

---

# Torchmm

*Release 0.0.1*

Sep 02, 2020



<b>1</b>	<b>Features</b>	<b>1</b>
<b>2</b>	<b>Contents</b>	<b>3</b>
2.1	About . . . . .	3
2.2	Tutorials . . . . .	3
2.3	Agents . . . . .	3
2.4	Common . . . . .	5
2.5	Environments . . . . .	6
2.6	Utilities . . . . .	7
2.7	Trainers . . . . .	7
2.8	Core . . . . .	7



# CHAPTER 1

---

## Features

---

- Fast and easy to use
- Ready to use out-of-the box implementations of popular algorithms and models
- Pretrained models



## 2.1 About

### 2.1.1 Introduction

Natural Language Processing has created a new paradigm shift in both academia and industry today. With the coming up of Transformers, the field has seen great growth and applications to fields which were previously unheard of. Google's BERT has further accelerated the progress in the field. This paper can be considered as the backbone of modern NLP. Multi-Modal systems utilize data from 2 or more modalities to make decisions. Our goal with this library is to provide an efficient, fast and reliable way to load and train models on multi-modal data.

With Tochmm, our goal is three-fold: - To educate the user about Natural Language Processing and Multi-Modal Systems.

- Easy to understand implementations of State of the Multi-Modal Algorithms.
- Develop efficient pipelines of existing Algorithms.

## 2.2 Tutorials

## 2.3 Agents

### 2.3.1 A2C

`genrl.agents.deep.a2c.a2c` module

### 2.3.2 DDPG

genrl.agents.deep.ddpg.ddpg module

### 2.3.3 DQN

genrl.agents.deep.dqn.base module

genrl.agents.deep.dqn.categorical module

genrl.agents.deep.dqn.double module

genrl.agents.deep.dqn.dueling module

genrl.agents.deep.dqn.noisy module

genrl.agents.deep.dqn.prioritized module

genrl.agents.deep.dqn.utils module

### 2.3.4 PPO1

genrl.agents.deep.ppo1.ppo1 module

### 2.3.5 VPG

genrl.agents.deep.vpg.vpg module

### 2.3.6 TD3

genrl.agents.deep.td3.td3 module

### 2.3.7 SAC

genrl.agents.deep.sac.sac module

### 2.3.8 Q-Learning

genrl.agents.classical.qlearning.qlearning module

### 2.3.9 SARSA

genrl.agents.classical.sarsa.sarsa module

### 2.3.10 Contextual Bandit

Base

Bootstrap Neural

Fixed

Linear Posterior

Neural Greedy

Neural Linear Posterior

Neural Noise Sampling

Variational

### **2.3.11 Multi-Armed Bandit**

Base

Bayesian Bandit

Bernoulli Bandit

Epsilon Greedy

Gaussian

Gradient

Thompson Sampling

Upper Confidence Bound

## **2.4 Common**

### **2.4.1 Classical Common**

`genrl.classical.common.models`

`genrl.classical.common.trainer`

`genrl.classical.common.values`

### **2.4.2 Bandit Common**

`genrl.bandit.core`

`genrl.bandit.trainer`

`genrl.bandit.agents.cb_agents.common.base_model`

`genrl.bandit.agents.cb_agents.common.bayesian`

`genrl.bandit.agents.cb_agents.common.neural`

`genrl.bandit.agents.cb_agents.common.transition`

## 2.5 Environments

### 2.5.1 Environments

Subpackages

Vectorized Environments

Submodules

`genrl.environments.vec_env.monitor` module

`genrl.environments.vec_env.normalize` module

`genrl.environments.vec_env.utils` module

`genrl.environments.vec_env.vector_envs` module

`genrl.environments.vec_env.wrappers` module

Module contents

Submodules

`genrl.environments.action_wrappers` module

`genrl.environments.atari_preprocessing` module

`genrl.environments.atari_wrappers` module

`genrl.environments.base_wrapper` module

`genrl.environments.frame_stack` module

`genrl.environments.gym_wrapper` module

`genrl.environments.suite` module

`genrl.environments.time_limit` module

Module contents

## **2.6 Utilities**

2.6.1 Logger

2.6.2 Utilities

2.6.3 Models

## **2.7 Trainers**

2.7.1 On-Policy Trainer

2.7.2 Off-Policy Trainer

2.7.3 Classical Trainer

2.7.4 Deep Contextual Bandit Trainer

2.7.5 Multi Armed Bandit Trainer

2.7.6 Base Trainer

## **2.8 Core**

2.8.1 ActorCritic

2.8.2 Base

2.8.3 Buffers

2.8.4 Noise

2.8.5 Policies

2.8.6 RolloutStorage

2.8.7 Values