Wei Wang

(+1) 226-234-0626 | wwang828@uwo.ca | www.waybaba.com

Education

Western University — Ontario, Canada
MSc. of Computer Science — GPA: 4.0/4.0 — *Sep. 2020 – Present (one off-campus year at Microsoft)*

Tianjin University — Tianjin, China
Bachelor of Communication Engineering — GPA: 3.6/4.0 — *Sep. 2016 – June. 2020*

Key Coursework

Reinforcement Learning (98), Theoretical Machine Learning (94), Advanced A.I. (94), Probability Theory & Mathematical Statistics (93), Analysis of Algorithms (94), Computer Organization (90), Digital Signal Processing (98),

Honors

$17,500 Vector Scholarship in Artificial Intelligence, awarded to top AI students in Ontario, Canada (Top 0.1%)
National Student Challenge Cup, National Award, ranked 2nd province-wide (Top 1%)
National Undergraduate Electronic Design Contest of China — Provincial Prize (Top 1%)

Publications († First Author, †† Second Author)

† [1] **Addressing Signal Delay in Deep Reinforcement Learning (ICLR 2024 Spotlight Top 5.01%)** 2024

- Addressed environment generalization in RL, especially focusing on examining the effect of signal delays in RL. Discovered that a merely 2-4 timestep delay can significantly reduce performance below 20% in SOTA RL algorithms.
- Provided a theoretical analysis and developed modules that restore Markovian properties, boosting performance back to around 90%. The proposed modules can be plugged into actor-critic frameworks to mitigate singal delay.

† [2] **Toward Open-ended Tasks Solving in Reinforcement Learning (NeurIPS 2023 Workshop)** 2023

- Previous goal-conditioned RL can only finish the $g' \in \mathcal{G}$ (e.g. navigate to location $g' \in \mathcal{G}$) where the task set $\mathcal{G}$ need to be specially pretrained and $\mathcal{G}$ is only a subset of all types of goals $\mathcal{A}$ (e.g. navigation, fighting, moving faster). We extend it to a more general cases where $g \in \mathcal{A}$ and require no pre-training.
- We use classifier-guided diffusion model which can generate plans to finish goal g as $\tau_g \sim p_\theta(\tau|g)$ with only trained on $\tau' \sim p_\theta(\tau)$. Our framework offer a possible solution to convert almost goal in language space $\mathcal{L}$ to goal space $\mathcal{G}$ finish general types of goal g, lead to a possible solution to open-ended AI.

† [3] **Dynamically Instance-Guided Backward-Free Test-Time Domain Adaptation (CVPR 2023)** 2023

- Setting: Traditional domain adaptation requires collecting target data before deploying, which is unpractical when target data is expensive (e.g. medical images) and even unavailable (e.g. real-time robot, auto-driving). To this end, we aim at directly deploying source model and adapt during test-time in an online fashion.
- Method: We proposed two modules, distribution adaptation module (DAM) mixes the instance and source BN statistics to encourage the model to capture robust representation. Semantic adaptation module (SAM) combines the historical prototypes with instance-level prototypes to adjust semantic predictions.
- Strength: 1)Efficiency. Both modules avoid back-propagation thus are highly efficient, which is crucial for Test-Time adatatation. 2) Usability. Our method is model-agnostic, which can be readily injected into existing models. 3) Anti-forgetting, model parameters are maintained during adaptation so that forgetting is avoided. 4) Performance: High mIoU is achieved over 15 adaptation setting.

†† [4] **Generalizing Across Temporal Domains with Koopman Operators (AAAI 2024)** 2023

- Setting: To achieve high performance on unseen out-of-distribution target $\mathcal{D}_{n+1}$ with only source domains sequence $\{\mathcal{D}_i\}_{i=1}^n$ (e.g. images under variant light conditions), existing DG focus on training one generalized model. Instead, we propose to exploit the evolving pattern in $\{\mathcal{D}_i\}_{i=1}^n$ and produce a model specifically for $\mathcal{D}_{n+1}$.
- Methods: Base on prototypical framework, we proposed a trainable domain transforming network which could mapping the support set of $\mathcal{D}_i$ to simulate the centroids of classes in the $\mathcal{D}_{i+1}$, thus a model for $\mathcal{D}_{n+1}$ can be got without accessing $\mathcal{D}_{n+1}$.
- Theoretical bound: Our theoretical result reveals the benefits of modeling the relation between two consecutive domains by learning a globally consistent directional mapping function.

†† [5] **Data Augmentation for Domain Generalizationin in Non-Stationary Environment (AAAI 2023)** 2022

- Setting: To achieve high performance on unseen out-of-distribution target $\mathcal{D}_{n+1}$ with only source domains sequence $\{\mathcal{D}_i\}_{i=1}^n$ (e.g. images under variant light conditions), existing DG focus on training one generalized model. Instead, we propose to exploit the evolving pattern in $\{\mathcal{D}_i\}_{i=1}^n$ and produce a model specifically for $\mathcal{D}_{n+1}$.
- Methods: We propose to generating middle samples based on the between $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$.

†† [6] **Consecutive Learning from Accumulated Knowledge (ACM TKDD)** 2021

- Setting: Under the multi-task learning setting, we focus on scenarios where both tasks and samples comes sequentially (e.g. Google provides searching and video service, in both of which user data comes sequentially).
- Method: For each task, a new expert is introduced and we dynamically combining the the prediction of both current expert and previous expert in the knowledge database. A model selection strategy is introduced in further to reduce the space cost of historical experts.
- Results: We finally achieve a 20.1% performance degradation for a 78.9% model space compression

† [7] **TesTransor: Test-time Transformer Adaptor (CVPR 2024 Under Review)** 2023

- Investigated how transformer-based models adapt to new environments (e.g. varying weather conditions and urban settings) during test-time.
- Expirically Demonstrated that applying existing adaptation methods on transformer is not effective. We proposed an novel token tuning techniques for transformer to adapt, which is plug-and-play, lightweight and non-forgetting.

Industrial Research Experience

Microsoft Research Asian Internship (Off-campus) Nov. 2022 - Dec. 2023

Microsoft Inc.

- Project 1: Study the dynamic of environment, focusing on delayed signal problem in RL, resulting in a paper accepted by ICLR 2024 as spotlight [1].
- Project 2: Focused on employing diffusion models to realize embodied AI and address open-ended goals in RL, leading to a paper presentation at the NeurIPS 2023 Workshop [2].

Bytedance/TikTok AI-Lab Research Scientist Internship Sep. 2020 - Jan. 2021

Bytedance/TikTok Inc.

- Engaged in applying reinforcement learning within a 3D MOBA game environment with human player. Key tasks included pathfinding and combat strategy development in a multi-agent setting.
- Managed the implementation of algorithms such as IMPALA and PPO for pathfinding and pre-combat training, contributing significantly to integrated solution development.
- Contributed to asynchronous versions of mainstream RL algorithms within companies' self-hosted frameworks.

Horizon Robotics AI Research Scientist Internship May 2019 - Nov. 2019

Horizon Robotics Inc.

- Specialized in action detection using pre-processed skeleton position data from videos for public surveillance (e.g., detecting social fights) and intelligent elderly care (e.g., identifying falls).
- Engaged in spatio-temporal information extraction using graph neural networks, formulated key frame selection strategies based on RL, and investigated multi-model multi-task fusion approaches.
- Delivered high detection accuracy in real-world scenarios, achieving a precision of 99.8% and a recall of 97.7%.

Technical Skills

Development: Proficient in Python, Pytorch, Docker, and Slurm; experienced in large model training with 400+ GPUs.
Mathematical: Convex Optimization, Functional Analysis, Real Analysis, Measure Theory.

Teaching

CS2214 - Discrete Structures for Computing; CS2210 - Data Structures and Algorithms; CS2211 - Software Tools and Systems Programming; CS9542/CS4442 - Artificial Intelligence II; CS1026 - Computer Science Fundamentals; CS3340 - Analysis of Algorithms; CS3121/CS9146 - Artificial Intelligence I; CS2209 - Applied Logic for Computer Science.