8 Appendix

8.1 Difficulty Bucket Statistics

The summary statistics of the difficulty scores in each bucket of the training and test datasets are presented in Tables 2 and 3, respectively.

8.2 Geometric Schedule

The Geometric-schedule-10% uses a schedule that starts training by weighting each bucket equally (i.e., the unbiased dataset), and ends by weighting each bucket proportional to the average difficulty score of the segments it contains. Specifically, at step t, the sampling weight for bucket k is $q_k = (q_k^i - q_k^f)\alpha^t + q_k^f$, where q_k^i and q_k^f are the initial and final weights for bucket k, and α is the common ratio of the geometric progression. The sample weights for all the buckets are then normalized to sum to 1 to acquire sampling probabilities. For the Geometric-schedule-10% variant, we set $\alpha = 0.999975$ and $q_k^i = 1$ for each bucket, and $\{q_f\}_{k=1}^{10}$ is given by the "Mean" row in table 2. This progression is visualized in Figure 4.

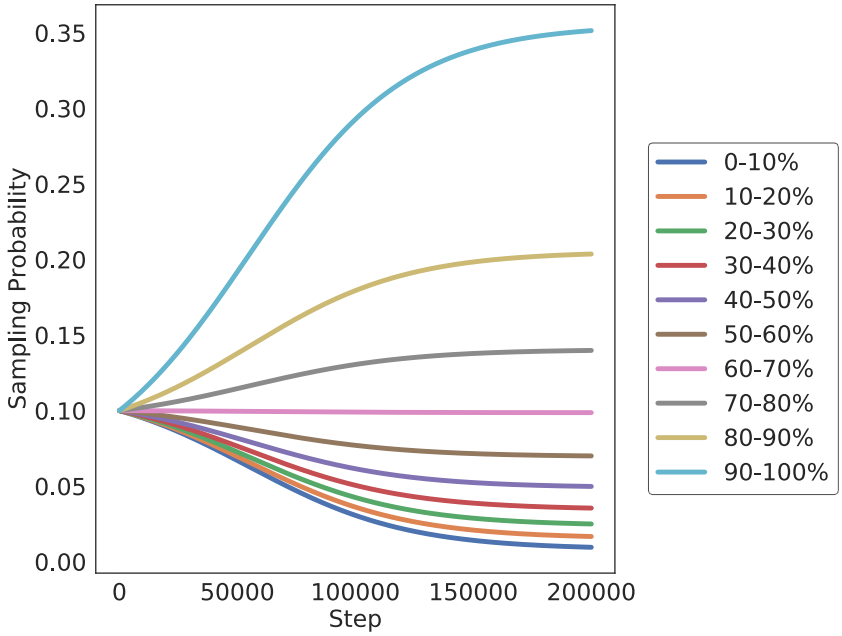


Figure 4: Sampling schedule of each bucket for the Geometric-schedule-10% variant.

Table 2: Summary statistics of the difficulty scores in each training data bucket.

	0-10%	10-20%	20-30%	30-40%	40-50%	50-60%	60-70%	70-80%	80-90%	90-100%
Min	0.001	0.019	0.031	0.046	0.066	0.094	0.133	0.189	0.270	0.407
Mean	0.013	0.025	0.038	0.056	0.079	0.112	0.159	0.227	0.331	0.573
Max	0.019	0.031	0.046	0.066	0.094	0.133	0.189	0.270	0.407	0.939

Table 3: Summary statistics of the difficulty scores in each test data bucket.

	0-10%	10-20%	20-30%	30-40%	40-50%	50-60%	60-70%	70-80%	80-90%	90-100%
Min	0.001	0.016	0.026	0.040	0.059	0.085	0.122	0.176	0.258	0.396
Mean	0.011	0.021	0.033	0.049	0.071	0.103	0.147	0.214	0.320	0.562
Max	0.016	0.026	0.040	0.059	0.085	0.122	0.176	0.258	0.396	0.939

8.3 Uniform Variant

The Uniform-10% variant sets the sampling weight of each bucket to be proportional to the range of difficulty scores of each bucket. The score ranges for the 10 training buckets are $[0.0180, 0.0126, 0.0150, 0.0199, 0.0276, 0.0392, 0.0557, 0.0814, 0.1368, 0.5324]$.

8.4 Metrics by Bucket

In Figures 5 and 6 we present the performance of each training variant and baseline for each bucket. The clear upward trend in collision rate with the increasing bucket scores demonstrates that collisions are highly correlated with the difficulty scores. We observe a similar, but not quite as strong, correlation in the route failure rate and off-road rate as well.

8.5 Adaptive Importance Sampling Variants

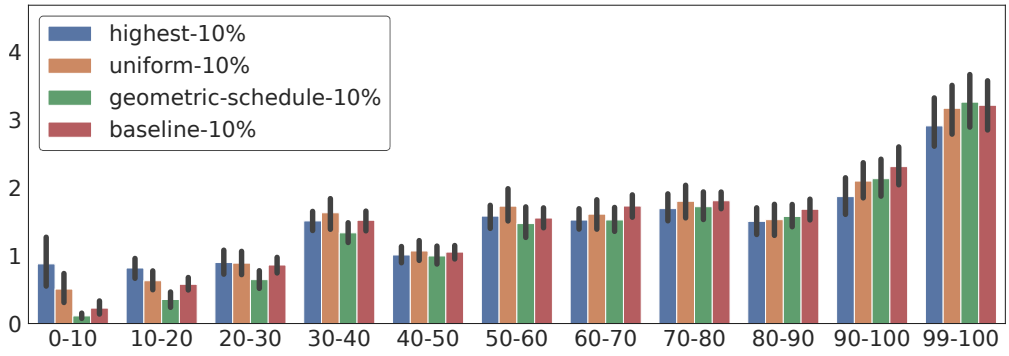
We conducted a series of experiments that perform adaptive importance sampling, wherein we upsample certain buckets based on the agent's performance during training, but then we correct for this upsampling via a *likelihood ratio*. Importance sampling measures the expectation of a statistic over a distribution P using a different distribution Q . In particular, $E_P[f(X)] = E_Q[f(X)w(X)]$, where $w(x) := p(x)/q(x)$ is the likelihood ratio for density functions p and q from distributions P and Q, respectively.

Our nominal distribution P is the natural distribution of run segments. Due to the infrastructure challenges surrounding large training datasets mentioned in Section 4.2, we implemented reweighting on the level of buckets rather than on the level of individual run segments. This allows our method to easily scale to arbitrarily large datasets since it depends only on the number of buckets, not on the dataset size. In this setting, the nominal density is $p_i := 1/N$. We constructed the sampling distribution Q as follows: every K training steps, we collected the average policy loss per bucket $(\bar{\mathcal{L}}_{\mathcal{P}})_i$ over the preceding window of K steps. Since our losses can be positive or negative, we set the sampling weights $q_i \propto \exp\left(\gamma \cdot (\bar{\mathcal{L}}_{\mathcal{P}})_i\right)$, where γ is the inverse temperature parameter. We also dedicated a small constant probability mass ϵ to all buckets that were not sampled during the last K iterations, which ensures a nonzero probability of sampling a run segment from any given bucket. During training, we multiplied the loss for a segment from bucket i by the ratio $w_i := 1/(N \cdot q_i)$. To evaluate the effect of different degrees of importance reweighting, we also considered w_i^β for different values of $\beta \in [0, 1]$. Algorithm 1 describes this procedure in the context of training the planning agent. We note that this approach is similar to Prioritized Experience Replay (PER) [23], but adapted to our setting with priority weights assigned over a discrete set of buckets.

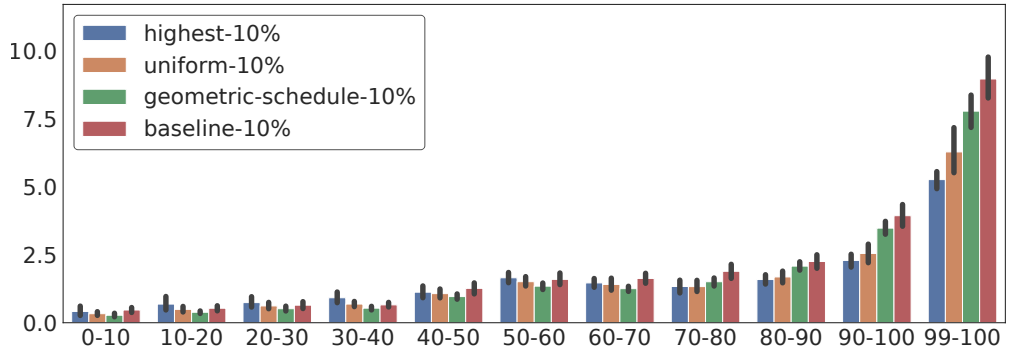
We expanded on this approach using Distributionally Robust Optimization (DRO) [42], which introduces an additional loss weighting term with hyperparameter $\rho \in [0, 1]$. Larges values of ρ allow for a greater deviation of the loss weights from the importance sampling weights that would be needed to exactly account for the non-uniform sampling.

We show the dataset sampling probabilities q_i for the PER variant in Figure 7 with two settings of the inverse temperature parameter γ: "PER ($\gamma = 0.1, \beta = 1$)-10%" and "PER ($\gamma = 1, \beta = 1$)-10%". While $\gamma = 0.1$ results in a distribution that is close to uniform, $\gamma = 1$ quickly produces a heavily skewed distribution that samples from the most difficult bucket at least 75% of the time. In both cases, the sampling probability of each bucket is directly related to its difficulty scores: the higher a bucket's difficulty scores, the more frequently it is sampled. This clearly demonstrates that a run segment's difficulty score is a strong predictor for how challenging it will be for a planning agent to navigate successfully.

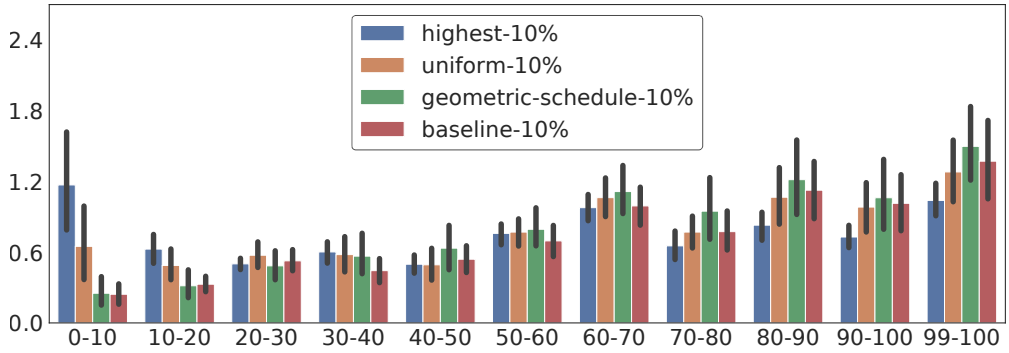
We present our results for PER and DRO in Table 4 with different values of γ, β, and ρ. For these experiments, we use 10% of the available training data, and we set $K = 1000$ and $\epsilon = 3.125 \times 10^4$. We observe that certain settings of PER and DRO achieve the lowest route failure and off-road rates. They also result in the best collision rate, overall failure rate, and route progress ratio, though other non-adaptive variants achieve comparable results that are within the confidence bounds. This suggests that adaptive importance sampling is a promising curriculum learning approach that can provide comparable or better results to fixed sampling strategies without the need for hand-tuning custom sampling weights and schedules.



(a) Route failure rate (%)

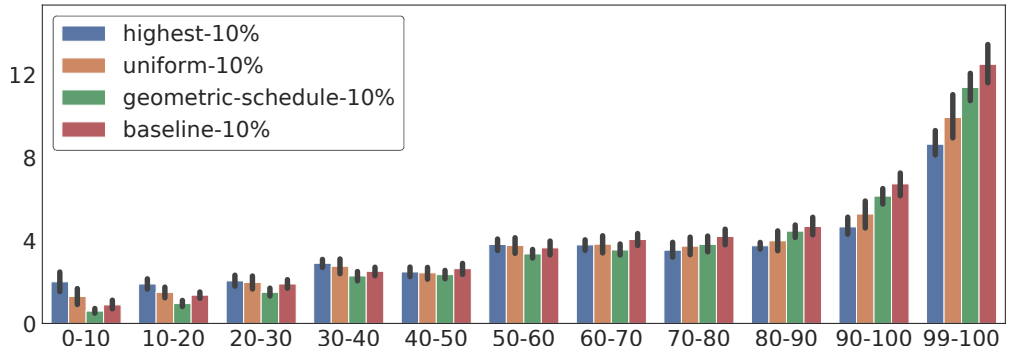


(b) Collision rate (%)

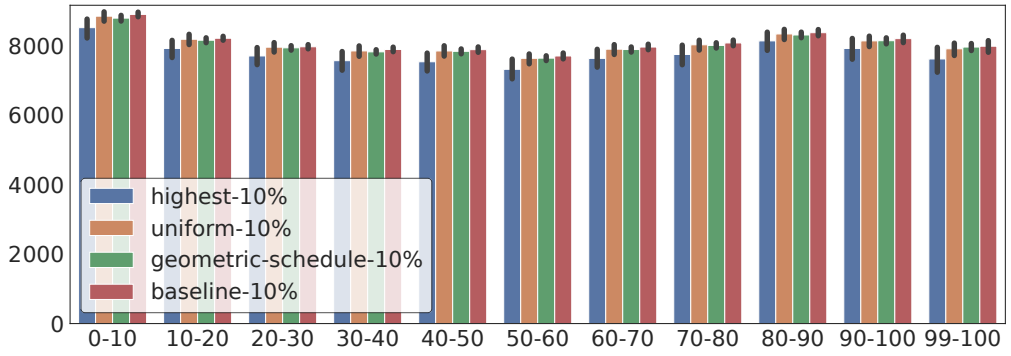


(c) Off-road rate (%)

Figure 5: Metrics by test decile bucket for each of the training variants.



(a) Overall failure rate (%)



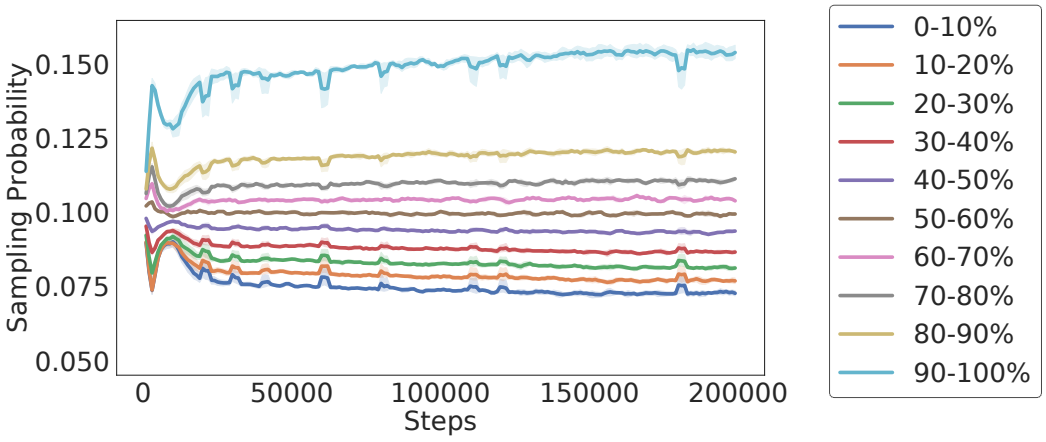
(b) Progress rate (%)

Figure 6: Metrics by test decile bucket for each of the training variants.

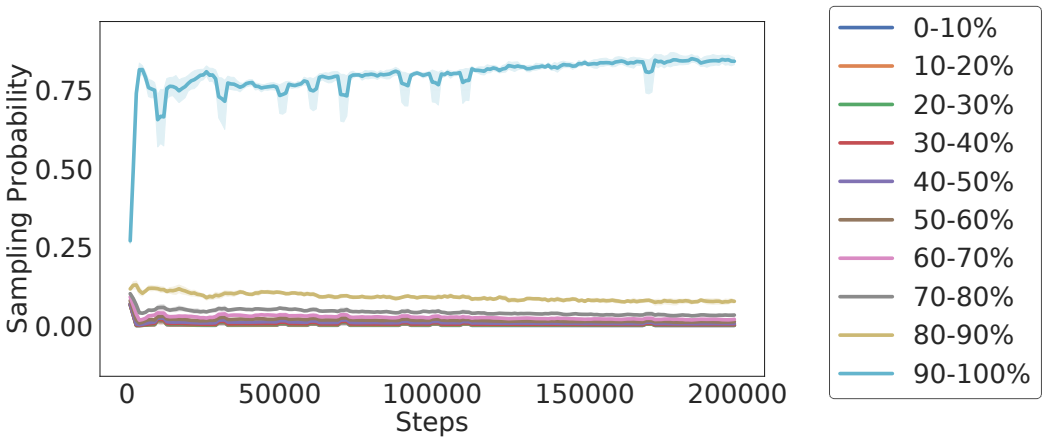
Algorithm 1 MGAIL curriculum training with bucket-wise adaptive importance sampling

Input: datasets $\{D\}_{i=1}^{N}$, sampling period K, step size η, inverse temperature parameter γ, IS weight exponent β, budget T, minibatch size B

Initialize sampling probabilities $q_i = \frac{1}{N}$, loss buffers $\mathcal{H}_i = \emptyset$ for $i = 1, ..., N$
for $t = 1$ **to** T **do**
  if $t \equiv 0 \mod K$ **then**
    Compute per-dataset mean policy loss $(\bar{\mathcal{L}}_{\mathcal{P}})_i$ from each buffer $\mathcal{H}_i$
    Compute dataset sampling weights $q_i \leftarrow \exp(\gamma \cdot (\bar{\mathcal{L}}_{\mathcal{P}})_i)$
    Normalize dataset sampling probabilities $q_i \leftarrow q_i / \sum_{j=1}^{N} q_j$
    Reset $\mathcal{H}_i = \emptyset$ for $i = 1, ..., N$
  end if
  Sample B dataset indices $\{b\}_{i=1}^{B} \sim Q$, where Q has probability mass function q
  Sample minibatch $\{x\}_{i=1}^{B}$, where $x_i \overset{\text{i.i.d.}}{\sim} U[D_{b_i}]$ ▷ Example x_i is from dataset D_{b_i}
  Compute per-example policy losses $\mathcal{L}_{\mathcal{P}}(x_i)$ and discriminator losses $\mathcal{L}_{\mathcal{D}}(x_i)$
  Add $\texttt{stopgrad}(\mathcal{L}_{\mathcal{P}}(x_i))$ to $\mathcal{H}_{b_i}$
  Compute IS weights $w_i \leftarrow 1/(N \cdot q_i)$
  Compute weighted policy losses $l_{\mathcal{P}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} w_{b_i}^{\beta} \mathcal{L}_{\mathcal{P}}(x_i)$
  Compute weighted discriminator losses $l_{\mathcal{D}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} w_{b_i}^{\beta} \mathcal{L}_{\mathcal{D}}(x_i)$
  Update policy weights $\theta \leftarrow \theta + \eta \cdot \nabla_{\theta} l_{\mathcal{P}}$
  Update discriminator weights $\omega \leftarrow \omega + \eta \cdot \nabla_{\omega} l_{\mathcal{D}}$
end for



(a) $\gamma = 0.1$



(b) $\gamma = 1$

Figure 7: Dataset sampling probabilities throughout training for the PER adaptive importance sampling variant for two different values of the inverse temperature parameter γ.

8.6 Other Variants

We implemented two additional variants, which differ from the other variants in their training dataset sizes and sampling strategies. Their performance is shown in Table 4.

1. The "Highest-1%" variant is an agent trained on only the top 1% of training examples ordered by difficulty score. As such, it is not strictly comparable to the other variants which use 10% of the data. Our results show that while it achieves a lower collision rate than the baselines, its performance on all other metrics is worse. This demonstrates that extreme upsampling strategies on the most difficult examples, combined with using significantly less training data, can lead to worse overall performance. However, the fact that its collision rate is lower than that of the baselines suggests that upsampling difficult segments has a strong positive effect on metrics that are highly correlated with the difficulty score. It is possible that incorporating other metrics into our definition of "difficulty" for the difficulty model could improve this variant's performance on those metrics as well.
2. We also trained the "Highest-10% + Lowest-10%" variant on the combination of the most difficult bucket and the least difficult bucket. This variant achieves among the best performance overall, matching that of the Geometric-schedule-10% variant. By incorporating the least difficult bucket, it addresses the shortcomings of Highest-10%, which has high failure rates on segments in the 0-40% range of difficulty scores. However, this variant uses 20% of the available data, twice as much as the other variants.

Table 4: Evaluation of agent variants and baselines on the full unbiased test set (mean ± standard error of each metric across 10 seeds, unless noted otherwise). For all metrics except route progress, lower is better.

Agent Variant	Route Failure rate (%)	Collision rate (%)	Off-road rate (%)	Route Progress ratio (%)	Failure rate (%)
Baseline-all	1.38±0.13	1.46±0.09	0.73±0.07	**81.21±0.39**	3.33±0.20
Baseline-10%	1.34±0.06	1.50±0.09	0.67±0.06	81.12±0.37	3.28±0.13
Baseline-lowest-10%	1.14±0.05	4.15±0.11	0.98±0.10	**81.88±0.41**	5.91±0.13
Highest-10%	1.33±0.06	1.23±0.09	0.74±0.02	77.95±1.33	3.10±0.10
Uniform-10%	1.35±0.09	**1.17±0.08**	0.75±0.07	80.67±0.73	**3.07±0.17**
Highest-1%	1.53±0.08	1.39±0.06	0.99±0.11	79.35±1.18	3.66±0.13
Highest-10% + Lowest-10%	1.18±0.06	1.28±0.07	0.65±0.06	79.97±0.81	**2.94±0.12**
Geometric-schedule-10%	1.19±0.07	1.25±0.04	0.74±0.10	80.48±0.36	**2.92±0.11**
PER($\gamma = 0.1, \beta = 0$)-10% (6 seeds)	1.37±0.06	1.32±0.02	**0.55±0.07**	**81.91±0.58**	2.99±0.05
PER($\gamma = 0.1, \beta = 0.5$)-10% (7 seeds)	1.28±0.09	1.39±0.11	0.73±0.11	**82.89±0.68**	3.15±0.20
PER($\gamma = 0.1, \beta = 1$)-10%	1.31±0.09	1.63±0.09	**0.51±0.05**	80.34±0.47	3.28±0.14
PER ($\gamma = 1, \beta = 0$)-10%	1.28±0.08	**1.06±0.03**	0.71±0.08	81.16±0.88	**2.88±0.08**
PER($\gamma = 1, \beta = 0.5$)-10%	1.21±0.06	1.47±0.10	0.90±0.12	**82.60±0.69**	3.36±0.20
PER($\gamma = 1, \beta = 1$)-10%	1.20±0.06	1.99±0.07	1.06±0.21	**82.34±0.64**	3.93±0.25
DRO ($\gamma = 0.1, \beta = 0, \rho = 0.25$) 10% (4 seeds)	1.23±0.10	1.34±0.13	0.85±0.12	80.01±0.52	3.27±0.18
DRO ($\gamma = 0.1, \beta = 1, \rho = 0.05$) 10% (8 seeds)	1.19±0.09	1.16±0.04	0.69±0.07	80.86±0.65	**2.83±0.10**
DRO ($\gamma = 0.1, \beta = 1, \rho = 0.25$) 10% (9 seeds)	1.24±0.03	1.49±0.08	0.70±0.03	81.23±0.48	3.25±0.08
DRO ($\gamma = 0.1, \beta = 1, \rho = 1$) 10% (4 seeds)	**1.02±0.04**	1.65±0.09	0.87±0.21	78.28±0.81	3.43±0.18
DRO ($\gamma = 1, \beta = 0, \rho = 0.25$) 10% (4 seeds)	1.27±0.15	1.31±0.07	**0.58±0.06**	79.43±1.02	**3.04±0.24**
DRO ($\gamma = 1, \beta = 0.5, \rho = 0.25$) 10% (7 seeds)	1.33±0.05	1.18±0.08	0.73±0.06	80.72±1.22	3.02±0.06
DRO ($\gamma = 1, \beta = 1, \rho = 0.05$) 10% (7 seeds)	1.20±0.02	1.58±0.05	0.72±0.12	81.72±0.46	3.31±0.14
DRO ($\gamma = 1, \beta = 1, \rho = 0.25$) 10% (6 seeds)	1.18±0.09	1.65±0.10	1.12±0.29	**82.22±0.91**	3.70±0.37
DRO ($\gamma = 1, \beta = 1, \rho = 1$) 10% (5 seeds)	1.28±0.14	2.33±0.19	1.95±0.20	77.97±1.84	5.14±0.20

8.7 Planning Agent Details

We use the same hierarchical planning agent described in Bronstein et al. [41]; additional details can be found in that work. This planning agent consisting of a high-level route-generation policy and a low-level action policy trained using MGAIL. The high-level policy uses an A* search to produce multiple lane-specific routes through a pre-mapped roadgraph and selects the lowest-cost route. We can either evaluate the low-level policy in a standalone fashion by conditioning it on a given route, or the high-level and low-level policies together by allowing the agent to choose its own goal routes given a destination. The low-level action policy and discriminator use stacked transformer-based observation models [43, 44] to encode the goal route, AV's state, other vehicles' states, roadgraph points, and traffic light signals. Similar to Set-Transformer [45] and Perceiver [46], this observation encoder uses learned latent queries and a stack of cross-attention blocks, one for each group of features. A delta actions model is used for the AV's dynamics, where the action a is the offset from the current state s: $s' = s + a$. The policy head predicts the parameters (weights, means, and covariances) of a Gaussian Mixture Model (GMM) with 8 Gaussians, used to parameterize the delta actions. We trained the action policy and discriminator using a combination of MGAIL and behavior cloning (BC). The total policy loss is given by $\lambda_{\mathcal{P}}\mathcal{L}_{\mathcal{P}} + \lambda_{\mathcal{BC}}\mathcal{L}_{\mathcal{BC}}$, where $\mathcal{L}_{\mathcal{P}} = -\mathbb{E}_{s\sim\pi_\theta}[\log\ D_\omega(s)]$ is the MGAIL policy loss, $\mathcal{L}_{\mathcal{BC}} = -\mathbb{E}_{s,a\sim\pi_E}[\log\ \pi_\theta(a|s)]$ is the BC loss, and $\lambda_{\mathcal{P}}$ and $\lambda_{\mathcal{BC}}$ are hyperparameters. The MGAIL discriminator loss is $\mathcal{L}_{\mathcal{D}} = \mathbb{E}_{s\sim\pi_\theta}[\log\ D_\omega(s)] + \mathbb{E}_{s\sim\pi_E}[\log(1 - D_\omega(s))]$. The discriminator is only conditioned on the state s as in [47]. During backpropagation, only the policy parameters θ are updated for $\mathcal{L}_{\mathcal{P}}$ and $\mathcal{L}_{\mathcal{BC}}$, and only the discriminator parameters ω are updated for $\mathcal{L}_{\mathcal{D}}$.

8.8 Evaluation With Interactive Agents

A potential concern is that the planning agent is trained and evaluated with other agents replaying their logged trajectories. This may result in unrealistic behavior if the planning agent behaves in a significantly different way than the logged AV and other agents don't react realistically. It is possible that a planning agent trained in this way would not perform well when deployed in the real world, in which other road users influence and interact with the AV. To determine whether this is an issue, we evaluated our planning agent alongside interactive agents controlling a subset of other vehicles in the scene. The interactive agent policy, which was trained separately, has the same model architecture, dynamics model, and training loss function as our planning agent. The main difference is that the

interactive agent is not goal-conditioned because its task is to drive in a realistic manner and not necessarily reach a specific destination.

For each bucket in the test dataset, we constructed a subset in which each segment has at least 8 other vehicles that could be controlled by an interactive agent. Note that this is a more challenging dataset because segments with more road users tend to be more difficult. Starting from an equal number of segments per bucket and discarding segments with an insufficient number of other vehicles, the number of segments remaining in each bucket in order of increasing difficulty accounted for 0.81%, 1.59%, 2.47%, 3.66%, 5.76%, 8.46%, 11.67%, 16.97%, 22.97%, and 25.64% of the total. We evaluated the Baseline-10% and Uniform-10% planning agent variants on this dataset by using the same initial conditions and goal route as the original test dataset. Table 5 demonstrates that the route failure rate decreases for all variants when using interactive agents, and the collision, off-road, and overall failure rates either decrease or remain the same. Additional investigation is needed to determine why the route progress ratio increases for the Baseline-10% variant but decreases for the Uniform-10% variant. These results indicate that our training procedure for the planning agent allows it to perform better in a more realistic simulated environment, not worse. In fact, we expect the agent to have even better performance when evaluated with interactive agents on the original data distribution (i.e., without the requirement that at least 8 vehicles are available for replacement with interactive agents), which would be inherently easier than the subset with interactive agents. While training the planning agent with interactive agents may result in performance gains, this approach is orthogonal to the curriculum learning framework we present and can be easily combined with it. In fact, concurrent work by Zhang et al. [48] investigates this idea, finding that targeted training on more challenging closed-loop scenarios results in more robust agents while requiring less data.

Table 5: Evaluation of agent variants and baselines without interactive agents on the original test set vs. with interactive agents on a subset where at least 8 vehicles are available for replacement. For all metrics except route progress, lower is better.

Agent Variant	Without Interactive Agents					With Interactive Agents				
	Route Failure rate (%)	Collision rate (%)	Off-road rate (%)	Route Progress ratio (%)	Failure rate (%)	Route Failure rate (%)	Collision rate (%)	Off-road rate (%)	Route Progress ratio (%)	Failure rate (%)
Baseline-10%	**1.34**$\pm$**0.06**	1.50$\pm$0.09	**0.67**$\pm$**0.06**	**81.12**$\pm$**0.37**	**3.28**$\pm$**0.13**	1.03$\pm$0.05	1.47$\pm$0.10	0.67$\pm$0.03	**84.58**$\pm$**0.22**	3.05$\pm$0.13
Uniform-10%	**1.35**$\pm$**0.09**	**1.17**$\pm$**0.08**	**0.75**$\pm$**0.07**	**80.67**$\pm$**0.73**	**3.07**$\pm$**0.17**	**0.78**$\pm$**0.07**	**1.16**$\pm$**0.06**	**0.29**$\pm$**0.04**	75.5$\pm$0.61	**2.13**$\pm$**0.11**