# Imitation Learning: A Review

Emanuele Ghelfi

February 28, 2023

# Imitation Learning

Assumptions:

- Reward function not easy to formalize
- Access to expert demonstrations

Goal:

- Determine a policy that imitates the expert policy

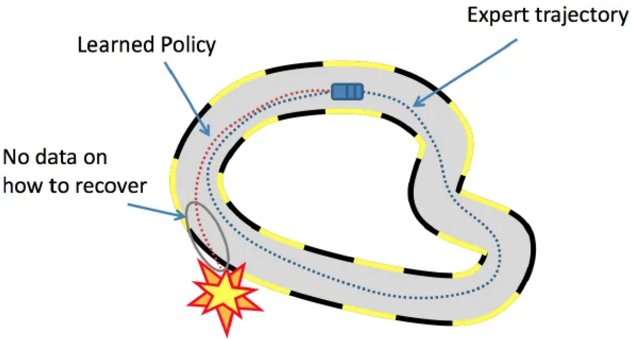
# Imitation Learning

Two main variants:

- Behavior Cloning
- Inverse Reinforcement Learning

# Behavior Cloning

- **Supervised Learning** approach
- Demonstrations not uniformly sampled across the state space
- Compounding error



# Inverse Reinforcement Learning

- What's the reward function "followed" by the expert?
- Recover a reward function from the expert demonstrations
- Train a RL agent using the learned reward function
- Highly inefficient

# Dagger [Ross et al., 2011]

- One of the main BC method
- Simple idea: use dataset aggregation to improve generalization on unseen scenario

**Algorithm 1** Dagger

1: Collect an initial set of trajectories using the expert
2: **for** $i = 1$ to $N$ **do**
3: Train policy with BC
4: Collect data with policy and correct it with the expert
5: Aggregate Dataset
6: **end for**

# Dagger

Cons:

- Frequent interactions with environment and expert using trained policy
- Alleviated by Learning by Cheating [Chen et al., 2020]
  - Train an agent with **privileged** (perfect) information (expert)
  - The privileged agent acts a **teacher** that trains a standard (non-privileged) agent

# GAIL [Ho and Ermon, 2016]

- GAN and maximum entropy IRL
- Main idea: use a discriminator that learns to discriminate real (expert) trajectories ($\pi_E$) from fake (agent) trajectories ($\pi$)
- Combines IRL and BC
- Discriminator $\rightarrow$ Reward
- Generator learns directly the policy during training
- Does **NOT** require expert interaction during training
- Requires **environmental interaction** for closed loop training

# GAIL

Learning objective for the discriminator D:

$$\max_{D} \mathbb{E}_{\pi} [\log(D(s,a))] + \mathbb{E}_{\pi_E} [\log(1 - D(s,a))]$$

# GAIL

Learning objective for the discriminator D:

$$\max_D \mathbb{E}_{\pi} \left[\log(D(s, a)\right] + \mathbb{E}_{\pi_E} \left[\log(1 - D(s, a)\right]$$

Simple idea:

- D(s, a) $\rightarrow$ probability that $(s, a) \sim \pi$ (fake trajectory)

# GAIL

Learning objective for the discriminator D:

$$\max_D \mathbb{E}_{\pi} [\log(D(s, a)] + \mathbb{E}_{\pi_E} [\log(1 - D(s, a)]$$

Simple idea:

- D(s, a) $\rightarrow$ probability that $(s, a) \sim \pi$ (fake trajectory)
- $D(s, a) \rightarrow 1$ when $(s, a) \sim \pi$

# GAIL

Learning objective for the discriminator D:

$$\max_D \mathbb{E}_{\pi}\left[\log(D(s,a))\right] + \mathbb{E}_{\pi_E}\left[\log(1 - D(s,a))\right]$$

Simple idea:

- D(s, a) $\rightarrow$ probability that $(s, a) \sim \pi$ (fake trajectory)
- $D(s, a) \rightarrow 1$ when $(s, a) \sim \pi$
- $D(s, a) \rightarrow 0$ when $(s, a) \sim \pi_E$

# GAIL

Learning objective for the agent is to "confuse" the discriminator:

$$\min_{\pi} \max_{D} \mathbb{E}_{\pi} [\log(D(s,a)] + \mathbb{E}_{\pi_E} [\log(1 - D(s,a)]$$

The agent uses the discriminator as reward (cost) function.
$D(s,a) \to 0$

# GAIL

**Algorithm 2** GAIL

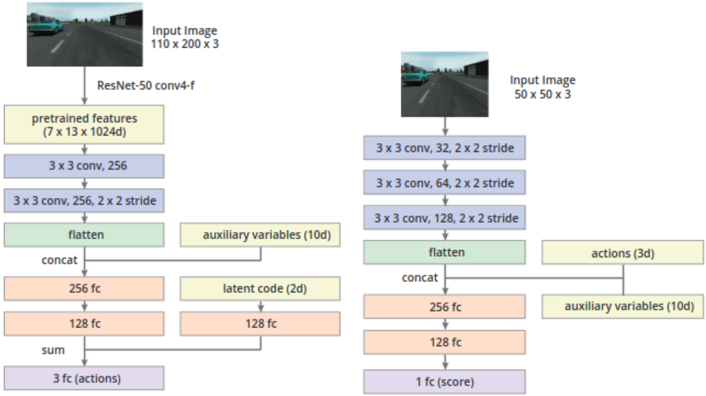
Collect an initial set of trajectories using the expert policy $\pi_E$
**for** $i = 1$ to $N$ **do**
  Sample trajectories with agent policy $\pi_i$
  Update discriminator D with the gradient:

$$\nabla_D = \mathbb{E}_{\pi_i}\left[\nabla \log(D(s,a)\right] + \mathbb{E}_{\pi_E}\left[\nabla \log(1 - D(s,a)\right]$$

  $\pi_i \rightarrow \pi_{i+1}$ using TRPO with cost function $\log(D(s,a))$
**end for**

# InfoGAIL [Li et al., 2017]

GAIL applied to autonomous driving (TORCS) [Video]




(a) Network architecture for the policy/generator $\pi_\theta$. (b) Network architecture for the discriminator $D_\omega$.

# InfoGAIL

- End to End
- Just 3 actions: steering, acceleration, breaking
- Handles multiple experts [Chen et al., 2016]
- Better than BC, GAIL and Human

Table 2: Average rollout distances.

| Method | Avg. rollout distance |
|---|---|
| Behavior Cloning | 701.83 |
| GAIL | 914.45 |
| InfoGAIL \ RB | 1031.13 |
| InfoGAIL \ RA | 1123.89 |
| InfoGAIL \ WGAN | 1177.72 |
| **InfoGAIL (Ours)** | **1226.68** |
| Human | 1203.51 |

# GAIL - Issues

$$\mathbb{E}_{\pi}\left[\log(D(s,a)\right] + \mathbb{E}_{\pi_E}\left[\log(1 - D(s,a)\right]$$

What's the gradient wrt $\pi$ ?

# GAIL - Issues

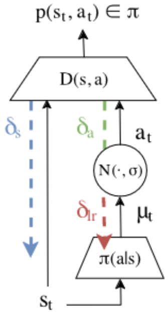
$$\mathbb{E}_{\pi}\left[\log(D(s,a)\right] + \mathbb{E}_{\pi_E}\left[\log(1 - D(s,a)\right]$$

What's the gradient wrt $\pi$ ?

- $\pi$ affects the data distribution but does not appear in the objective
- A common solution is to use REINFORCE-like methods ([Williams, 1992] [Schulman et al., 2015]) that tends to have high variance
- Model-Based methods (MGAIL) make the objective differentiable end-to-end

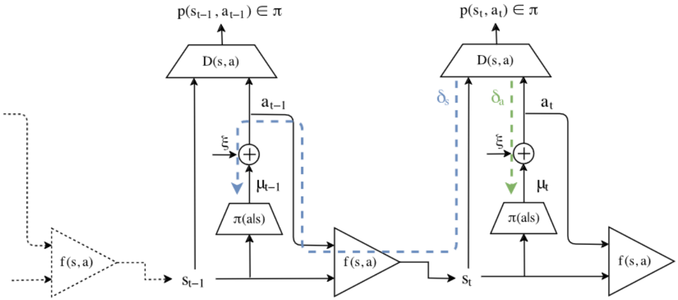
# MGAIL [Baram et al., 2016]

Model-free GAIL



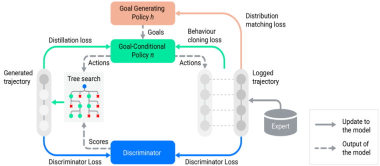
# MGAIL [Baram et al., 2016]

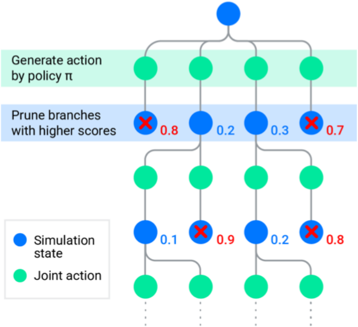
Model-based GAIL



# Symphony [Igl et al., 2022]

- Making simulation realistic by learning agents behaviour from **logged trajectories**
- Parallel Beam Search, Model Based GAIL (MGAIL) and Hierarchical Policies
- Playback Agents and **Interactive Agents**
- Useful when you need **Interactive Agents** in simulation



# Symphony - Beam Search

- Select a set of trajectories from the training set
- Propose a goal with a goal generating policy
- Goal-conditioned policy proposes actions evaluated by the beam search algorithm
- Each node in the tree is evaluated by the discriminator
- Nodes with high score (fake) are **pruned** away
- Nodes with low score are **kept** and the rollout continues

# Symphony - Goal Generation

- Goal: routes $\rightarrow$ sequence of roadgraph lane segments
- For each agent, explore the road graph and use a network to classify the most probable ordered set of segments
- During training, train the network to classify the set of segments with minimal displacement from the GT trajectory
- Goal as input to the control policy

# Symphony - Architecture

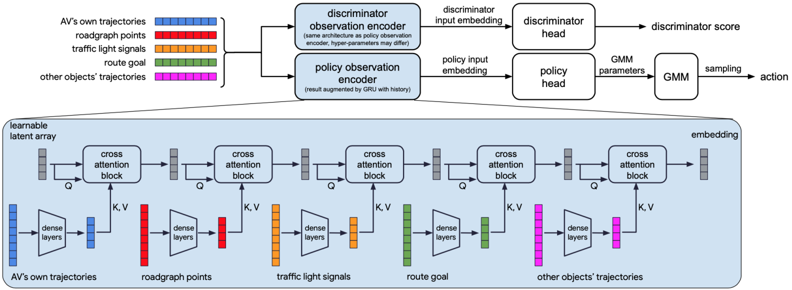
- BC: Discrete action space of 7 x 21 discretized accelerations and steering angles
- MGAIL: Continuous action space specifying the xy displacement (Delta Action Model)

# Symphony - Dataset

- Each training batch: 16 run segments of 10 s. Actions computed at 5 Hz
- Proprietary Dataset: 1.1M run segments
- WOMD: 64.5K run segments

# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving [Bronstein et al., 2022]

- Focus on Ego-vehicle
- Closed loop **evaluation** with simulated agents (Symphony)
- Delta Action model
- BC loss with MGAIL losses



# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving - Method

- High-Level route generation
- Given the scene features and the route the policy outputs the parameters of a GMM
- The GMM gives a distribution over actions (delta-actions)

# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving - Losses

$$\mathcal{L} = \color{red}{\mathcal{L}_{\mathcal{D}}} + \color{blue}{\mathcal{L}_{\mathcal{P}}} + \color{orange}{\mathcal{L}_{\mathcal{BC}}}$$

# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving - Losses

$$\mathcal{L} = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{BC}}$$

$\mathcal{L}_{\mathcal{D}}$: Discriminator Loss (GAIL)

$$\mathcal{L}_{\mathcal{D}} = -\mathbb{E}_{\pi}\left[\log(D(s)\right] - \mathbb{E}_{\pi_E}\left[\log(1 - D(s)\right]$$

# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving - Losses

$$\mathcal{L} = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{BC}}$$

$\mathcal{L}_{\mathcal{D}}$: Discriminator Loss (GAIL)

$$\mathcal{L}_{\mathcal{D}} = -\mathbb{E}_{\pi}\left[\log(D(s)\right] - \mathbb{E}_{\pi_E}\left[\log(1 - D(s)\right]$$

$\mathcal{L}_{\mathcal{P}}$: Policy Loss (GAIL)

$$\mathcal{L}_{\mathcal{P}} = \mathbb{E}_{\pi}\left[\log(D(s)\right]$$

# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving - Losses

$$\mathcal{L} = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{BC}}$$

$\mathcal{L}_{\mathcal{D}}$: Discriminator Loss (GAIL)

$$\mathcal{L}_{\mathcal{D}} = -\mathbb{E}_{\pi}\left[\log(D(s)\right] - \mathbb{E}_{\pi_E}\left[\log(1 - D(s)\right]$$

$\mathcal{L}_{\mathcal{P}}$: Policy Loss (GAIL)

$$\mathcal{L}_{\mathcal{P}} = \mathbb{E}_{\pi}\left[\log(D(s)\right]$$

$\mathcal{L}_{\mathcal{BC}}$: BC Loss

$$\mathcal{L}_{\mathcal{BC}} = -\mathbb{E}_{\pi_E}\left[\log(\pi(s, a))\right]$$

# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving - Interactive agents

- Using just playback agents is not enough for closed loop **evaluation** (especially if we condition on novel routes)
- Road Users should be **interactive** and should diverge in response
- Idea: Use Symphony agents for other Road Users
- Symphony agents are not used during training because they condition the AV on the true route $\rightarrow$ the AV should not diverge too much from the GT trajectory

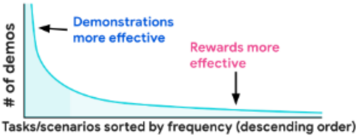
# Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving - Results

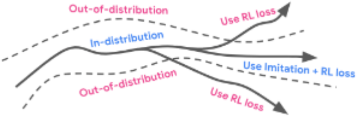
TABLE I: Logged Route on the *Unbiased Test Set*.

| Method | Success rate (%) | Route Failure rate (%) | Collision rate (%) | Off-road rate (%) | Route Progress ratio (%) |
|---|---|---|---|---|---|
| Playback | 98.62±0.08 | 1.07±0.07 | 0.05±0.02 | 0.26±0.03 | 100.00±0.00 |
| BC | 86.07±0.24 | 9.63±0.20 | 4.53±0.14 | 2.21±0.10 | 105.59±0.40 |
| BC + Route | 94.18±0.16 | **0.69±0.06** | 4.60±0.14 | 0.75±0.06 | 98.10±0.33 |
| MGAIL | 88.90±0.21 | 9.73±0.20 | 1.28±0.08 | 1.00±0.07 | 101.22±0.32 |
| MGAIL + Route | 97.45±0.11 | 0.74±0.06 | 1.20±0.07 | 0.77±0.06 | 100.85±0.29 |
| MGAIL + BC | 89.84±0.21 | 8.93±0.19 | 1.25±0.08 | 0.73±0.06 | 105.58±0.36 |
| MGAIL + BC + Route | **98.22±0.09** | **0.69±0.06** | **0.77±0.06** | **0.37±0.04** | 105.30±0.32 |

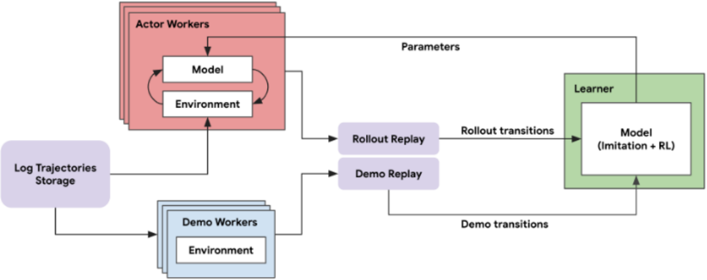
# Imitation Is Not Enough [Lu et al., 2022]

- Imitation Learning often fails to account for safety and reliability
- Imitation Learning with RL using simple rewards



# Imitation Is Not Enough



# Imitation Is Not Enough

Use a mixture of RL and IL objectives:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_t \gamma^r R(s_t, a_t)\right] + \lambda \mathbb{E}_{s,a\sim\pi_E}\left[\log(\pi(s,a))\right]$$

# Imitation Is Not Enough

Simple reward function:

$$R = R_{\text{collision}} + R_{\text{off-road}}$$

$$R_{\text{collision}} = \min(d_{\text{collision}} - 1.0, 0.0)$$

$$R_{\text{off-road}} = \text{clip}(-1.0 - d_{\text{to-edge}}, 0.0, 2.0)$$

# Imitation Is Not Enough

- Do not use reactive agents
- Imitative Loss discourage the learned policy to deviate too far from logs
- In this case the logged trajectories are enough

# Imitation Is Not Enough - Results

| Method | Training | Top1 (%) | Top10 (%) | Top50 (%) | All (%) | Route Progress Ratio, All(%) |
|---|---|---|---|---|---|---|
| BC | All | 9.74±0.49 | 6.72±0.47 | 5.14±0.39 | 4.35±0.27 | 99.00±0.39 |
| MGAIL | All | 7.28±0.98 | 4.22±0.77 | 3.40±0.97 | **2.48±0.29** | **99.55±1.91** |
| SAC | All | 5.29±0.66 | 4.64±1.08 | 4.12±0.74 | 6.66±0.44 | 77.82±8.21 |
| BC-SAC | All | **3.72±0.62** | **2.88±0.23** | **2.64±0.21** | 3.35±0.31 | 95.26±8.64 |
| BC | Top10 | 5.79±0.82 | 3.45±0.72 | 2.71±0.57 | 3.64±0.31 | **98.06±0.18** |
| MGAIL | Top10 | 4.21±0.95 | 2.57±0.52 | 2.20±0.52 | **2.45±0.35** | 96.57±1.19 |
| SAC | Top10 | 4.33±0.47 | 4.11±0.63 | 3.66±0.47 | 5.60±0.86 | 71.05±2.47 |
| BC-SAC | Top10 | **2.59±0.31** | **2.01±0.29** | **1.76±0.20** | 2.81±0.26 | 87.63±0.58 |
| BC | Top1 | 7.66±1.13 | 7.84±0.92 | 6.63±0.78 | 6.85±0.65 | **94.10±1.00** |
| MGAIL | Top1 | 4.24±0.95 | 3.16±0.43 | 2.74±0.46 | 3.79±0.46 | 93.10±11.72 |
| SAC | Top1 | 4.15±0.31 | 3.87±0.12 | 3.46±0.16 | 5.98±1.03 | 75.63±2.19 |
| BC-SAC | Top1 | **3.61±0.87** | **2.96±1.11** | **2.69±0.87** | **3.38±0.48** | 75.00±17.21 |

Table 2: Failure rates (lower is better) and progress ratios (higher is better) of BC-SAC and baselines on different training/evaluation subsets.

# Imitation Is Not Enough - Results

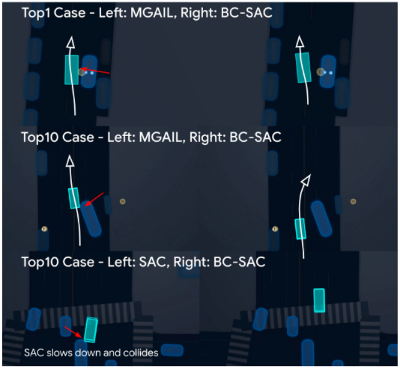
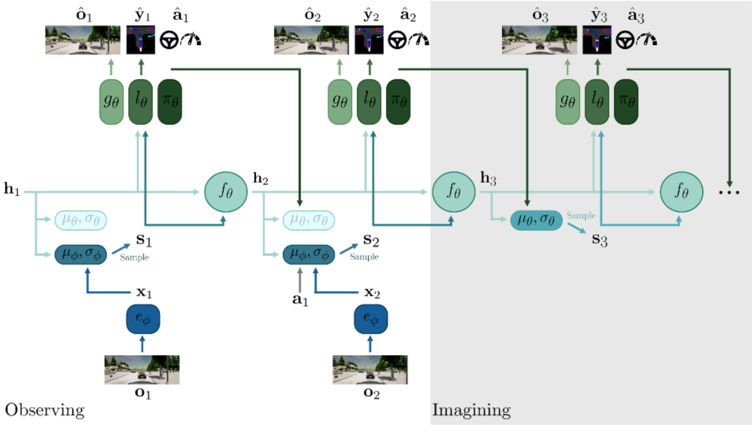



Figure 6: Visualizations of win cases against baseline agents. The cyan car is controlled. **Example 1**: MGAIL collides with a pedestrian coming out of a double parked car while BC-SAC was able to leave an appropriately wide-clearance. **Example 2**: MGAIL does not provide sufficient clearance and collides with the incoming vehicle. **Example 3**: SAC slows down in an intersection resulting in an rear collision. In contrast, BC-SAC keeps an appropriate speed profile through the intersection without a collision.

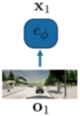
# Model-Based Imitation Learning for Urban Driving [Hu et al., 2022]

- Related to World Models [Ha and Schmidhuber, 2018]
- Jointly learns a model of the world and a policy
- Leverages 3D geometry
- Trained offline without online interaction with the environment
- Can predict diverse and plausible states and actions, that can be interpretably decoded to BEV semantic segmentation

# Model-Based Imitation Learning for Urban Driving



# Model-Based Imitation Learning for Urban Driving

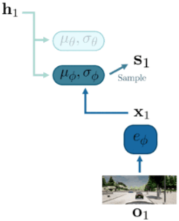



The observation encoder $e_\phi$ **lifts** image features **to 3D**, pools them to **bird's-eye view**, and compresses them to a **1D vector.**

Observing

1/5

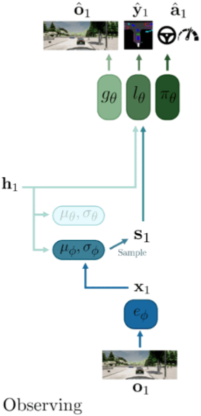
# Model-Based Imitation Learning for Urban Driving




Observing

The **prior distribution** $\mathcal{N}(\mu_\theta(\mathbf{h}_1), \sigma_\theta(\mathbf{h}_1)\boldsymbol{I})$ , representing what the model imagines will happen, and the **posterior distribution** $\mathcal{N}(\mu_\phi(\mathbf{h}_1, \mathbf{x}_1), \sigma_\phi(\mathbf{h}_1, \mathbf{x}_1)\boldsymbol{I})$ , representing what actually happened, **are matched** (KL divergence).

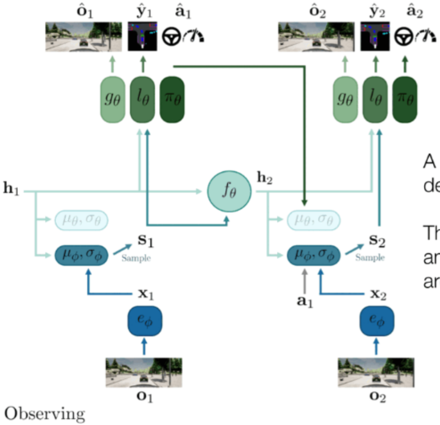
# Model-Based Imitation Learning for Urban Driving




The **image decoder** $g_\theta$ and **bird's-eye view decoder** $l_\theta$ output the reconstructed observation and bird's-eye segmentation respectively.

The **driving policy** $\pi_\theta$ outputs the vehicle control.

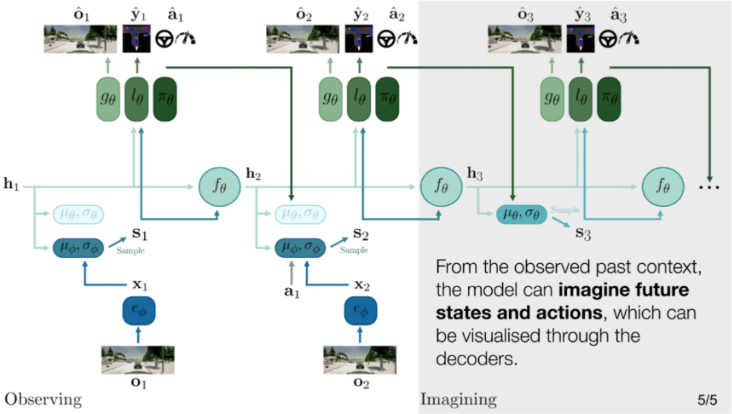
# Model-Based Imitation Learning for Urban Driving




A recurrent network computes the deterministic transition $\mathbf{h}_2 = f_\theta(\mathbf{h}_1, \mathbf{s}_1)$.

The prior $\mathcal{N}(\mu_\theta(\mathbf{h}_2, \hat{\mathbf{a}}_1), \sigma_\theta(\mathbf{h}_2, \hat{\mathbf{a}}_1)\boldsymbol{I})$ and posterior $\mathcal{N}(\mu_\phi(\mathbf{h}_2, \mathbf{a}_1, \mathbf{x}_2), \sigma_\phi(\mathbf{h}_2, \mathbf{a}_1, \mathbf{x}_2)\boldsymbol{I})$ are matched again.

# Model-Based Imitation Learning for Urban Driving

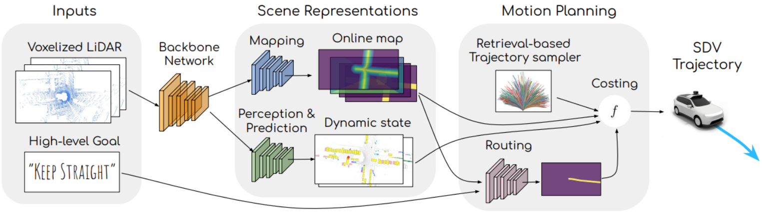


# Model-Based Imitation Learning for Urban Driving

- Decoding used just for closed loop **evaluation** (not training)
- Decoding just used as **auxiliary task**
- "For simplicity, we have set the weight parameter of the image reconstruction to 0." Cit.

# MP3 [Casas et al., 2021]

- Map, Perceive, Predict, Plan
- Input: 10 LiDAR sweeps (1s)
- Range: 140 x 80
- Resolution: 0.2m/voxels



# MP3

Map:

- Drivable area
- Reachable lanes
- Intersection

Occupancy:

- Current Occupancy
- Forward occupancy flow

Routing:

- For each pixel: probability that driving to it is aligned with the input plan
- 3 Networks: turn right, turn left, go straight

# More

- Learning from All the Vehicles (LAV) [Chen and Krähenbühl, 2022]
- Reinforcement Learning Coach (ROACH) [Zhang et al., 2021]
- TransFuser: Multi-Modal Fusion Transformer for End-to-End Autonomous Driving [Chitta et al., 2022]

# Our case

- BC training is ready using the recorded sequences
- Closed loop training requires a simulator
- Probably our simulator is not yet ready
  - Sim2real?
  - Environments?
  - Experts?
- Input representation problem:
  - Point cloud$\rightarrow$ simulation is difficult
  - Images $\rightarrow$ simulation is very difficult
  - Boxes $\rightarrow$ simulation is easy
- MVMap can help in the scenario generation but what about radars?

# Bibliography I


Baram, N., Anschel, O., and Mannor, S. (2016).
Model-based adversarial imitation learning.
*arXiv preprint arXiv:1612.02179*.

Bronstein, E., Palatucci, M., Notz, D., White, B., Kuefler, A., Lu, Y., Paul, S., Nikdel, P., Mougin, P., Chen, H., et al. (2022).
Hierarchical model-based imitation learning for planning in autonomous driving.
In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8652–8659. IEEE.

Casas, S., Sadat, A., and Urtasun, R. (2021).
Mp3: A unified model to map, perceive, predict and plan.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412.

# Bibliography II


Chen, D. and Krähenbühl, P. (2022).
Learning from all vehicles.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231.

Chen, D., Zhou, B., Koltun, V., and Krähenbühl, P. (2020).
Learning by cheating.
In *Conference on Robot Learning*, pages 66–75. PMLR.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016).
Infogan: Interpretable representation learning by information maximizing generative adversarial nets.
*Advances in neural information processing systems*, 29.

# Bibliography III


Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., and Geiger, A. (2022).
Transfuser: Imitation with transformer-based sensor fusion for autonomous driving.
*IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ha, D. and Schmidhuber, J. (2018).
World models.
*arXiv preprint arXiv:1803.10122*.

Ho, J. and Ermon, S. (2016).
Generative adversarial imitation learning.
*Advances in neural information processing systems*, 29.

# Bibliography IV


Hu, A., Corrado, G., Griffiths, N., Murez, Z., Gurau, C., Yeo, H., Kendall, A., Cipolla, R., and Shotton, J. (2022).
Model-based imitation learning for urban driving.
*arXiv preprint arXiv:2210.07729*.

Igl, M., Kim, D., Kuefler, A., Mougin, P., Shah, P., Shiarlis, K., Anguelov, D., Palatucci, M., White, B., and Whiteson, S. (2022).
Symphony: Learning realistic and diverse agents for autonomous driving simulation.
*arXiv*.

Li, Y., Song, J., and Ermon, S. (2017).
Infogail: Interpretable imitation learning from visual demonstrations.
*Advances in Neural Information Processing Systems*, 30.

# Bibliography V


Lu, Y., Fu, J., Tucker, G., Pan, X., Bronstein, E., Roelofs, B., Sapp, B., White, B., Faust, A., Whiteson, S., et al. (2022).
Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios.
*arXiv preprint arXiv:2212.11419*.

Ross, S., Gordon, G., and Bagnell, D. (2011).
A reduction of imitation learning and structured prediction to no-regret online learning.
In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015).
Trust region policy optimization.
In *International conference on machine learning*, pages 1889–1897. PMLR.

# Bibliography VI


Williams, R. J. (1992).
Simple statistical gradient-following algorithms for connectionist reinforcement learning.
*Reinforcement learning*, pages 5–32.

Zhang, Z., Liniger, A., Dai, D., Yu, F., and Van Gool, L. (2021).
End-to-end urban driving by imitating a reinforcement learning coach.
In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15222–15232.