into our model most efficiently?

What is a train destination?

What does train book do?

# Dialogue Policy Challenges

- How can we incorporate new information and actions into our model most efficiently?

What is a train destination?

What does train book do?

- How can we deal with the **ever-growing number of information** as domains are added?

I can not comprehend hundreds of domains at the same time

# Dialogue Policy Challenges



- How can we incorporate new information and actions into our model most efficiently?

What is a train destination?

What does train book do?

- How can we deal with the ever-growing number of information as domains are added?

I can not comprehend hundreds of domains at the same time

- How can we talk about **new domains** in a zero-shot fashion?

I need a train on sunday

I only learned to talk about hotels

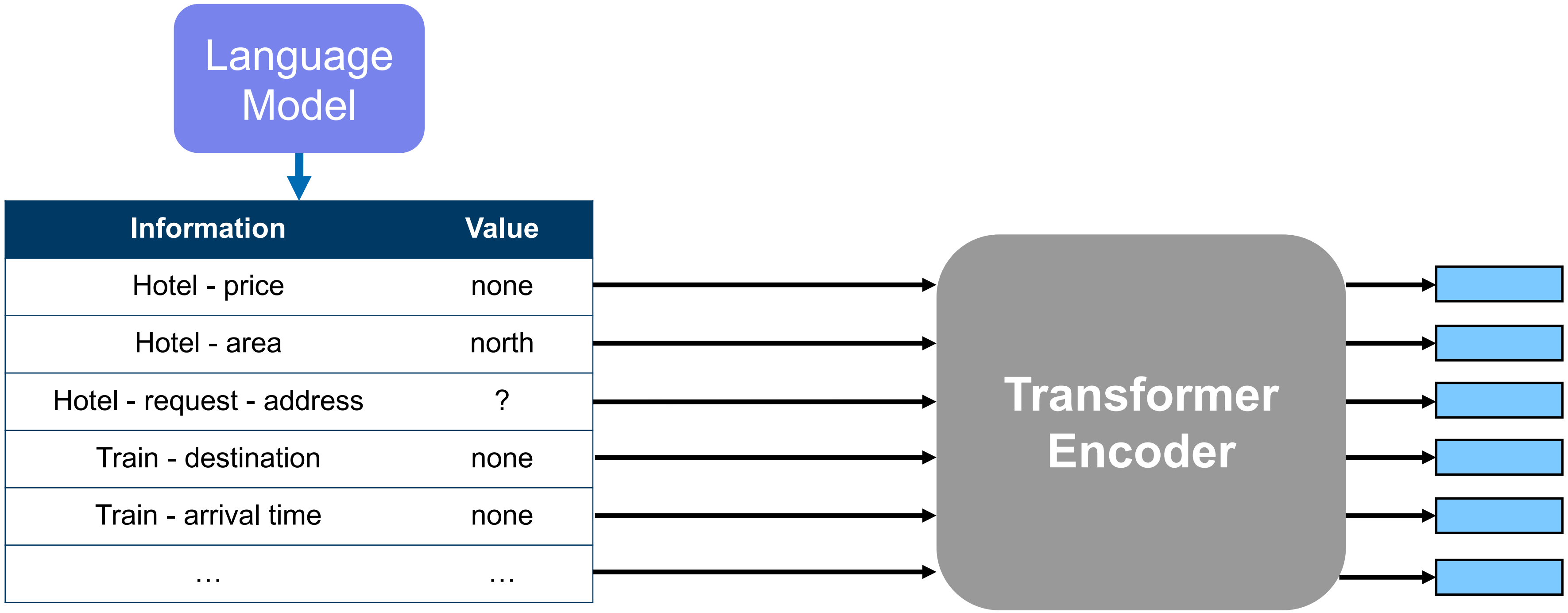
# Continual learning for dialogue policies

How can we enable continual learning for dialogue policies?

# Incorporate new information and actions

Heinrich Heine
Universität Düsseldorf

- Observation: every information and action is describable in natural language



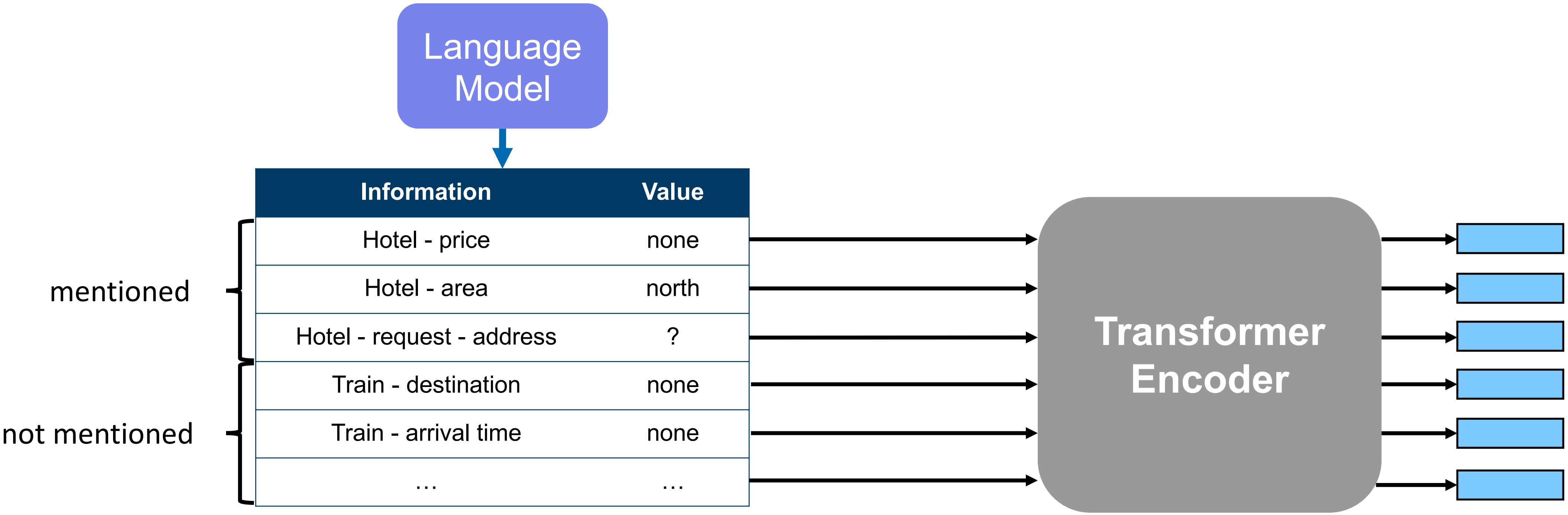

| Information | Value |
|---|---|
| Hotel - price | none |
| Hotel - area | north |
| Hotel - request - address | ? |
| Train - destination | none |
| Train - arrival time | none |
| … | … |

- Language model allows zero-shot understanding of new information

# Dealing with large amount of information

Heinrich Heine
Universität Düsseldorf

- Bound the information to process

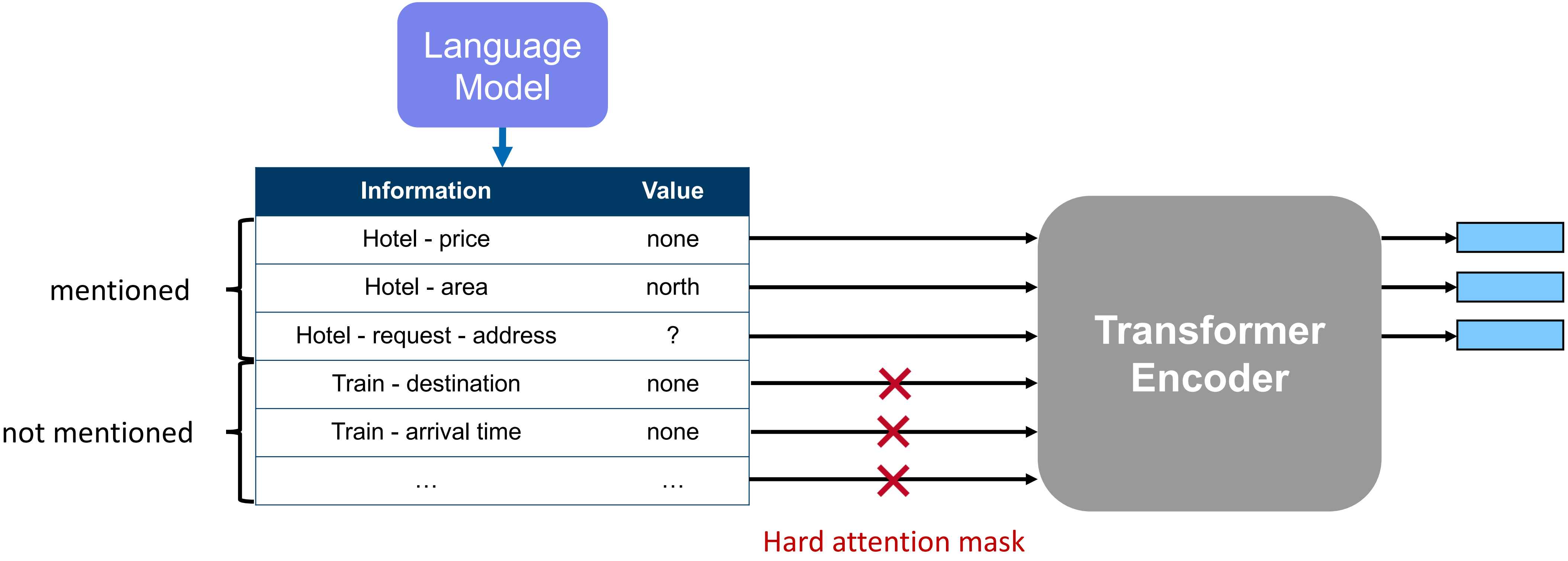



| Information | Value |
|---|---|
| Hotel - price | none |
| Hotel - area | north |
| Hotel - request - address | ? |
| Train - destination | none |
| Train - arrival time | none |
| … | … |

# Dealing with large amount of information



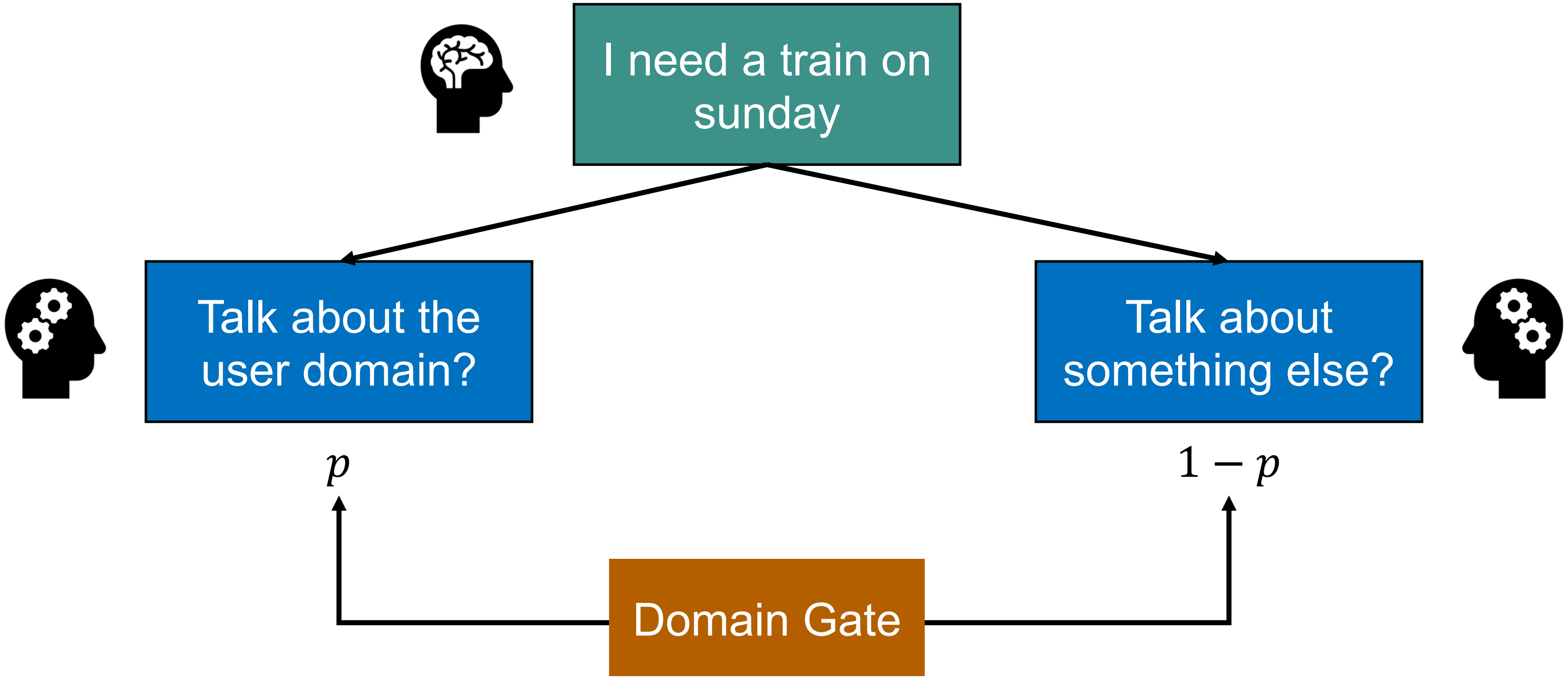
- Hard attention mask for bounding information (inspired by human focus)




| Information | Value |
|---|---|
| Hotel - price | none |
| Hotel - area | north |
| Hotel - request - address | ? |
| Train - destination | none |
| Train - arrival time | none |
| … | … |

# Talk about an unseen domain

- How can we talk about **new domains** in a zero-shot fashion?
- Idea: abstract the question about the domain to choose first

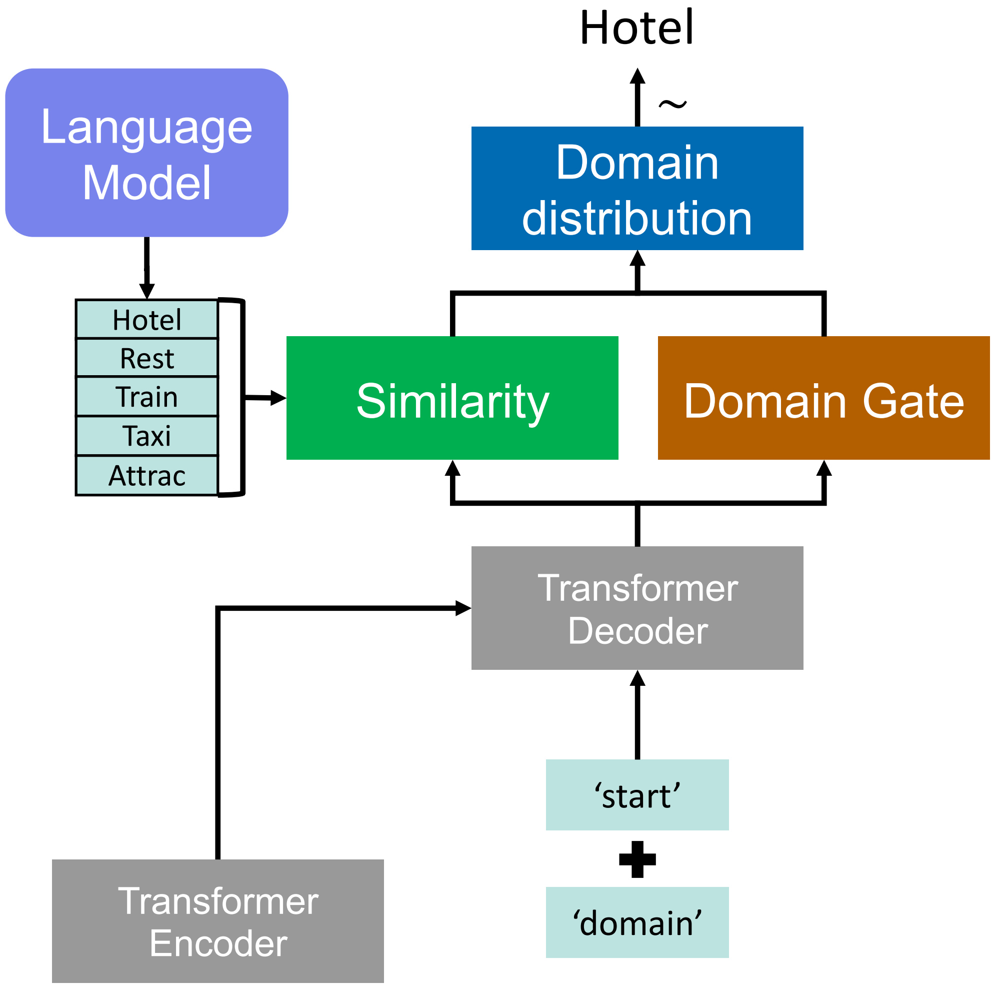


# Action Decoding in DDPT

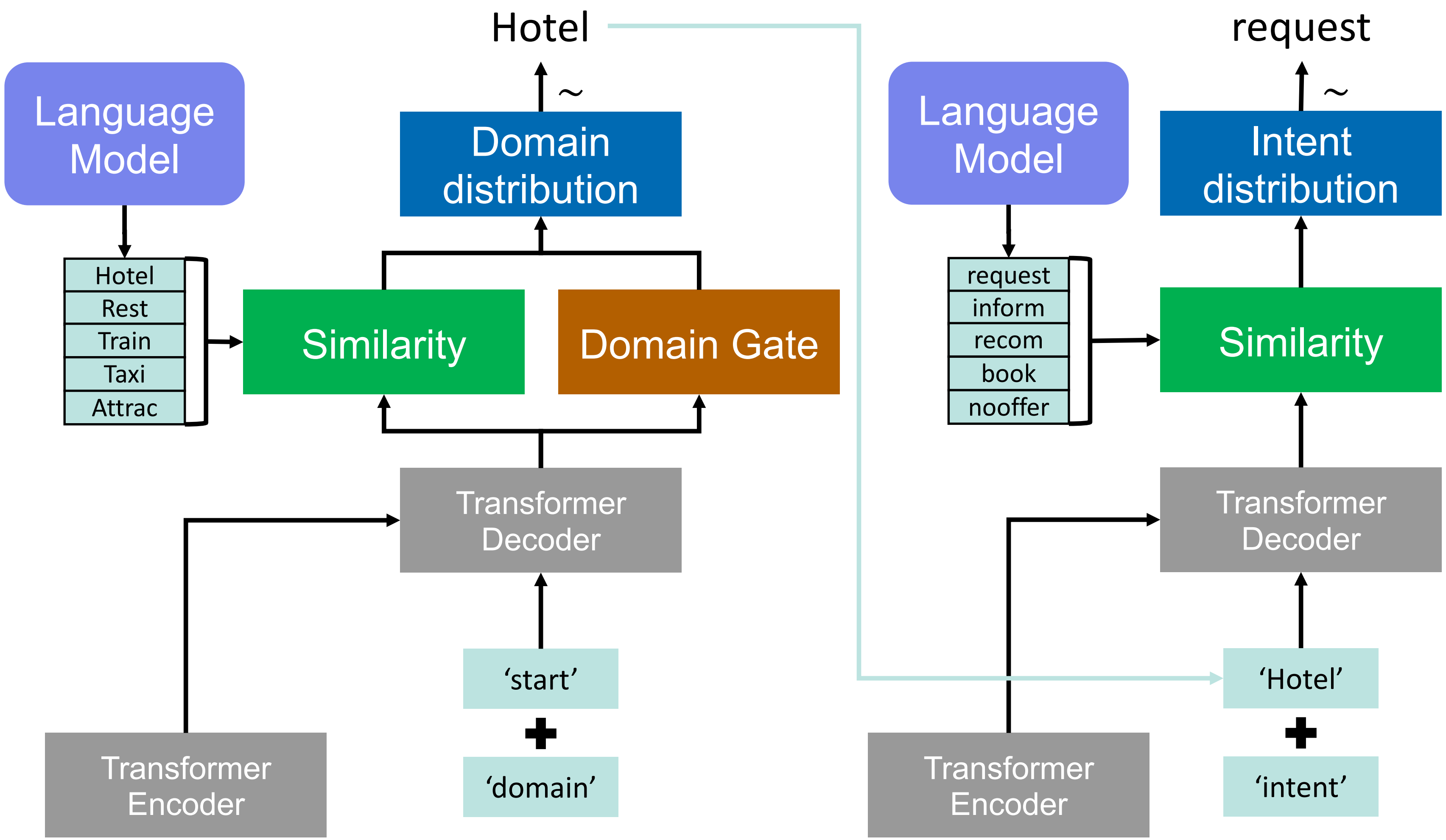
- Autoregressively produces triplets of domain-intent-slot

Hotel -> request -> area -> Hotel -> inform -> address -> …

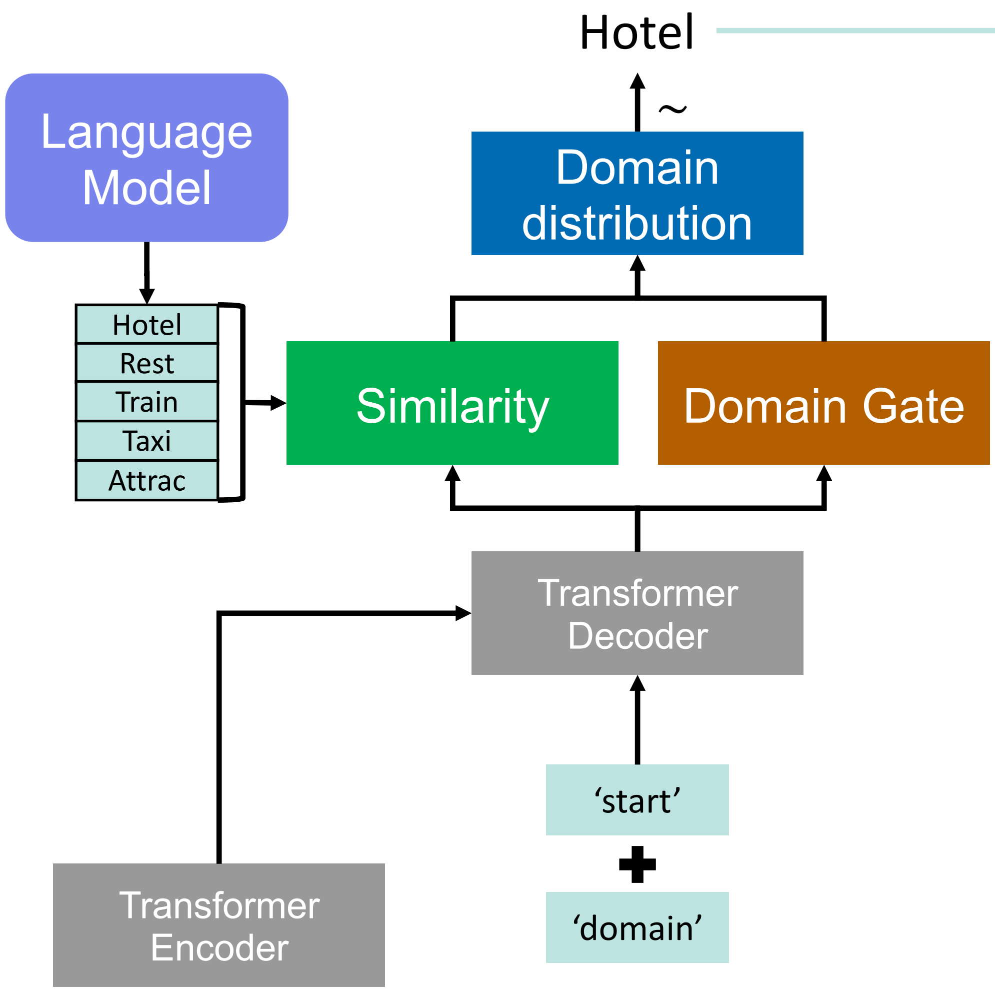
# Action Decoding in DDPT



# Action Decoding in DDPT

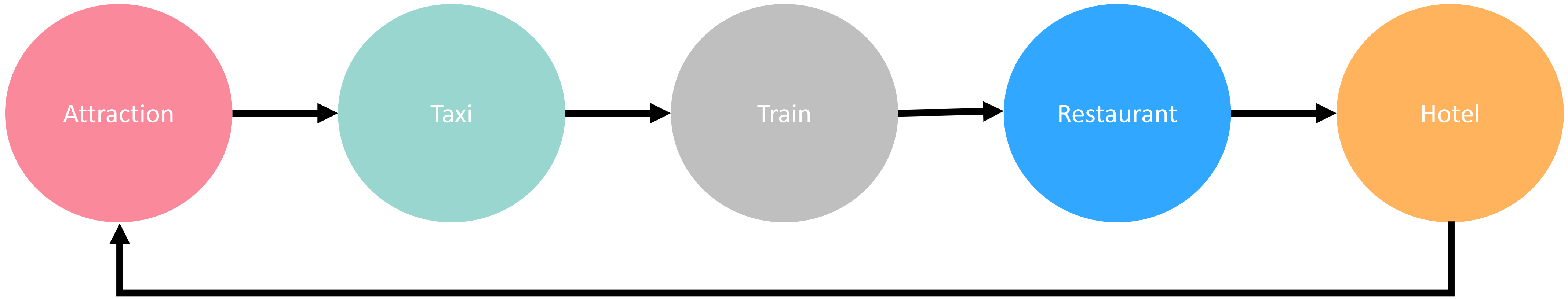


# Action Decoding in DDPT



# Experimental Setup

- Tested on different task sequences (easy2hard, hard2easy, mixed), e.g.




- Using 5 domains from MultiWOZ 2.0 (Budzianowski et al. 2018)

- All models optimized using CLEAR (Rolnick et al. 2018)

# CLEAR algorithm

- We want to prevent catastrophic forgetting and improve fast adaptation
- Fast adaptation: utilize online samples for updates in addition to replay buffer experience
- Prevent forgetting: sample experience from replay buffer
  - KL-divergence loss for regularizing actor policy towards behaviour policy
  - Mean-squared error loss for regularizing critic towards old predictions

$$L_{\text{policy-cloning}} := \sum_a \mu(a|h_s) \log \frac{\mu(a|h_s)}{\pi_\theta(a|h_s)}, \quad L_{\text{value-cloning}} := ||V_\theta(h_s) - V_{\text{replay}}(h_s)||_2^2.$$

- Uses off-policy actor critic algorithm V-trace (Espeholt et al. 2018)

# Baselines

- Bin: binary state representation (Weisz et al. 2018, Zhu et al. 2020)
  - Uses a **binary feature** for every information whether it is present or not

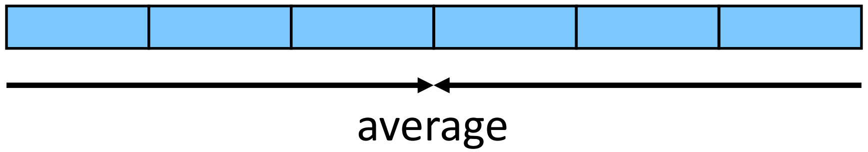
| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | … |

# Baselines

- Bin: binary state representation (Weisz et al. 2018, Zhu et al. 2020)
  - Uses a binary feature for every information whether it is present or not

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | … |

- Sem: semantic state representation (Xu et al. 2020)
  - Uses **trainable embeddings** for domains, intents and slots
  - Uses simple **averaging over domains** to obtain fixed size representation



# Baselines

Heinrich Heine
Universität Düsseldorf

- Bin: binary state representation (Weisz et al. 2018, Zhu et al. 2020)
  - Uses a binary feature for every information whether it is present or not

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | … |

- Sem: semantic state representation (Xu et al. 2020)
  - Uses **trainable embeddings** for domains, intents and slots
  - Uses simple **averaging over domains** to obtain fixed size representation

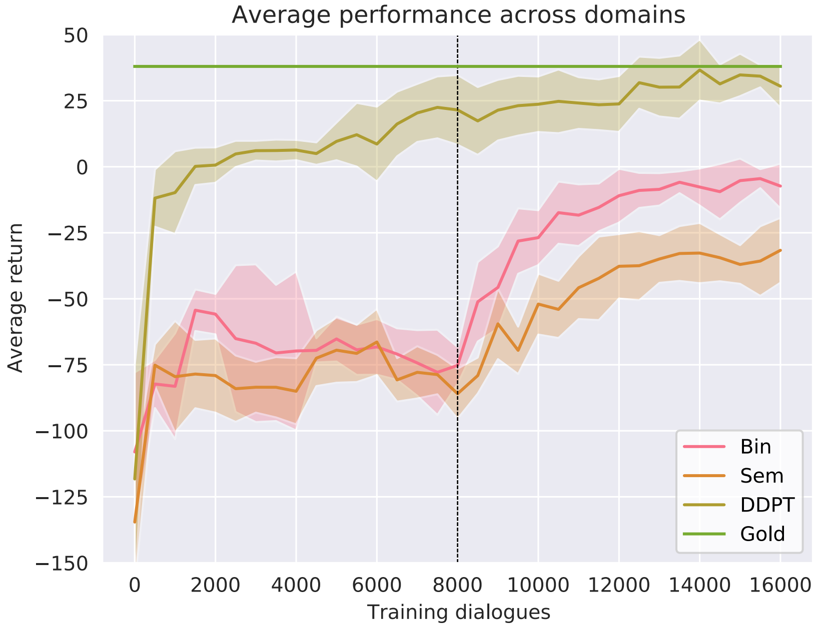



- Gold: serves as **upper bound**
  - obtained through training an (expert) Bin model for each domain until convergence

# Results



- Average performance across domains

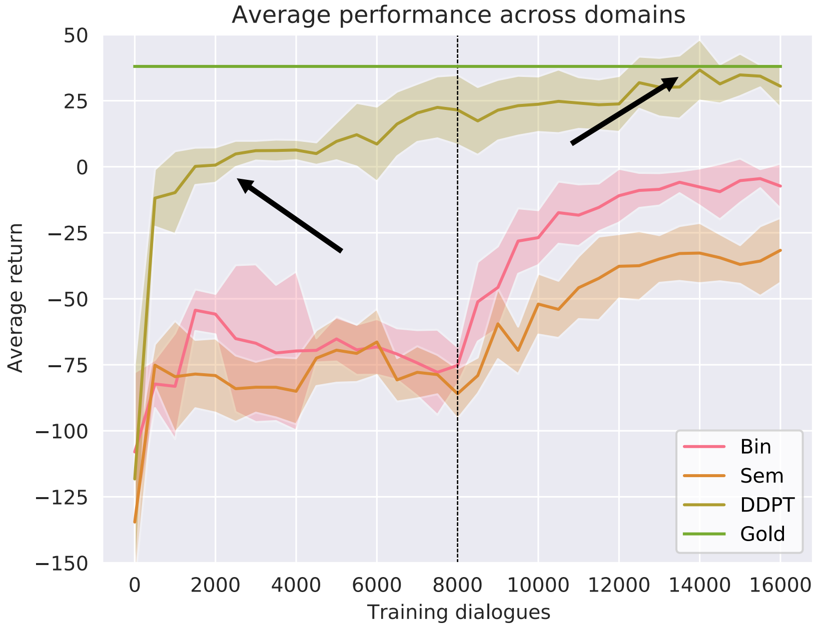



Attraction → Taxi → Train → Restaurant → Hotel

# Results

Heinrich Heine
Universität Düsseldorf

- Average performance across domains



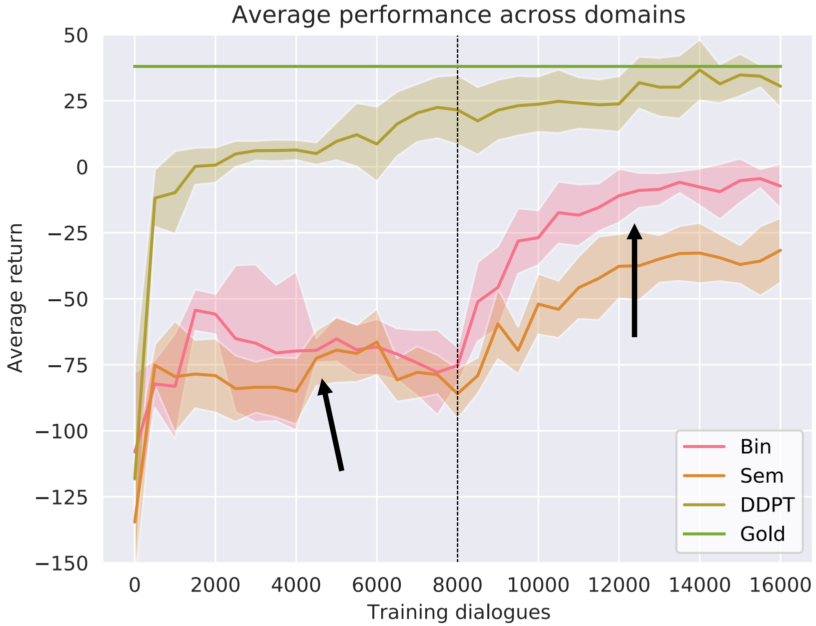

Attraction → Taxi → Train → Restaurant → Hotel

- DDPT quickly accelerates, constantly increasing
  - Indicates strong forward transfer
  - Achieves upper bound performance with fixed size number of parameters!

# Results



- Average performance across domains




Attraction → Taxi → Train → Restaurant → Hotel

- DDPT quickly accelerates, constantly increasing
  - Indicates strong forward transfer
  - achieves upper bound performance with fixed size number of parameters!

- Baselines struggle on first cycle
  - Indicates forgetting and weak forward transfer
  - Second cycle necessary

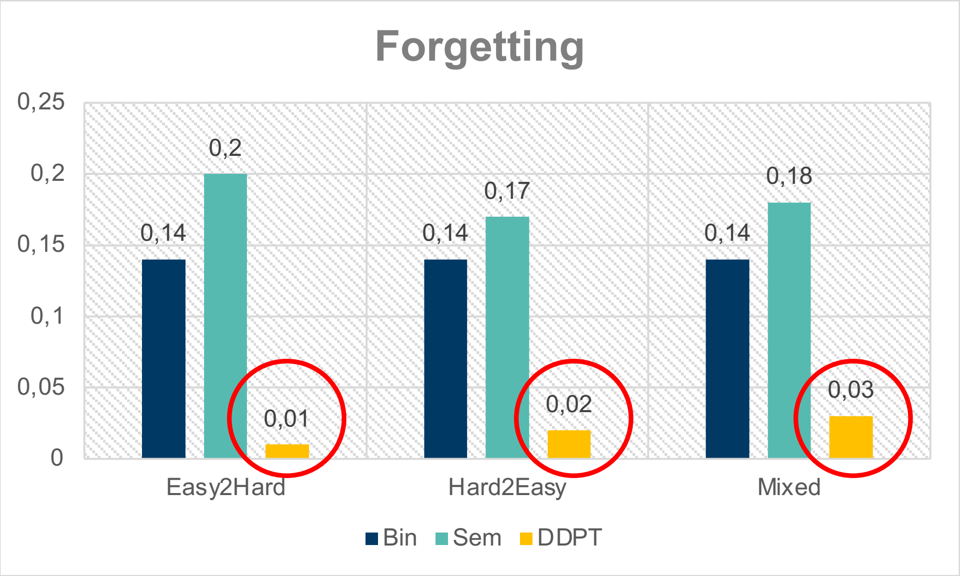
# Forgetting of old tasks

- Forgetting (the lower the better): performance decrease on old tasks after training on new tasks
  - in terms of success rate

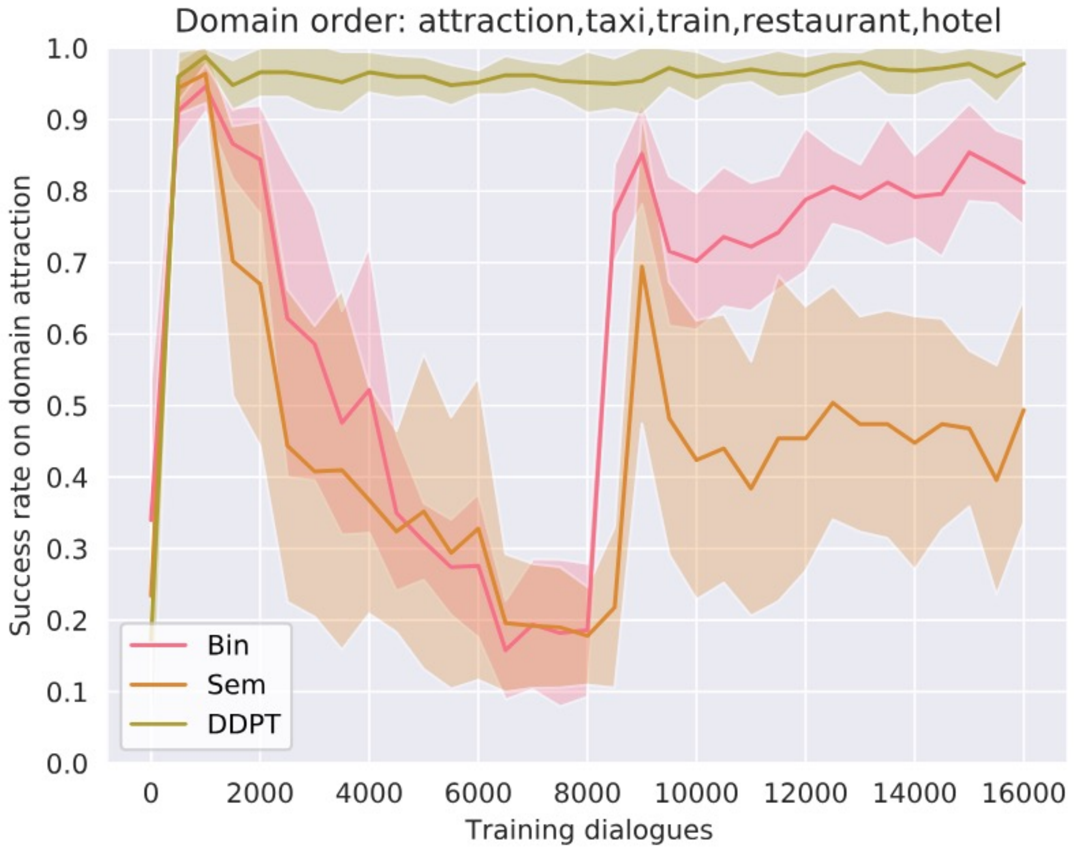
# Forgetting of old tasks

- Forgetting (the lower the better): performance decrease on old tasks after training on new tasks
  - in terms of success rate

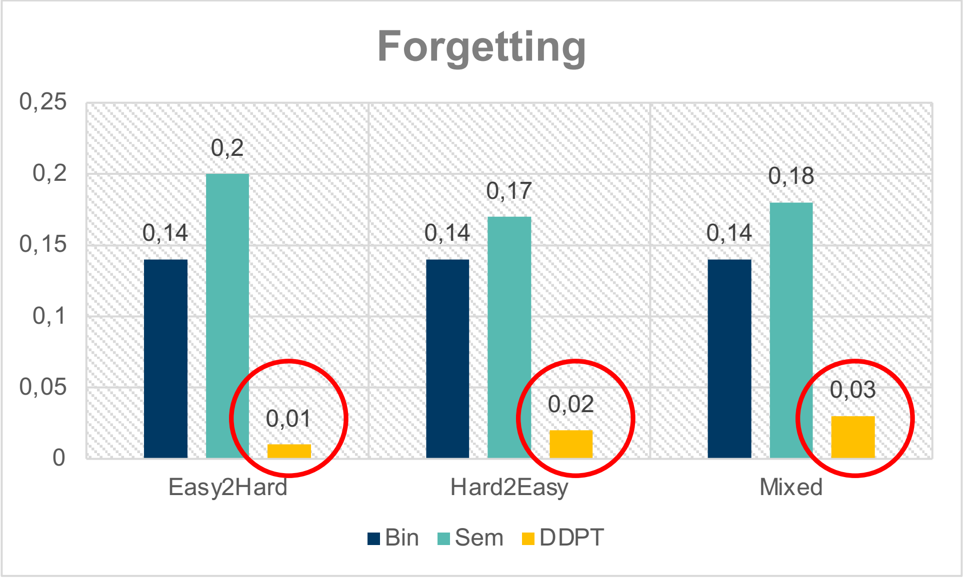
- DDPT is robust against forgetting
  - Frozen language model embeddings are more robust
  - Domain gate mitigates problem of choosing the correct domain



# Forgetting of old tasks




-mance decrease on old tasks after training on new tasks



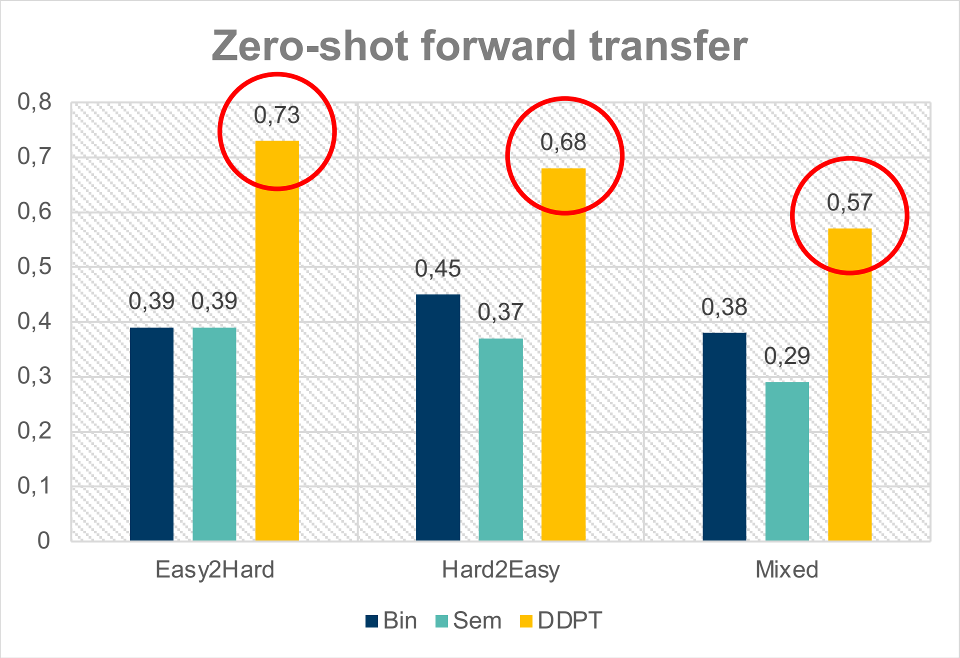
# Zero-shot forward transfer to unseen domains

- Zero-shot forward transfer (the higher the better): performance on unseen tasks
  - in terms of success rate

# Zero-shot forward transfer to unseen domains

- Zero-shot forward transfer (the higher the better): performance on unseen tasks
  - in terms of success rate

- DDPT has significant zero-shot transfer capabilities
  - Description embeddings build relationship between old and new information/actions
  - Domain gate enables talking about a new domain **immediately**



# Dense Rewards and Continual Reinforcement Learning

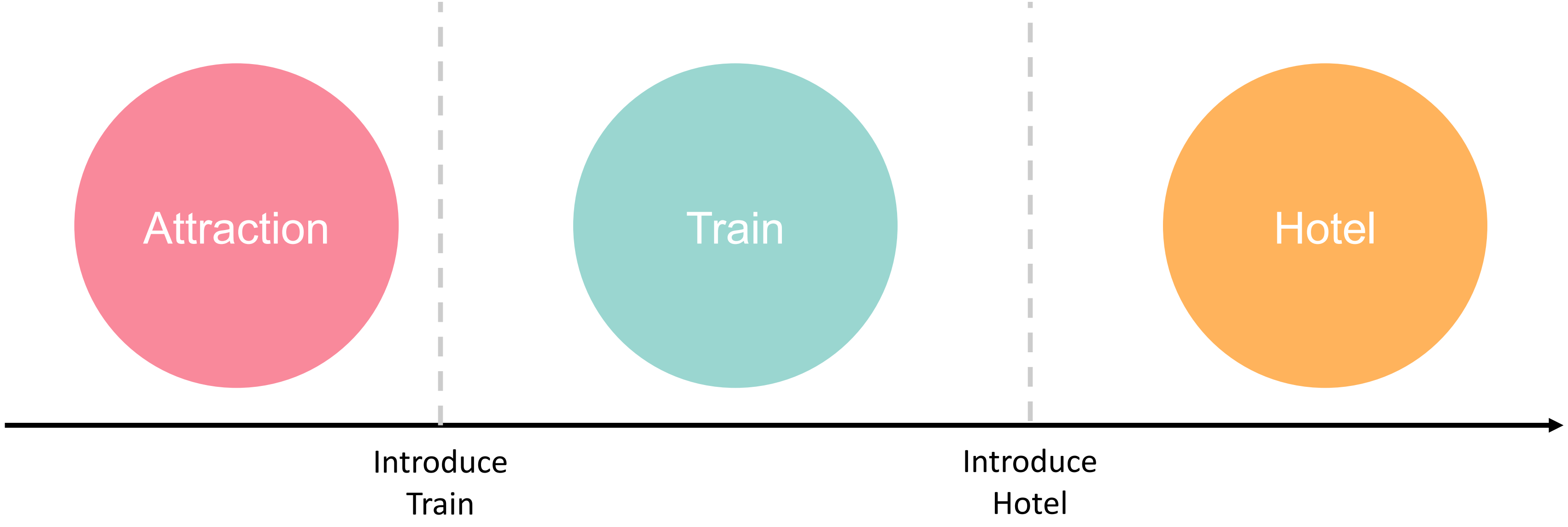


✓ Information gain as dense reward for increased sample efficiency

✓ DDPT architecture enabling continual reinforcement learning of dialogue policies

# Towards Realistic Environments for Continual Learning

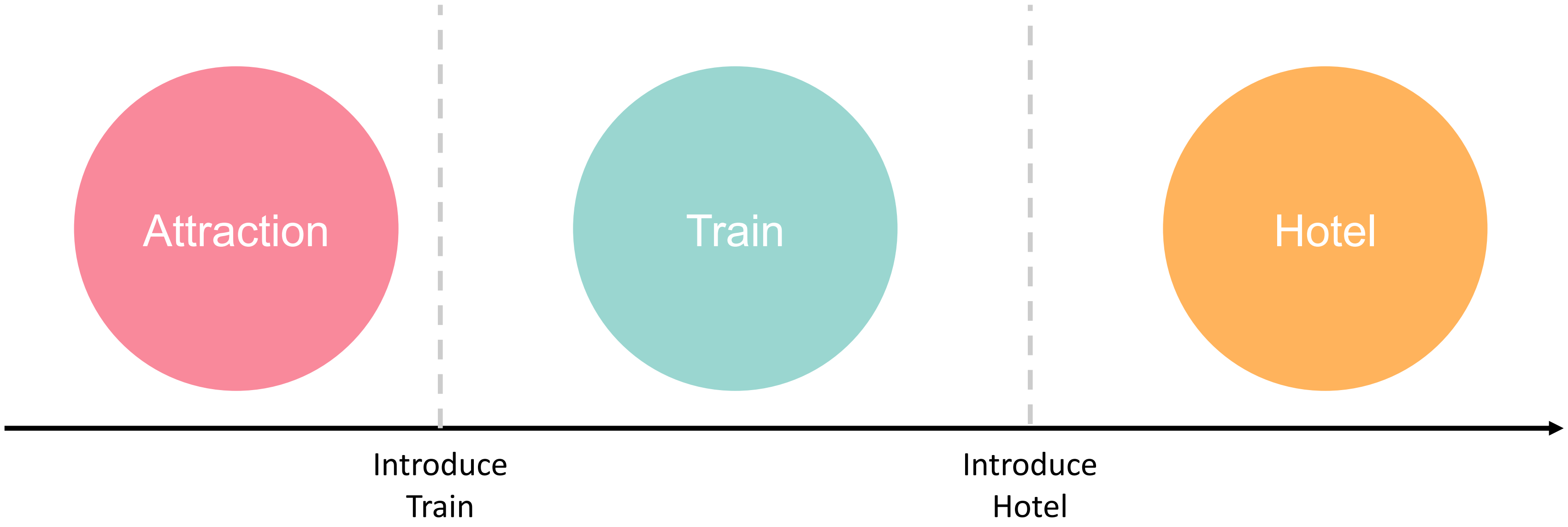
- Observing domains sequentially for a fixed amount of time is an adequate environment for measuring forward transfer and forgetting



# Towards Realistic Environments for Continual Learning



- But how realistic is it?
- It is unlikely that a domain will disappear once another domain is introduced
- Why should we evaluate the system on a domain it will never see again?



# Towards Realistic Environments for Continual Learning

Heinrich Heine Universität Düsseldorf

- Continually learning dialogue systems face various circumstances
  - Multiple domains can emerge within a dialogue (taxi + restaurant within a dialogue)
  - System experiences a multitude of user behaviours
  - User demands can change over time (e.g. due to seasonal changes)
  - Currently none of these circumstances is included in continual learning setups

- How should we evaluate continually learning dialogue policies?

# Realistic environments

How can we build realistic environments for continual learning?

# Realistic Environments for Continual Learning



- We propose a more realistic, flexible and controllable framework for continual reinforcement learning of dialogues (called RECORD)

# Realistic Environments for Continual Learning

- We propose a more realistic, flexible and controllable framework for continual reinforcement learning of dialogues (called RECORD)

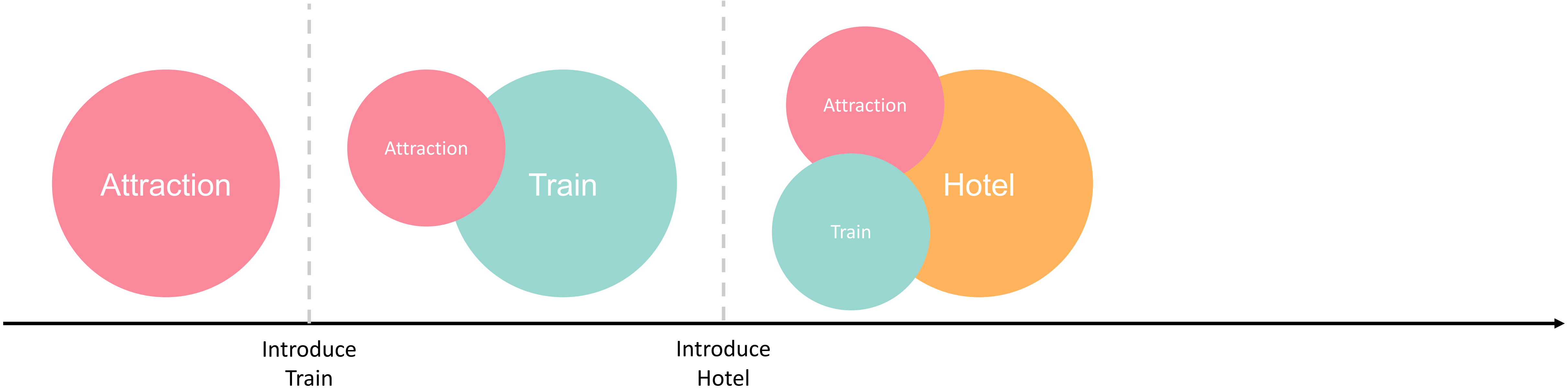
# Realistic Environments for Continual Learning

- We propose a more realistic, flexible and controllable framework for continual reinforcement learning of dialogues (called RECORD)


✓Domains occur again ✓Multi-domain dialogues

# Realistic Environments for Continual Learning

- We propose a more realistic, flexible and controllable framework for continual reinforcement learning of dialogues (called RECORD)

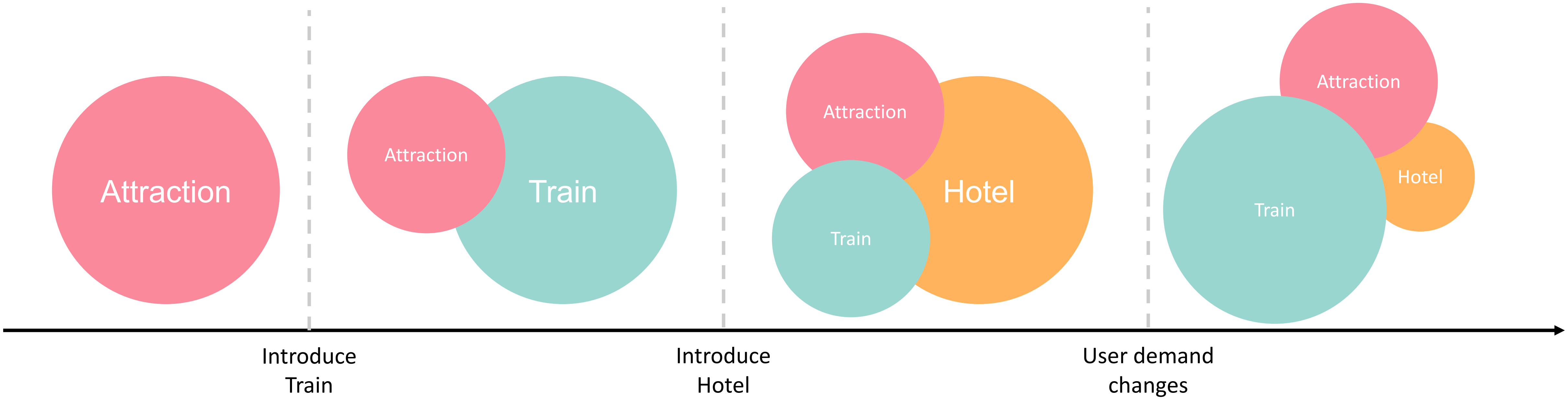



✓Domains occur again ✓Multi-domain dialogues

# Realistic Environments for Continual Learning



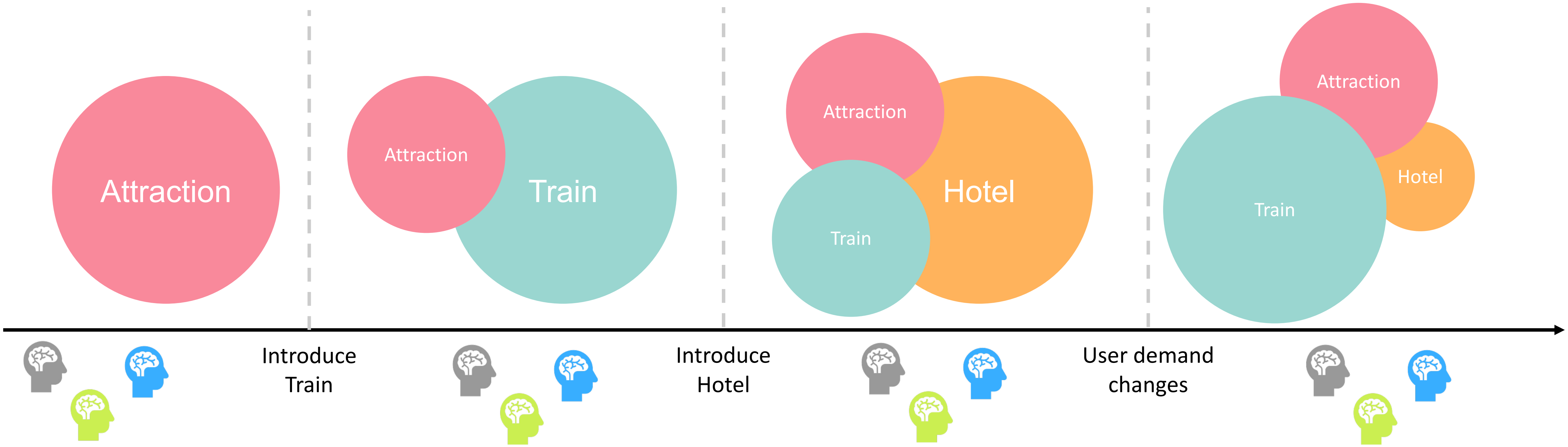
- We propose a more realistic, flexible and controllable framework for continual reinforcement learning of dialogues (called RECORD)




✓Domains occur again ✓Multi-domain dialogues ✓User demands changes

# Realistic Environments for Continual Learning

- We propose a more realistic, flexible and controllable framework for continual reinforcement learning of dialogues (called RECORD)




✓Domains occur again ✓Multi-domain dialogues ✓User demands changes ✓Multiple users

# Realistic Environments for Continual Learning

Heinrich Heine
Universität Düsseldorf

- We propose a more realistic, flexible and controllable framework for continual reinforcement learning of dialogues (called RECORD)
- RECORD generalizes the previous setup!

# How to evaluate the performance?

How should we evaluate continually learning dialogue policies?

# How to evaluate the performance?

- Continually learning agents should not be evaluated on how well they perform on tasks/domains that never occur again
  - Humans also forget over time if the knowledge is not required

- Agents should be evaluated on how well they perform during their lifetime
  - Be as good as possible on the circumstances that are actually observed during learning

→ We evaluate agents on their lifetime performance

# Optimization objectives

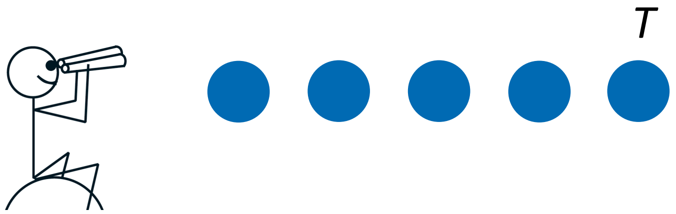
How should we optimize continually learning agents?

# Episodic and Lifetime Optimization

hhu
Heinrich Heine
Universität Düsseldorf

- Models are typically optimized for episodic return (per dialogue return)

$$G^{epi} = \sum_{t=0}^{T} \gamma^t \cdot r_t$$



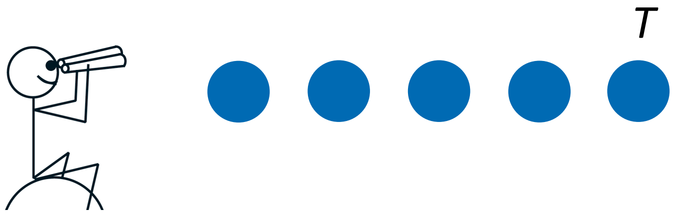

- This optimizes for the “present”, i.e. the current circumstances

- Does not take into account changing circumstances in the future that inevitably occur in continual learning

# Episodic and Lifetime Optimization

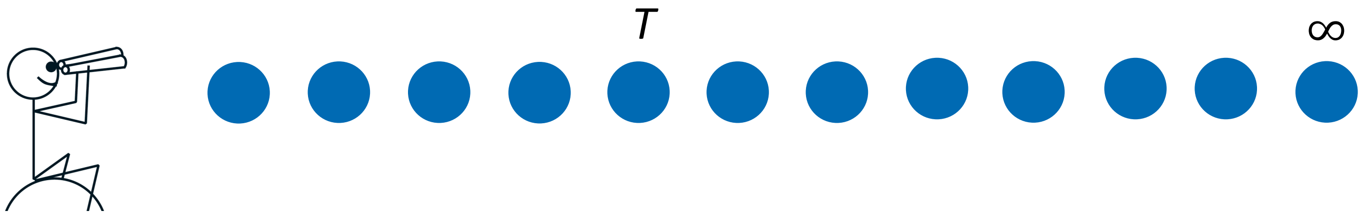
- Models are typically optimized for episodic return (per dialogue return)

$$G^{epi} = \sum_{t=0}^{T} \gamma^t \cdot r_t$$




- We propose to include **lifetime return** into the optimization to take changes into account

$$G^{life} = \sum_{t=0}^{\infty} \gamma^t \cdot r_t$$



# Meta-Learning Hyperparameters

hhu
Heinrich Heine
Universität Düsseldorf

- RL algorithms often have additional loss terms that affect learning

CLEAR

Hyperparameters

$\alpha \cdot \mathcal{L}_{entropy}$

Maximize entropy

$\beta \cdot \mathcal{L}_{regularize}$

Regularize towards past behaviour

- Optimal hyperparameters can vary throughout lifetime of the dialogue system

- Proposal: meta-learn hyperparameters towards maximization of lifetime performance

# Recap on Meta Learning



- Meta-learning consists of two models (potentially the same): base model and meta model

# Recap on Meta Learning



- Meta-learning consists of two models (potentially the same): base model and meta model
- Meta-learning consists of two phases: inner loop updates and outer loop updates

# Recap on Meta Learning

- Meta-learning consists of two models (potentially the same): base model and meta model

- Meta-learning consists of two phases: inner loop updates and outer loop updates
  - Inner loop update: update the base model $\theta_i$ using meta model predictions ($M$ times)

# Recap on Meta Learning

- Meta-learning consists of two models (potentially the same): base model and meta model

- Meta-learning consists of two phases: inner loop updates and outer loop updates
  - Inner loop update: update the base model $\theta_i$ using meta model predictions ($M$ times)
  - Outer loop update: update the meta model $\eta$ based on the updated parameters of the base model

# Recap on Meta Learning

- Meta-learning consists of two models (potentially the same): base model and meta model

- Meta-learning consists of two phases: inner loop updates and outer loop updates
  - Inner loop update: update the base model $\theta_i$ using meta model predictions ($M$ times)
  - Outer loop update: update the meta model $\eta$ based on the updated parameters of the base model
    - The updated parameters $\theta_{i+M}$ of the base model depend on the meta model parameters $\eta$

$$\theta_i \xrightarrow{\eta} \theta_{i+1} \xrightarrow{\eta} \ldots \xrightarrow{\eta} \theta_{i+M-1} \xrightarrow{\eta} \theta_{i+M}$$

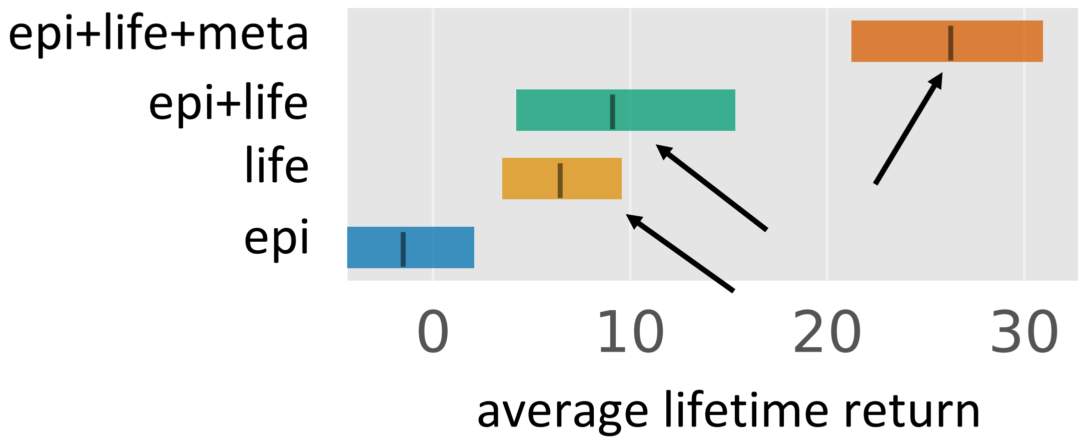
Depends on $\eta$

# Experimental Setup

- We test 4 different settings
  - epi: optimize for episodic return
  - life: optimize for lifetime return
  - epi+life: optimize for episodic and lifetime return
  - epi+life+meta: optimize for episodic and lifetime return and meta-learn hyperparameters
- We use CLEAR as base algorithm

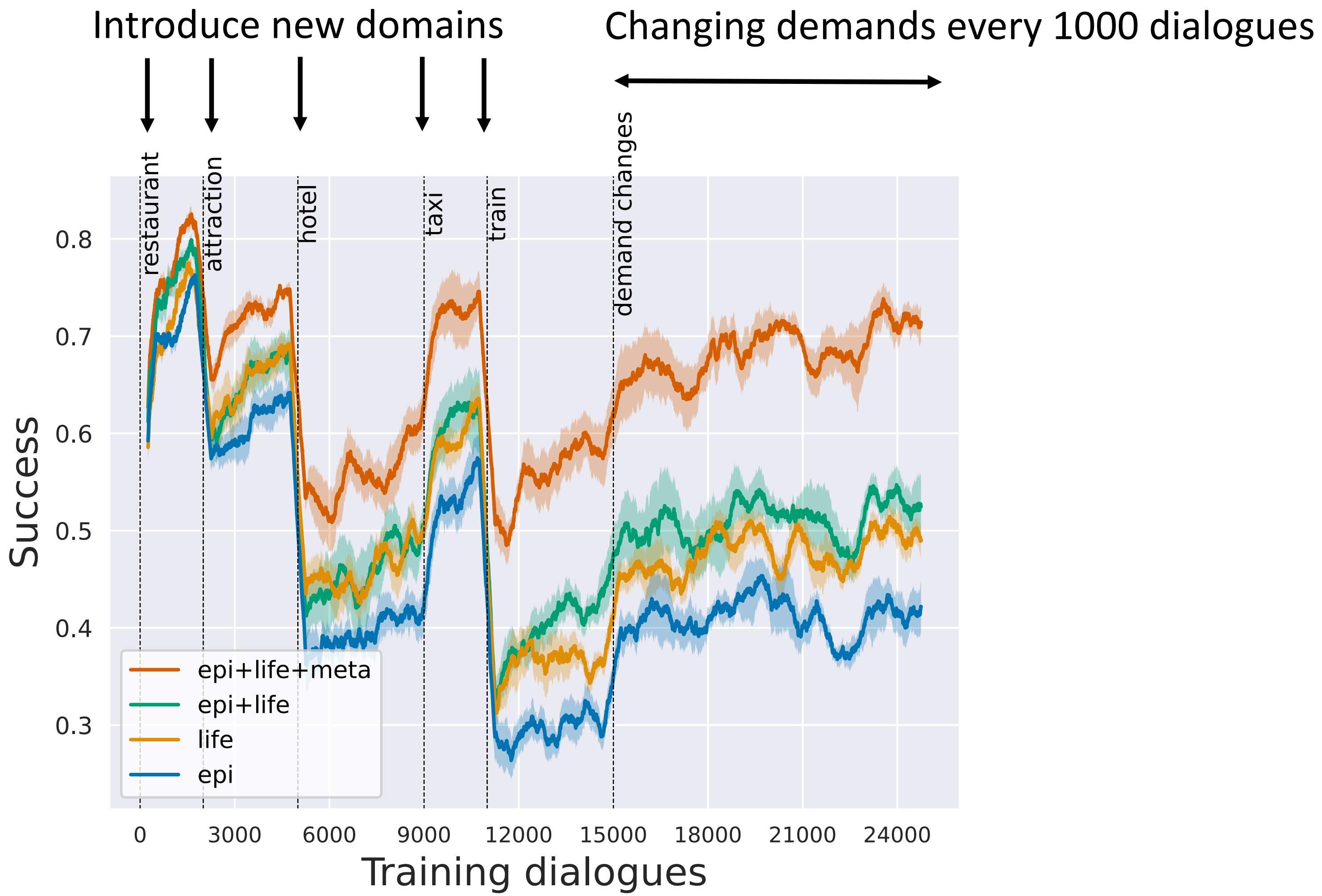
# Results

- Average lifetime return for different optimization objectives



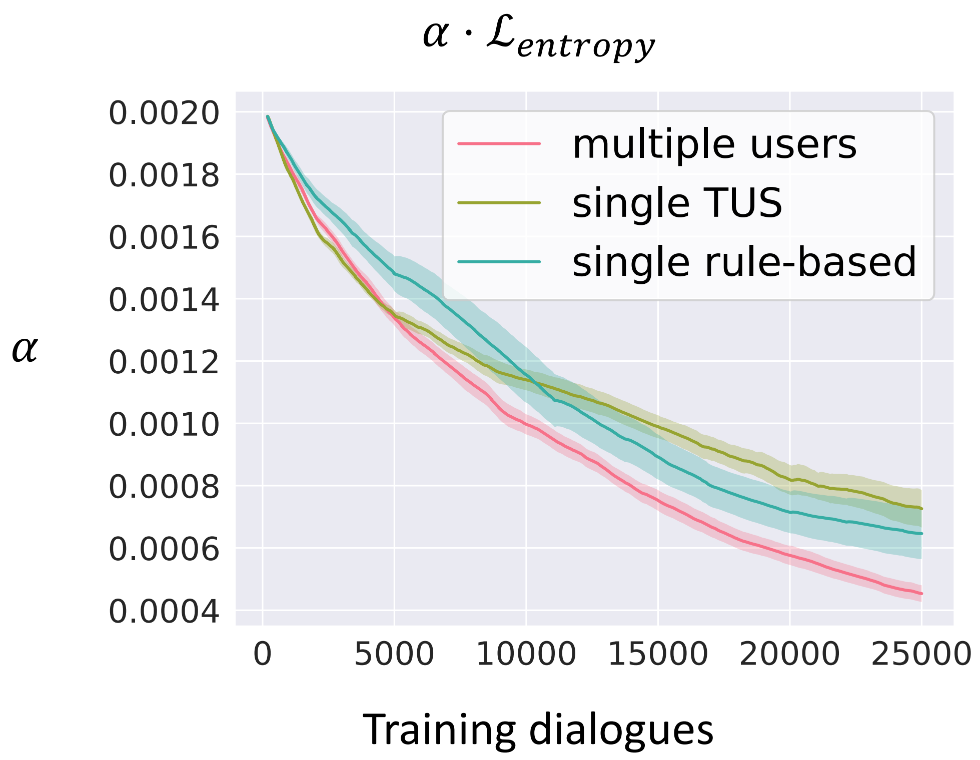
# Results

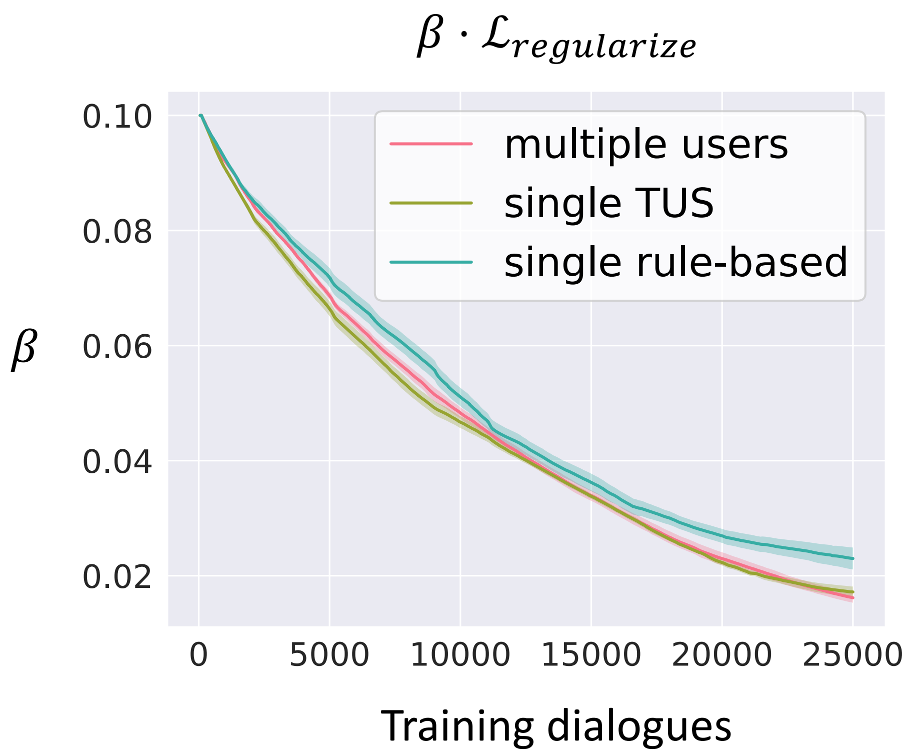
- Average success rate within a time-window of 500 dialogues
- Performance drops when new domain is introduced
- All models adapt over time after domain introduction



# Results

- Loss weights decrease over time as more experience is collected



# Dense Rewards and Continual Reinforcement Learning

Heinrich Heine Universität Düsseldorf

✓ Information gain as dense reward for increased sample efficiency

✓ DDPT architecture enabling continual reinforcement learning of dialogue policies

✓ RECORD framework for realistic environments

✓ Lifetime return optimization and meta-learning for enhanced continual learning

# Conclusions and future works

# Future Directions

Heinrich Heine Universität Düsseldorf

- Improve adaptation of continually learning agents
  - Episodic memory
  - Exploration

- Reward learning
  - Learn intrinsic reward functions during the lifetime of the agent that adapts to circumstances

- Large language model (LLM) integration
  - Utilize generalization capabilities of LLMs for fast adaptation in continual learning
  - Combine task-oriented dialogue system modules with LLMs

# Thank you for listening