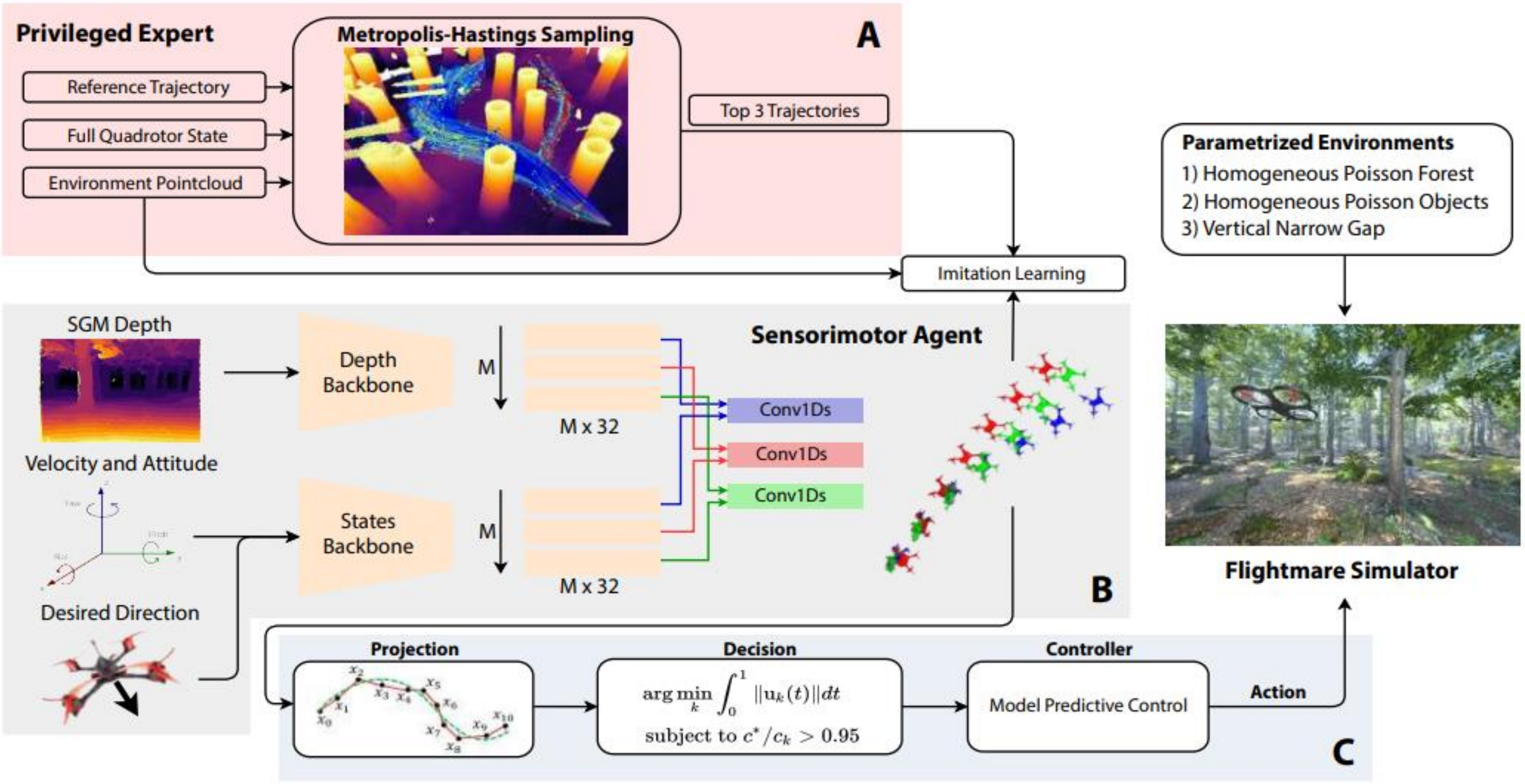
Review in algorithmic development

For **Baseline #1:** Learning High-Speed flight in the wild

(Loquercio et al., 2021)

Overall System Architecture



The Privileged Expert

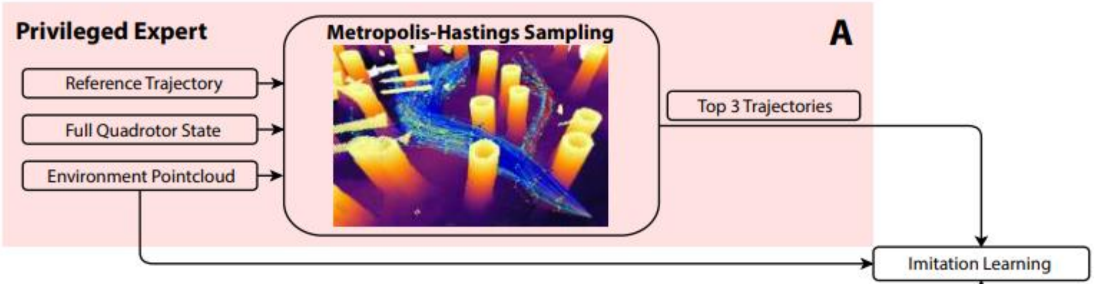
- Sampling-based Motion-Planning Algorithm.
  (M-H Sampling)
- Distribution of collision-free trajectories:

$$P(\boldsymbol{\tau} \mid \boldsymbol{\tau}_{\text{ref}}, \mathcal{C}) = \frac{1}{Z} \exp(-c(\boldsymbol{\tau}, \boldsymbol{\tau}_{\text{ref}}, \mathcal{C}))$$

- Cost Function:

$$c(\boldsymbol{\tau}, \boldsymbol{\tau}_{\text{ref}}, \mathcal{C}) = \int_0^1 \lambda_c C_{collision}(\boldsymbol{\tau}(t)) + \int_0^1 [\boldsymbol{\tau}(t) - \boldsymbol{\tau}_{ref}(t)]^\top \boldsymbol{Q} [\boldsymbol{\tau}(t) - \boldsymbol{\tau}_{ref}(t)] \, dt$$

$$C_{collision}(\boldsymbol{\tau}(t)) = \begin{cases} 0 & \text{if } d_c > 2r_q \\ -d_c^2 / r_q^2 + 4 & \text{otherwise.} \end{cases}$$



The Privileged Expert

Analytically computing P is intractable because:

❖ Arbitrary obstacles

❖ Multimodality

- M-H algorithm is used to approximate the density P.
- **Target score function:** $s(\boldsymbol{\tau}) = \exp(-c(\boldsymbol{\tau}, \boldsymbol{\tau}_{\text{ref}}, \mathcal{C}))$ (because $s(\boldsymbol{\tau}) \propto P(\boldsymbol{\tau} \mid \boldsymbol{\tau}_{\text{ref}}, \mathcal{C})$ so there are asymptotic convergence guarantees for all different modes of P.
- **Representation of trajectories:** Cubic B-splines $\bar{\tau}_{bspline} \in \mathbb{R}^{3\times3}$ with three control points and a uniform knot vector (interpolation with high computational efficiency).
- **Sampling:** Instead of naively sampling trajectory states, the B-splines control points are sampled (in a spherical coordinate system) to vary the state of the trajectory.
- Can’t run in real-time. (Large sampling computational overhead)

The sensorimotor agent (student policy)

- Real-time trajectory generation: 1. Depth (Semi-Global matching), 2. Velocity and attitude, 3. Flight direction vector
- Challenges: Partially observable environment, P is multimodal
- NN Policy: Latent encoding of inputs, o/p: M=3 trajectories + respective collision costs
- MoblieNet V3 architecture is used to extract features from depth images. (pre-trained)
- Then processed by 1D conv. To generate 3 feature vectors of size 32.
- The visual and state features are again processed individually (using 4-layered MLP).
- …….
- Final output: Trajectory-cost (relative) pair for each mode.

$$\mathcal{T}_n = \{(\boldsymbol{\tau}_n^k, c_k) \mid k \in [0, 1, \ldots, M-1]\}, \text{ where } c_k \in \mathbb{R}_+$$

- The network traj. Does not describe full state evolution but only its position component.

The sensorimotor agent (student policy)

- Network Trajectory: $\boldsymbol{\tau}_n^k = [\ \boldsymbol{p}(t_i)\ ]_{i=1}^{10}, \quad t_i = \frac{i}{10},$

- More general representation than B-splines. (to handle computation workload of interpolation during sampling)
- Training: NN is trained using supervised learning (on 3 trajectories with lowest cost found by the expert)
- Multi-Hypotheses Prediction: (Relaxed- Winner-Takes-All) loss for each sample

$$\text{R-WTA}(\mathcal{T}_e, \mathcal{T}_n) = \sum_{i=0}^{|\mathcal{T}_e|} \sum_{k=0}^{|\mathcal{T}_n|} \alpha(\boldsymbol{\tau}_{e,p}^i, \boldsymbol{\tau}_n^k) \|\boldsymbol{\tau}_{e,p}^i - \boldsymbol{\tau}_n^k\|^2$$

$$\alpha(\boldsymbol{\tau}_{e,p}^i, \boldsymbol{\tau}_n^k) = \begin{cases} 1-\epsilon & \text{if } \|\boldsymbol{\tau}_{e,p}^i - \boldsymbol{\tau}_n^k\|^2 \leq \|\boldsymbol{\tau}_{e,p}^j - \boldsymbol{\tau}_n^k\|^2 \quad \forall j \neq i \\ \frac{\epsilon}{M-1} & \text{otherwise.} \end{cases}$$

The sensorimotor agent (student policy)

- Final training loss: $$\mathcal{L}((\mathcal{T}_e, \mathcal{T}_n) = \lambda_1 \text{R-WTA}(\mathcal{T}_e, \mathcal{T}_n) + \lambda_2 \sum_{k=0}^{|\mathcal{T}_n|} \|c_k - C_{collision}(\boldsymbol{\tau}_n^k)\|^2,$$

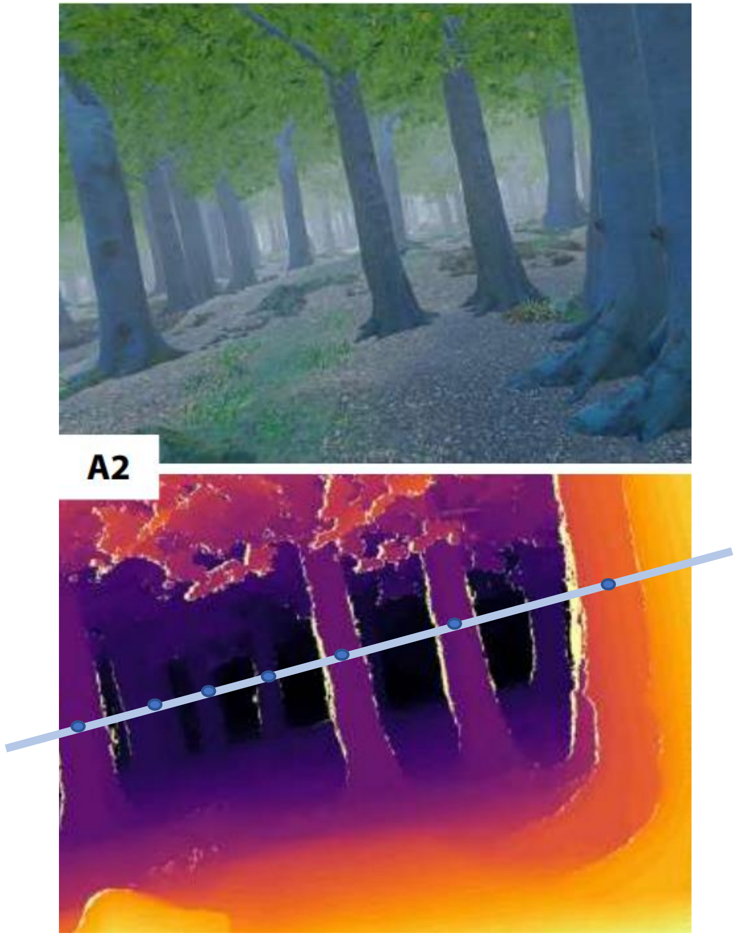
Possible Improvements

Better Heuristics:

- In the privileged expert module, more informed heuristics can be added, hence ultimately leveraging the available information.
- Reference: Global collision-free trajectory (BIT*)
- Testing Different MCMC-based sampling methods

Local Plan:

- Instead of using just the nearest point, we could use a combination of navigational FOV and global heuristics to generate real-time navigation. (Faster plans) (Teacher)
- For safety considerations, We could just link these objects (as points) with weighted graphs and (Student).



Possible Improvements

- **Ensemble planning:** Switching between optimizers (already available) (Keeping multiple options during training)
- Example: CHOMP for narrow gaps
- Searching for an optimal trajectory (primitives) in a lower dimensional space
- End-game planning database
- Training with better heuristics vs testing with better heuristics